



Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics.

P.H. Maire, Boniface Nkonga

► To cite this version:

P.H. Maire, Boniface Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics.. Journal of Computational Physics, 2009, Volume 228 (Issue 3, 20 February 2009), pp.Pages 799-821. 10.1016/j.jcp.2008.10.012 . inria-00290717

HAL Id: inria-00290717

<https://inria.hal.science/inria-00290717>

Submitted on 26 Jun 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics.

Pierre-Henri Maire[†] and Boniface Nkonga[‡]

[†] UMR CELIA, Université Bordeaux I, 33405 Talence Cedex, France

[‡] IMB, Université Bordeaux I, INRIA Bordeaux-Sud Ouest, 33 405 Talence Cedex, France

June 26, 2008

Abstract

This work presents a multidimensional cell-centered unstructured finite volume scheme for the solution of multimaterial compressible fluid flows written in the Lagrangian formalism. This formulation is considered in the Arbitrary-Lagrangian-Eulerian (ALE) framework with the constraint that the mesh and the fluid velocity coincide. The link between the vertex velocity and the fluid motion is obtained by a formulation of the momentum conservation on a class of multi-scale encased volumes around mesh vertices. The vertex velocity is derived with a nodal Riemann solver constructed in such a way that the mesh motion and the face fluxes are compatible. Finally, the resulting scheme conserves both momentum and total energy and, it satisfies a semi-discrete entropy inequality. The numerical results obtained for some classical 2D and 3D hydrodynamic test cases show the robustness and the accuracy of the proposed algorithm.

1 Introduction

The main feature of the Lagrangian numerical methods lies in the fact that the motion of the fluid is intrinsically linked to the geometrical transformation that follows fluid path. This ensures that there is no mass flux crossing the boundary of control volume moving with the fluid. Thus, interfaces in multidimensional flows are sharply resolved. In this framework, one has to discretize not only the gas dynamics conservation laws but also the points motion in order to move the mesh.

At the discrete level, the most natural framework employs staggered-grid hydrodynamics (SGH). The term staggered refers to spatial centering in which position, velocity and kinetic energy are centered at points, while density, pressure and internal energy are within cells. The one-dimensional scheme was developed by the pioneer work of von Neumann and Richtmyer [32] and its two-dimensional extension was achieved by Wilkins [33]. In its original version this finite difference scheme has some drawback related to conservation loss, control of entropy production via artificial viscosity and spurious grid distortion. Many improvements have been made in order to increase the accuracy and the robustness of SGH discretization. In [12], artificial grids distortion and hourglass-types motions are under control by means of Lagrangian sub-zonal masses and pressures. The construction of a *compatible* SGH discretization [10] leads to a scheme that conserves total energy. Moreover, this compatible derivation allows for the specification of forces such as those derived from an artificial viscosity or sub-zonal pressures. Finally, the discretization of artificial viscosity has been considerably improved: first by introducing formulations for

multidimensional shock wave computations [9] and then by deriving a tensorial artificial viscosity with a mimetic finite difference discretization [7]. The three-dimensional extension of the SGH discretization can be found in [11] and also in [6].

In [28, 27], the variational multi-scale stabilized (VMS) approach was applied in finite element computations of Lagrangian hydrodynamics. In that case, a piecewise linear continuous approximation was adopted for the variables. The case of Q1/P0 finite element is investigated in [29] wherein the kinematic variables are represented using a piecewise linear continuous approximation while the thermodynamic variables utilize a piecewise constant one. To capture shock wave, VMS method needs a tensorial artificial viscosity. It shows promising results for two and three-dimensional shock computations. However, it must be noted that it cannot properly preserve interface in the case of multimaterial flows.

An alternative to the previous discretizations is to use a cell-centered discretization in which all the physical variables (density, momentum, pressure, total and internal energy) are within the cell. The fluxes and the node displacement are both computed using Riemann problems at interfaces. This method for Lagrangian gas dynamics in one dimension, has been first introduced by Godunov, see [17] and [26]. Its two-dimensional extension has been performed during the eighties, [2] and [16]. This scheme is a cell-centered finite volume scheme on moving structured or unstructured meshes. It is constructed by integrating directly the system of conservation laws on each moving cell. The flux across the boundary of the cell is computed by solving exactly or approximately a one-dimensional Riemann problem in the direction normal to the boundary. The main problem with the two-dimensional version lies in the fact that the node velocity needed to move the mesh cannot be directly calculated. In [2], the node velocity is computed via a special least squares procedure. It consists in minimizing the error between the normal velocity coming from the Riemann solver and the normal projection of the vertex velocity. It turns out that it leads to an artificial grid motion, which requires a very expensive treatment [15]. Moreover, with this approach the flux calculation is not consistent with the node motion. Recently, new cell-centered methods have been proposed in [19] and [1]. These new approaches use a fully Lagrangian form of the gas dynamics equations, that is, the gradient and divergence operators are expressed in the Lagrangian coordinates. This type of discretization needs to follow the Jacobian matrix associated to the map between Lagrangian and Eulerian spaces. In [19], the coefficients of the Jacobian matrix are introduced as new independent variables located within the cells. The vertex velocity is obtained by inverting compatibility conditions. In this case, the node displacement is not consistent with the flux computation. To solve this problem Després and Mazeran in [13] use the free divergence constraint to discretize the coefficients of the Jacobian matrix at the interfaces. They show that it amounts to compute the node velocity in a consistent way. The key point in this work is the introduction of nodal Riemann solver such as to derive the nodal velocity coherently with the interface fluxes. The global conservation of momentum and total energy is achieved and semi-discrete entropy inequality is provided. However, this nodal solver exhibits a strong dependence to the cell aspect ratio which can lead to severe numerical instabilities. This difficulty is critical for Lagrangian hydrodynamics computations and thus has motivated Maire et al. [22] to propose an alternative scheme that successfully solves the aspect ratio problem and in the same time keeps the consistency between the node displacement and the fluxes computation. The main new feature of this algorithm is the introduction of four pressures on each edge, two for each node on each side of the edge, this is the main difference from [13]. This scheme also locally conserves momentum, total energy and it satisfies a local entropy inequality.

In the present paper, the Lagrangian formulation is considered in the Arbitrary-Lagrangian-Eulerian (ALE) framework with the constraint that the mesh and the fluid velocity coincide. We develop a finite volume formulation for unstructured grids, wherein control volumes are made of polygons in 2D and polyhedra in 3D. The only assumption needed for our scheme is that the faces of the cells must be $(d - 1)$ simplices, where d is the dimension of the space.

For $d = 2$ this assumption is always satisfied, for $d = 3$ we split each face of the polyhedral cells into triangles. The link between the vertex velocity and the fluid motion is obtained by a formulation of the momentum conservation on a class of sub-scales encased volumes around mesh vertices. This lead to a balance relation between sub-zonal forces which are computed by half-Riemann problems to define vertices motion and control the numerical viscosity. In order to ensure conservation properties and consistency with the Geometrical Conservation Law (GCL), velocity and pressure are interpolated on interfaces with two different sets of linear functions, satisfying an orthogonality property. The resulting numerical fluxes are compatible with the mesh motion. Moreover, in two dimensional case, this formulation recovers the schemes proposed in [13] and [22].

The paper is organized as follows. First, we derive the spatial first order discretization of the gas dynamics equations in Lagrangian form and give its main features. Then, we show how to construct the second order extension thanks to piecewise linear monotonic reconstruction. We achieve the derivation of the numerical scheme by developing its second order time discretization and by giving some practical issues related to the time step limitations. Last, we validate our new scheme with 2D and 3D test cases. They are representative test cases for compressible fluid flows and demonstrate the robustness and the accuracy of this new scheme.

2 Spatial discretization

2.1 Governing equations

Let \mathcal{D} be an open subset of \mathbb{R}^d , $d = 2$ or 3 , filled with an inviscid ideal fluid and equipped with an orthonormal frame. We are interested in discretizing the equations of the Lagrangian hydrodynamics. It is convenient, from the point of view of subsequent discretization to write the unsteady compressible Euler equations in the control volume formulation which holds for an arbitrary moving control volume

$$\frac{d}{dt} \int_{V(t)} \rho \, dV + \int_{S(t)} \rho (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS = 0, \quad (1a)$$

$$\frac{d}{dt} \int_{V(t)} \rho \mathbf{U} \, dV + \int_{S(t)} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS = - \int_{S(t)} P \mathbf{N} \, dS, \quad (1b)$$

$$\frac{d}{dt} \int_{V(t)} \rho E \, dV + \int_{S(t)} \rho E (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS = - \int_{S(t)} P \mathbf{U} \cdot \mathbf{N} \, dS. \quad (1c)$$

Here, $V(t)$ is the moving control volume, and $S(t)$ its boundary. ρ , \mathbf{U} , P , E are the mass density, velocity, pressure and specific total energy of the fluid. \mathbf{N} denotes the unit outward normal vector to the moving boundary $S(t)$ whose velocity is denoted by $\boldsymbol{\kappa}$ (kinematic velocity). Implicit in the use of these equations is also the conservation of volume:

$$\frac{d}{dt} \int_{V(t)} dV - \int_{V(t)} \boldsymbol{\kappa} \cdot \mathbf{N} \, dS = 0. \quad (2)$$

This equation is also named geometric conservation law (GCL) and, it is equivalent to the local kinematic equations

$$\frac{d\mathbf{X}}{dt} = \boldsymbol{\kappa}, \quad \mathbf{X}(0) = \mathbf{x}, \quad (3)$$

where \mathbf{X} are coordinates defining the control volume surface at time $t > 0$ and \mathbf{x} are the coordinates at time $t = 0$. Then, $\mathbf{X} = \mathbf{X}(\mathbf{x}, t)$ are implicitly defined by the local kinematic equations, which are also named the trajectory equations. This enables us to define the map

$$\begin{aligned} \mathcal{M}_t : V(0) &\rightarrow V(t) \\ \mathbf{x} &\mapsto \mathbf{X}(\mathbf{x}, t) \end{aligned}$$

where \mathbf{X} is the unique solution of (3). With fixed t , this map advances each fluid particle from its position at time $t = 0$ to its position at time t . Let us introduce $\mathbf{F} = \nabla_{\mathbf{x}} \mathbf{X}$, the Jacobian matrix of this map and J its determinant. Then, time differentiation of J gives the classical equation

$$\frac{dJ}{dt} - J \nabla \cdot \boldsymbol{\kappa} = 0,$$

which is nothing but the local version of the GCL equation (2).

The thermodynamical closure of the set of equations (1) is obtained by the addition of an equation of state which is taken to be of the form

$$P = P(\rho, \varepsilon), \quad (4)$$

where the specific internal energy, ε , is related to the specific total energy by $\varepsilon = E - \frac{1}{2} \|\mathbf{U}\|^2$. The set of previous equations is referred to as the Arbitrary Lagrangian Eulerian (ALE) integral form of the Euler equations and can be found in many papers [2, 18]. Equations (1a), (1b) and (1c) express the conservation of mass, momentum and total energy.

In the Lagrangian formalism the rates of change of volume, mass, momentum and energy are computed assuming that the computational volumes are following the material motion. This leads to the following set of equations

$$\frac{d}{dt} \int_{V(t)} \rho \, dV = 0, \quad (5a)$$

$$\frac{d}{dt} \int_{V(t)} \rho \mathbf{U} \, dV + \int_{S(t)} P \mathbf{N} \, dS = \mathbf{0}, \quad (5b)$$

$$\frac{d}{dt} \int_{V(t)} \rho E \, dV + \int_{S(t)} P \boldsymbol{\kappa} \cdot \mathbf{N} \, dS = 0, \quad (5c)$$

where the kinematic velocity $\boldsymbol{\kappa}$ is obtained from the kinematic constraint

$$\forall \mathbf{X} \in S(t), \quad \boldsymbol{\kappa}(\mathbf{X}, t) = \mathbf{U}(\mathbf{X}, t). \quad (6)$$

We notice that equation (5a) implies that the mass of the control volume remains constant.

2.2 Notations and assumptions

Let us consider the physical domain $V(0)$ that is initially filled by the fluid. We assume that we can map it by a set of cells without gaps or overlaps. Each cell may be a general polygon in 2D or a general polyhedron in 3D. It is assigned a unique index c , and is denoted by $\Omega_c(0)$. For a given time $t > 0$, we set $\Omega_c(t) = \mathcal{M}_t[\Omega_c(0)]$, where \mathcal{M}_t is the map previously defined. Here, we assume that $\Omega_c(t)$ is still a polygon in 2D or a polyhedron in 3D, that is, **the map \mathcal{M}_t is a continuous and linear function over each element of the mesh**. Here, we have used the term polyhedron to describe the types of three-dimensional cells over which we discretize the conservation laws. Our definition of a polyhedral cell is a volume enclosed by an arbitrary number of faces, each determined by an arbitrary number (3 or more) of vertices. If a face has four or more vertices, they can be non-coplanar, thus the face is not a plane and it is not possible to define its unit outward normal. In order to avoid this problem, we decide to divide each face into triangular facets without adding supplementary vertices because we do not want to add supplementary unknowns. For instance, in the case of an hexahedron cell, each quadrangular face is divided into two triangular faces. For the subsequent discretization, we assume that the boundary, $\partial\Omega_c(t)$, of the cell c is the reunion of faces which are $(d-1)$ -simplices, that is, segment for $d = 2$ and triangles for $d = 3$.

Each vertex of the mesh is assigned a unique index p and we denote by $\mathcal{C}(p)$ the set of cells that share a particular vertex p . We subdivide each cell into a set of sub-cells. Each sub-cell is

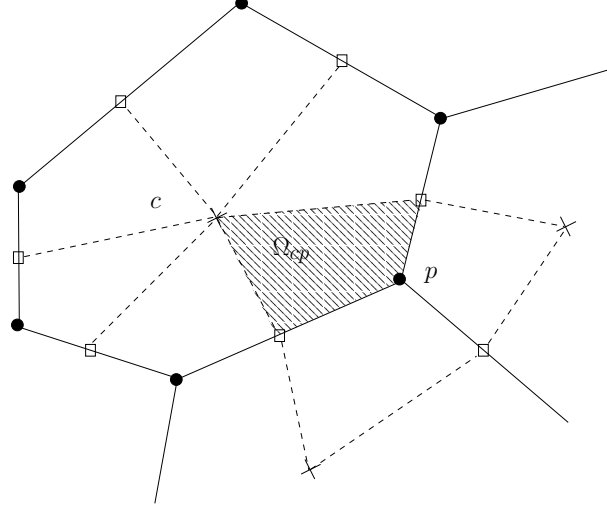


Figure 1: Fragment of a 2D unstructured grid, including cell Ω_c and point p . The solid lines defines the primal grid, and the dashed lines show the median mesh. The median mesh is formed by connecting the cell centers, \times , to the mid-side points, \diamond . The dashed area shows the sub-cell Ω_{cp} .

uniquely defined by a pair of indices c and p and is denoted by $\Omega_{cp} = \Omega_{pc}$. In 2D, this sub-cell is constructed by connecting the centroid of the cell Ω_c to the midpoints of cell edges impinging on point p , see Figure 1. In order to define the sub-cell in 3D we must define the following auxiliary points: the centroid of the cell, the centroid of each face and the midpoint of each edge impinging on point p . In addition, these points are connected by straight lines. The reunion of the sub-cells Ω_{cp} that share a particular point p enables to define the vertex-centered cell, Ω_p , related to the vertex p

$$\Omega_p = \bigcup_{c \in \mathcal{C}(p)} \Omega_{cp}. \quad (7)$$

With the previous notations, we have introduced a primal mesh formed by the cells Ω_c and a dual one formed by the vertex-centered cells Ω_p . This dual mesh is also named median mesh. Here, we use completely similar notations to those introduced in [8]. We denote by $\partial\Omega_p$ and $\partial\Omega_{cp}$ the boundaries of the cells Ω_p and Ω_{cp} .

In order to improve the readability of the paper we give hereafter the list of the different sets of index that will be used throughout the paper. Let us note that the generic indices utilized in the sequel are c for the cells, p for the points (vertices) and f for the faces.

- Sets related to the cell c :
 $\mathcal{P}(c)$ denotes the set of points of the cell c ;
 $\mathcal{F}(c)$ denotes the set of faces of the cell c ;
 $\mathcal{F}_p(c)$ denotes the set of faces of the cell c that share point p .
- Sets related to the point p :
 $\mathcal{C}(p)$ denotes the set of cells that share point p ;
 $\mathcal{F}(p)$ denotes the set of faces that share point p .
- Sets related to the face f :
 $\mathcal{P}(f)$ denotes the set of points of the face f .

- Sets related to the sub-cell cp :
 $\mathcal{F}_c(cp)$ denotes the set of faces of the sub-cell cp that share the centroid of cell c ;
 $\mathcal{F}_p(cp)$ denotes the set of faces of the sub-cell cp that share the point p .

Using the previous notations it is straightforward to show that

$$\mathcal{F}(c) = \bigcup_{p \in \mathcal{P}(c)} \mathcal{F}_p(cp). \quad (8)$$

2.3 Discretization of the GCL

In order to define a numerical scheme for the GCL we have to properly define an approximated map \mathcal{M}_t . For sake of simplicity we first make the supplementary assumption that the cells are d -simplices. Therefore, a continuous and linear map over each element will preserve the mesh structure. Using the finite element formalism, this transformation is defined on the cell $\Omega_c(t)$ by the position of its vertices $\mathbf{X}_p(t)$

$$\mathbf{X}_c(\mathbf{x}, t) = \sum_{p \in \mathcal{P}(c)} \varphi_p(\mathbf{X}) \mathbf{X}_p(t), \quad (9)$$

where $\mathcal{P}(c)$ is the set of vertices of cell $\Omega_c(t)$ and $\varphi_p(\mathbf{X})$ is the barycentric coordinate related to vertex p . Since the transformation is linear, the barycentric coordinate is invariant, that is, $\varphi_p(\mathbf{X}) = \varphi_p(\mathbf{x})$. Therefore, the approximated kinematic velocity field is obtained by time differentiation of (9)

$$\boldsymbol{\kappa}_c(\mathbf{X}, t) = \sum_{p \in \mathcal{P}(c)} \varphi_p(\mathbf{x}) \boldsymbol{\kappa}_p(t), \quad (10)$$

where $\boldsymbol{\kappa}_p$ is the velocity of vertex p . The Jacobian matrix \mathbf{F}_c of this transformation is written

$$\mathbf{F}_c = \sum_{p \in \mathcal{P}(c)} \boldsymbol{\kappa}_p(t) \otimes \nabla_{\mathbf{x}} \varphi_p,$$

where $\nabla_{\mathbf{x}}$ is the gradient operator defined with the \mathbf{x} coordinates. \mathbf{F}_c is a constant function since φ_p is linear. Therefore, its determinant J_c is also a constant function over the cell Ω_c , which is equal to the volume of the cell.

The GCL equation is written

$$\begin{aligned} \frac{dV_c}{dt} &= \int_{\partial\Omega_c(t)} \boldsymbol{\kappa}_c \cdot \mathbf{N} \, dS, \\ &= \sum_{f \in \mathcal{F}(c)} \int_{S_f} \boldsymbol{\kappa}_c \cdot \mathbf{N} \, dS. \end{aligned}$$

Here, $V_c = \int_{\Omega_c} dV$ denotes the volume of the cell Ω_c , $\boldsymbol{\kappa}_c$ is defined by (10) and $\mathcal{F}(c)$ denotes the set of faces of Ω_c . We notice that each face S_f of the cell Ω_c is a $(d-1)$ -simplex and, the restriction of $\boldsymbol{\kappa}_c$ over the face S_f is linear. Therefore, the integral in the previous equation is written

$$\int_{S_f} \boldsymbol{\kappa}_c \cdot \mathbf{N} \, dS = \frac{1}{d} \left(\sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p \right) \cdot S_f \mathbf{N}_f^c,$$

where $\mathcal{P}(f)$ is the set of vertices of face S_f and \mathbf{N}_f^c its unit outward normal. Combining the previous results we obtain the discretization of the GCL equation for a simplicial mesh

$$\frac{dV_c}{dt} - \sum_{f \in \mathcal{F}(c)} \frac{1}{d} \left(\sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p \right) \cdot S_f \mathbf{N}_f^c = 0. \quad (11)$$

This fundamental equation describes the Lagrangian representation of fluid flow. It is generalized to the case of a non simplicial mesh in 2D (resp. in 3D) by triangularizing (resp. tetrahedralizing) polygons (resp. polyhedra) of arbitrary order so that the volume variation of any cells can be computed in a general manner on an unstructured grid.

Comment 1. *The GCL discretization can be utilized in a different way by noticing that equation (11) determines the discrete form of the divergence of the velocity field over the cell Ω_c*

$$\begin{aligned} (\nabla \cdot \boldsymbol{\kappa})_c &= \frac{1}{V_c} \frac{dV_c}{dt} \\ &= \frac{1}{V_c} \sum_{f \in \mathcal{F}(c)} \frac{1}{d} \left(\sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p \right) \cdot S_f \mathbf{N}_f^c \\ &= \frac{1}{V_c} \sum_{p \in \mathcal{P}(c)} \frac{1}{d} \left(\sum_{f \in \mathcal{F}_p(c)} S_f \mathbf{N}_f^c \right) \cdot \boldsymbol{\kappa}_p. \end{aligned}$$

Here, $\mathcal{F}_p(c)$ denotes the subset of faces of Ω_c that share point p . By defining the corner vector, $\boldsymbol{\Gamma}_p^c$, that is associated with cell c and point p

$$\boldsymbol{\Gamma}_p^c = \frac{1}{d} \sum_{f \in \mathcal{F}_p(c)} S_f \mathbf{N}_f^c,$$

we finally obtain

$$(\nabla \cdot \boldsymbol{\kappa})_c = \frac{1}{V_c} \sum_{p \in \mathcal{P}(c)} \boldsymbol{\Gamma}_p^c \cdot \boldsymbol{\kappa}_p. \quad (12)$$

This last equation has been previously derived and utilized in [10, 11] in order to construct compatible hydrodynamics algorithms using the method of Support Operators.

2.4 Discretization of the physical conservation laws

The aim of this section is to provide a discretization for the system of the physical conservation laws. Setting

$$\mathbf{q} = \begin{pmatrix} 1 \\ \mathbf{U} \\ E \end{pmatrix}, \quad \phi(\mathbf{q}, \mathbf{N}) = \begin{pmatrix} 0 \\ P\mathbf{N} \\ P\boldsymbol{\kappa} \cdot \mathbf{N} \end{pmatrix}, \quad (13)$$

the system (5) can be written in the general form

$$\frac{d}{dt} \int_{\Omega_c(t)} \rho \mathbf{q} \, dV + \int_{\partial\Omega_c(t)} \phi(\mathbf{q}, \mathbf{N}) \, dS = 0. \quad (14)$$

Here, we have used the control volume defined by the primal cell Ω_c . We notice that the first equation of the above system corresponds to mass conservation. Its integration provides

$$\int_{\Omega_c(t)} \rho \, dV = m_c,$$

where m_c is the constant mass of the cell Ω_c . Let us define the mass averaged value of q over the cell Ω_c

$$\mathbf{q}_c = \frac{1}{m_c} \int_{\Omega_c} \rho \mathbf{q} \, dV.$$

By applying this definition for $q = \frac{1}{\rho}$ we get the following form of mass conservation

$$V_c(t) = \frac{m_c}{\rho_c(t)}, \quad (15)$$

where ρ_c is the mean value of the mass density over cell Ω_c . By using (15), we notice that the GCL can be rewritten

$$m_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) - \int_{\partial\Omega_c(t)} \boldsymbol{\kappa} \cdot \mathbf{N} \, dS = 0. \quad (16)$$

Therefore, the GCL can be viewed also as a physical conservation law related to the specific volume $\frac{1}{\rho_c}$. With the previous comments the system of physical conservation laws has the generic form

$$m_c \frac{d}{dt} \mathbf{q}_c + \int_{\partial\Omega_c(t)} \phi(\mathbf{q}, \mathbf{N}) \, dS = 0, \quad (17)$$

where we have set

$$\mathbf{q}_c = \begin{pmatrix} \frac{1}{\rho_c} \\ \mathbf{U}_c \\ E_c \end{pmatrix}, \quad \phi(\mathbf{q}, \mathbf{N}) = \begin{pmatrix} -\boldsymbol{\kappa} \cdot \mathbf{N} \\ P\mathbf{N} \\ P\boldsymbol{\kappa} \cdot \mathbf{N} \end{pmatrix} \quad (18)$$

We assume that the boundary, $\partial\Omega_c(t)$, of cell c is the reunion of faces $S_f(t)$ which are $(d-1)$ -simplices. Thus, the physical conservation laws become

$$m_c \frac{d}{dt} \mathbf{q}_c + \sum_{f \in \mathcal{F}(c)} \phi_f^c = 0, \quad (19)$$

where the face flux ϕ_f^c is defined by

$$\phi_f^c = \int_{S_f(t)} \phi(\mathbf{q}, \mathbf{N}) \, dS.$$

Since the face $S_f(t)$ is a $(d-1)$ -simplex its unit outward normal is constant, i.e. $\mathbf{N}(\mathbf{X}, t) = \mathbf{N}(t)$. Therefore, the numerical flux is written

$$\phi_f^c = \begin{pmatrix} -\boldsymbol{\kappa}_f^* \cdot S_f \mathbf{N}_f^c \\ P_f^* S_f \mathbf{N}_f^c \\ (P\boldsymbol{\kappa})_f^* \cdot S_f \mathbf{N}_f^c \end{pmatrix}. \quad (20)$$

$\boldsymbol{\kappa}_f^*$, P_f^* and $(P\boldsymbol{\kappa})_f^*$ denote the face fluxes defined by

$$\boldsymbol{\kappa}_f^* = \frac{1}{S_f(t)} \int_{S_f(t)} \boldsymbol{\kappa}(\mathbf{X}, t) \, dS, \quad (21a)$$

$$P_f^* = \frac{1}{S_f(t)} \int_{S_f(t)} P(\mathbf{X}, t) \, dS, \quad (21b)$$

$$(P\boldsymbol{\kappa})_f^* = \frac{1}{S_f(t)} \int_{S_f(t)} (P\boldsymbol{\kappa})(\mathbf{X}, t) \, dS. \quad (21c)$$

Finally, we obtain the following set of discrete equations for the discrete variables $(\frac{1}{\rho_c}, \mathbf{U}_c, E_c)$

$$m_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) - \sum_{f \in \mathcal{F}(c)} S_f \mathbf{N}_f^c \cdot \boldsymbol{\kappa}_f^* = 0, \quad (22a)$$

$$m_c \frac{d}{dt} (\mathbf{U}_c) + \sum_{f \in \mathcal{F}(c)} S_f \mathbf{N}_f^c P_f^* = \mathbf{0}, \quad (22b)$$

$$m_c \frac{d}{dt} (E_c) + \sum_{f \in \mathcal{F}(c)} S_f \mathbf{N}_f^c \cdot (P\boldsymbol{\kappa})_f^* = 0. \quad (22c)$$

The points motion is given by the discrete trajectory equation

$$\frac{d}{dt} \mathbf{X}_p = \boldsymbol{\kappa}_p, \quad \mathbf{X}_p(0) = \mathbf{x}_p, \quad (23)$$

where $\boldsymbol{\kappa}_p \equiv \boldsymbol{\kappa}_p(\frac{1}{\rho_c}, \mathbf{U}_c, E_c)$ is the point velocity. We notice that all primary variables, including material velocity, are cell-centered as it is done in [2, 22]. In order to complete this discretization the following important problems arise:

- How do we compute the face fluxes defined by system (21)?
- How do we compute the point velocities $\boldsymbol{\kappa}_p$?
- These velocities being known, how can we ensure the compatibility between the mesh motion and the volume variations of the cells?

We have already answered to the last question. In fact we have shown previously that (22a) is fully equivalent to the GCL provided the face flux $\boldsymbol{\kappa}_f^*$ is written

$$\boldsymbol{\kappa}_f^* = \frac{1}{d} \sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p. \quad (24)$$

This fundamental relation enables to write two equivalent discretizations of the specific volume variation. It can be given in term of the flux through the faces, or equivalently, in term of vertex fluxes. Moreover, the two discretizations are *compatible* with the point motion under the condition (24).

Hence, one can consider two methods for computing the face velocities:

- The first one relies on the evaluation of the normal velocity using a one-dimensional Riemann solver at faces. The vertex velocities have to be computed by solving a linear system built from equations (24) written for all the faces. This system is in general singular, that is why we give up this approach and shall adopt a more robust method.
- In the second method, the point velocities are first evaluated using some still-to-define solver. The face velocities are then computed from (24). This is the technique we will use in the sequel of the paper because this method will guarantee the compatibility between vertex motion and cell volumes variation.

Thus, our task consists in building a numerical solver that can compute the face fluxes P_f^* , $(P\boldsymbol{\kappa})_f^*$ and the point velocity $\boldsymbol{\kappa}_p$. We resolve these questions in the next sections.

2.5 Derivation of the point velocity

We define a strategy to compute the nodal velocity. We recall that the kinematic velocity is assumed to be continuous and linear in order to define a coherent map \mathcal{M}_t that preserves the structure of the mesh.

One natural way to obtain the nodal velocity consists in choosing the dual cell Ω_p instead of the primal cell Ω_c as a control volume for the momentum equation

$$\frac{d}{dt} \int_{\Omega_p} \rho \boldsymbol{\kappa} \, dV + \int_{\partial\Omega_p} P \mathbf{N} \, dS = \mathbf{0}. \quad (25)$$

It leads to a staggered spatial placement of the variables wherein the position and the velocity are defined at grid points, and density, internal energy, and the pressure are defined at cell centers [10].

Here, we will proceed differently by keeping a centered spatial placement of the physical variables (density, momentum, pressure, total energy) and in the same time defining coherently

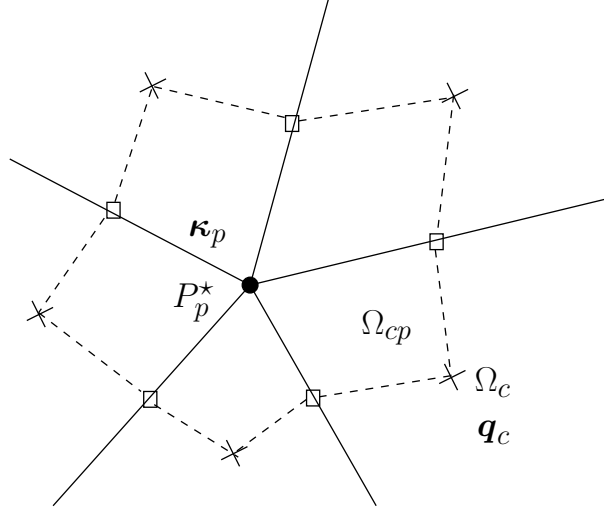


Figure 2: Initial conditions for the multi-dimensional Riemann problem at point p .

the kinematic velocity to move the mesh. Consider a point p and the cells $c \in \mathcal{C}(p)$ that share this point. In each cell c the fluid flow is characterized by the constant state $\mathbf{q}_c = (\frac{1}{\rho_c}, \mathbf{U}_c, E_c)^t$. One way to compute the point velocity $\boldsymbol{\kappa}_p$ consists in solving the multi-dimensional Riemann problem at point p defined by the initial conditions $\mathbf{q}_c, c \in \mathcal{C}(p)$, see Figure 2. Its solution would provide also the point pressure P_p^* , which is the instantaneous value of pressure at point p immediately following the breakdown of the initial discontinuity. Knowing $\boldsymbol{\kappa}_p$ and P_p^* , we could easily compute the face fluxes ϕ_f^c and design a conservative scheme. Unfortunately, such an approach is not possible since up to our knowledge the solution of such a multi-dimensional Riemann problem is still not known.

2.5.1 The classical approach

Dukowicz and its co-authors [2] suggest an alternative approach based on one-dimensional Riemann problems. Let us consider c_1 and c_2 , two adjacent cells that share point p , see Figure 3. We denote by f the common face to these cells and by $\mathbf{N}_f^{c_1}, \mathbf{N}_f^{c_2}$ the unit outward normals related to this face. We set $\mathbf{N}_f = \mathbf{N}_f^{c_2} = -\mathbf{N}_f^{c_1}$. Here, the one-dimensional Riemann problem is defined by the discontinuity of the state variables \mathbf{q}_c on either side of the cell face f , in the vicinity of point p . The solution of this Riemann problem provides the unique pressure P_{pf}^* and normal velocity $\boldsymbol{\kappa}_{pf}^* \cdot \mathbf{N}_f$ of the contact surface. In the case of an approximate acoustic Riemann solver, these values satisfy the following linear system [14]

$$P_{c_1} - P_{pf}^* = -Z_{c_1}(\boldsymbol{\kappa}_{pf}^* - \mathbf{U}_{c_1}) \cdot \mathbf{N}_f, \quad (26a)$$

$$P_{c_2} - P_{pf}^* = Z_{c_2}(\boldsymbol{\kappa}_{pf}^* - \mathbf{U}_{c_2}) \cdot \mathbf{N}_f, \quad (26b)$$

where Z_{c_i} for $i = 1, 2$ is the acoustic impedance. A straightforward calculation shows that

$$P_{pf}^* = \frac{Z_{c_1}P_{c_2} + Z_{c_2}P_{c_1}}{Z_{c_1} + Z_{c_2}} - \frac{Z_{c_1}Z_{c_2}}{Z_{c_1} + Z_{c_2}}(\mathbf{U}_{c_1} - \mathbf{U}_{c_2}) \cdot \mathbf{N}_f,$$

$$\boldsymbol{\kappa}_{pf}^* \cdot \mathbf{N}_f = \frac{(Z_{c_1}\mathbf{U}_{c_1} + Z_{c_2}\mathbf{U}_{c_2}) \cdot \mathbf{N}_f}{Z_{c_1} + Z_{c_2}} - \frac{P_{c_1} - P_{c_2}}{Z_{c_1} + Z_{c_2}}.$$

Knowing P_{pf}^* and $\boldsymbol{\kappa}_{pf}^* \cdot \mathbf{N}_f$ one gets immediately the face fluxes however, the point velocity is still unknown. In [2], $\boldsymbol{\kappa}_p$ is taken to be the vector whose components normal to adjacent cell

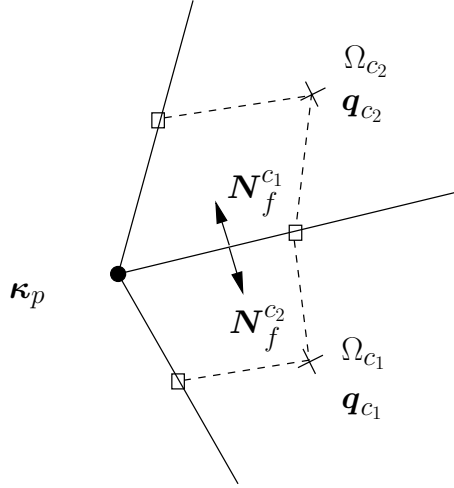


Figure 3: Initial conditions for the one-dimensional Riemann problem at face f in the vicinity of point p .

faces agrees with the Riemann velocity at each adjacent cell face, in a weighted least squares sense

$$\kappa_p = \operatorname{argmin} \sum_{f \in \mathcal{F}(p)} \omega_f (\kappa_p \cdot \mathbf{N}_f - \kappa_{pf}^* \cdot \mathbf{N}_f)^2,$$

where $\mathcal{F}(p)$ is the set of faces that share point p and ω_f is a positive weight defined in [2]. By noticing that in general $\kappa_p \cdot \mathbf{N}_f \neq \kappa_{pf}^* \cdot \mathbf{N}_f$, we realize that this way of deriving the point velocity leads to an inconsistency with the GCL discretization. In addition, with such an approach the kinematic velocity κ will be discontinuous in the vicinity of point p .

2.5.2 A new approach

To ensure consistency with the GCL discretization and continuity of the kinematic velocity around point p , we propose to solve two half one-dimensional Riemann problems at each cell interface by assuming that its velocity is equal to the point velocity κ_p . Thus, at each face f connected to point p , we introduce two interface pressures P_{pc1f}^* and P_{pc2f}^* , see Figure 4, defined by

$$P_{c1} - P_{pc1f}^* = -Z_{c1} (\kappa_p - \mathbf{U}_{c1}) \cdot \mathbf{N}_f, \quad (27a)$$

$$P_{c2} - P_{pc2f}^* = Z_{c2} (\kappa_p - \mathbf{U}_{c2}) \cdot \mathbf{N}_f. \quad (27b)$$

By subtracting (27b) from (27a), we get

$$P_{pc2f}^* - P_{pc1f}^* = (Z_{c1} + Z_{c2}) (\kappa_{pf}^* \cdot \mathbf{N}_f - \kappa_p \cdot \mathbf{N}_f).$$

We notice that $P_{pc1f}^* = P_{pc2f}^*$ if and only if $\kappa_{pf}^* \cdot \mathbf{N}_f = \kappa_p \cdot \mathbf{N}_f$. In this case, we recover the solution of the one-dimensional Riemann problem and we get $P_{pc1f}^* = P_{pc2f}^* = P_{pf}^*$ and $\kappa_{pf}^* \cdot \mathbf{N}_f = \kappa_p \cdot \mathbf{N}_f = \kappa_{pf}^* \cdot \mathbf{N}_f$. Since in general $\kappa_{pf}^* \cdot \mathbf{N}_f \neq \kappa_p \cdot \mathbf{N}_f$ we obtain the discontinuity $P_{pc1f}^* \neq P_{pc2f}^*$.

Finally, for each simplicial face f we introduce $2d$ face pressures P_{pc1f}^* and P_{pc2f}^* for $p \in \mathcal{P}(f)$, from which we are able to compute the face fluxes. The discontinuity of these pressures implies the loss of momentum conservation for our cell-centered discretization. Moreover, the point

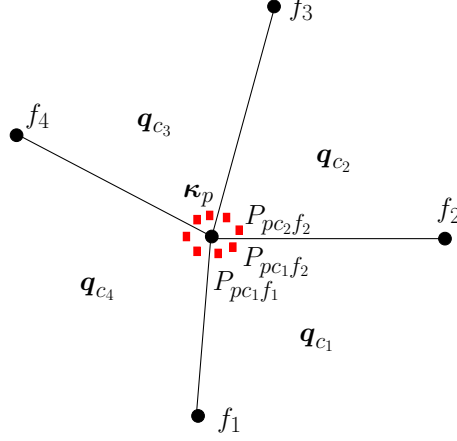


Figure 4: Localization of the multiple pressures for the half Riemann problems in the vicinity of point p .

velocity has not been yet determined. We will show that these two critical issues-momentum conservation and point velocity computation- are strongly linked.

Since the momentum conservation equation, in its ALE form, (1b), is valid for any moving control volume V , we can write it for the dual cell, Ω_p , and for all the subcells, Ω_{cp} , that surround point p

$$\frac{d}{dt} \int_{\Omega_p} \rho \mathbf{U} \, dV + \int_{\partial\Omega_p} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \int_{\partial\Omega_p} P \mathbf{N} \, dS = \mathbf{0}, \quad (28a)$$

$$\frac{d}{dt} \int_{\Omega_{cp}} \rho \mathbf{U} \, dV + \int_{\partial\Omega_{cp}} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \int_{\partial\Omega_{cp}} P \mathbf{N} \, dS = \mathbf{0}, \quad \forall c \in \mathcal{C}(p). \quad (28b)$$

We notice that (28a) expresses momentum conservation over the dual cell, whereas (28b) expresses momentum balance locally over each subcell. Knowing that

$$\Omega_p = \bigcup_{c \in \mathcal{C}(p)} \Omega_{cp}, \quad \partial\Omega_p = \bigcup_{c \in \mathcal{C}(p)} \bigcup_{f \in \mathcal{F}_c(cp)} S_f^{cp},$$

where $\mathcal{F}_c(cp)$ is the subset of faces of Ω_{cp} that share point c , equation (28a) is rewritten

$$\sum_{c \in \mathcal{C}(p)} \left[\frac{d}{dt} \int_{\Omega_{cp}} \rho \mathbf{U} \, dV + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} P \mathbf{N} \, dS \right] = \mathbf{0}. \quad (29)$$

The subdivision of the subcell boundary

$$\partial\Omega_{cp} = \left(\bigcup_{f \in \mathcal{F}_c(cp)} S_f^{cp} \right) \cup \left(\bigcup_{f \in \mathcal{F}_p(cp)} S_f^{cp} \right),$$

where $\mathcal{F}_p(cp)$ is the subset of faces of Ω_{cp} that share point p , enables to rewrite (28b) as

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_{cp}} \rho \mathbf{U} \, dV + \sum_{f \in \mathcal{F}_p(cp)} \int_{S_f} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \sum_{f \in \mathcal{F}_p(cp)} \int_{S_f} P \mathbf{N} \, dS \\ + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} P \mathbf{N} \, dS = \mathbf{0}, \quad \forall c \in \mathcal{C}(p). \end{aligned}$$

We notice that the second term in the left-hand side is equal to zero since Ω_c is a Lagrangian cell ($\mathbf{U} = \boldsymbol{\kappa}$ for $f \in \mathcal{F}_p(cp)$). The summation of the previous equation over all the subcells that surround point p yields

$$\sum_{c \in \mathcal{C}(p)} \left[\frac{d}{dt} \int_{\Omega_{cp}} \rho \mathbf{U} \, dV + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} \rho \mathbf{U} (\mathbf{U} - \boldsymbol{\kappa}) \cdot \mathbf{N} \, dS + \sum_{f \in \mathcal{F}_c(cp)} \int_{S_f} P \mathbf{N} \, dS \right] + \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} \int_{S_f} P \mathbf{N} \, dS = \mathbf{0}.$$

Finally, by subtracting this last equation from (29) we obtain

$$\sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} \int_{S_f} P \mathbf{N} \, dS = \mathbf{0}. \quad (30)$$

By setting

$$P_{pcf}^* = \frac{1}{S_f(t)} \int_{S_f(t)} P(\mathbf{X}, t) \, dS,$$

the equation (30) can be rewritten

$$\sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} S_f P_{pcf}^* \mathbf{N}_f^c = \mathbf{0}. \quad (31)$$

Note that this equation is invariant by any homothety centered at point p . It expresses the balance of momentum locally around point p . Hence P_{pcf}^* can be viewed as a nodal pressure located at point p and related to cell c and face f . Since the velocity of face f in the vicinity of point p is equal to the nodal velocity $\boldsymbol{\kappa}_p$, the nodal pressure P_{pcf}^* is computed using the following half approximate Riemann problem

$$P_c - P_{pcf}^* = Z_c(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c, \quad \text{for } f \in \mathcal{F}_p(cp). \quad (32)$$

Here, Z_c denotes the acoustic impedance defined in cell c and \mathbf{N}_f^c is the unit outward normal related to the face S_f for $f \in \mathcal{F}_p(cp)$. The substitution of (32) into (31) leads to

$$\sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} S_f Z_c (\mathbf{N}_f^c \otimes \mathbf{N}_f^c) \boldsymbol{\kappa}_p = \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} [S_f P_c \mathbf{N}_f^c + S_f Z_c (\mathbf{N}_f^c \otimes \mathbf{N}_f^c) \mathbf{U}_c]. \quad (33)$$

The nodal velocity satisfies a $d \times d$ system. Since the geometric variables (surface S_f and unit outward normal \mathbf{N}_f^c) depend on the nodal velocity $\boldsymbol{\kappa}_p$ via the trajectory equation (23), we notice that the previous system is non linear. This nodal solver is the multi-dimensional extension of the two-dimensional solver derived in [22].

2.5.3 Summary

By setting

$$\mathbf{M}_{pcf} = S_f Z_c (\mathbf{N}_f^c \otimes \mathbf{N}_f^c), \quad \text{for } f \in \mathcal{F}_p(cp), \quad \text{and } \mathbf{M}_p = \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} \mathbf{M}_{pcf}, \quad (34)$$

the point velocity $\boldsymbol{\kappa}_p$ and the point pressure P_{pcf}^* related to cell c and face f are written

$$\boldsymbol{\kappa}_p = \mathbf{M}_p^{-1} \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} [S_f P_c \mathbf{N}_f^c + \mathbf{M}_{pcf} \mathbf{U}_c],$$

$$P_{pcf}^* = P_c - Z_c(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c, \quad \text{for } f \in \mathcal{F}_p(cp).$$

We note that the matrix \mathbf{M}_p is symmetric positive definite and thus always invertible.

Comment 2. Here, we have derived the nodal velocity utilizing the acoustic approximate Riemann solver for which $Z_c = \rho_c a_c$, where a_c is the local speed of sound. As suggested by Dukowicz [14] we can use the artificial shock viscosity approximation by rewriting (32) as

$$P_c - P_{pcf}^* = Z_{pcf}(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c, \quad \text{for } f \in \mathcal{F}_p(cp), \quad (35)$$

with $Z_{pcf} = \rho_c [a_c + A_c |(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c|]$, where A_c is a material-dependent parameter that is given in terms of the density ratio in the limit of very strong shocks. In the case of a gamma law gas one gets $A_c = \frac{\gamma+1}{2}$ where γ is the polytropic index.

Comment 3. Instead of using the unit outward normal, \mathbf{N}_f^c , related to the face S_f in equation (32), one can introduce the average corner normal \mathbf{N}_{pc} defined by

$$S_{pc}\mathbf{N}_{pc} = \sum_{f \in \mathcal{F}_p(cp)} S_f \mathbf{N}_f^c.$$

Then, there is only one half Riemann problem corresponding to this unit normal and, it is written

$$P_c - P_{pc}^* = Z_c(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_{pc}. \quad (36)$$

This amounts to define only one nodal pressure P_{pc}^* for each cell that surrounds point p . With this choice, the system satisfied by the nodal velocity is written

$$\sum_{c \in \mathcal{C}(p)} S_{pc} Z_c (\mathbf{N}_{pc} \otimes \mathbf{N}_{pc}) \boldsymbol{\kappa}_p = \sum_{c \in \mathcal{C}(p)} [S_{pc} P_c \mathbf{N}_{pc} + S_{pc} Z_c (\mathbf{N}_{pc} \otimes \mathbf{N}_{pc}) \mathbf{U}_c]. \quad (37)$$

Using the corner normal \mathbf{N}_{pc} in the definition of the half Riemann problem (32), we have recovered the nodal solver developed in [13]. The force corresponding to the single nodal pressure P_{pc}^* reads $\mathbf{F}_{pc} = S_{pc} P_{pc}^* \mathbf{N}_{pc}$. We note that this subcell force is always colinear to the geometric direction \mathbf{N}_{pc} . Due to this fact, it appears that the nodal solver proposed in [13] exhibits a strong dependence to the cell aspect ratio. This problem can lead to numerical instability.

2.6 Fluxes approximation

The aim of this section is to provide an approximation of the face fluxes $\boldsymbol{\kappa}_f^*$, P_f^* , $(P\boldsymbol{\kappa})_f^*$ for the discrete equations (22) related to the physical conservation laws. The face fluxes approximations are constructed by using a linear mapping over each simplicial faces. This linear mapping utilizes the nodal velocity and pressures provided by the nodal solver. We shall show also that this approximation enables us to recover momentum and total energy conservation.

Let us consider a cell c and one face $f \in \mathcal{F}(c)$. Since S_f is a $(d-1)$ -simplex, for each $p \in \mathcal{P}(f)$ we introduce, φ_p , the barycentric coordinate related to the vertex p . We recall the classical result

$$\frac{1}{S_f} \int_{S_f} \varphi_p(\mathbf{X}) dS = \frac{1}{d}. \quad (38)$$

2.6.1 GCL flux

In order to be self-consistent, we recall briefly the face flux approximation that corresponds to the GCL equation. We have shown previously that the velocity over the face S_f is written

$$\boldsymbol{\kappa}(\mathbf{X}, t) = \sum_{p \in \mathcal{P}(f)} \varphi_p(\mathbf{X}) \boldsymbol{\kappa}_p(t), \quad \text{for } \mathbf{X} \in S_f$$

The substitution of the velocity field into the definition of the face flux (21a) leads to

$$\boldsymbol{\kappa}_f^* = \frac{1}{d} \sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p. \quad (39)$$

2.6.2 Momentum flux

The computations of the momentum and total energy fluxes are made using the following linear interpolation for the pressure

$$P(\mathbf{X}, t) = \sum_{p \in \mathcal{P}(f)} \psi_p(\mathbf{X}) P_{pcf}^*(t), \quad \text{for } \mathbf{X} \in S_f.$$

Here, ψ_p is a linear function over S_f to be determined. Since the nodal pressure P_{pcf}^* is discontinuous across S_f we denote the face flux P_{cf}^* instead of P_f^* . After substitution of the linear interpolation we get

$$\begin{aligned} P_{cf}^* &= \frac{1}{S_f} \int_{S_f} P(\mathbf{X}, t) \, dS \\ &= \sum_{p \in \mathcal{P}(f)} \alpha_p P_{pcf}^*, \end{aligned}$$

where the unknown coefficient α_p is written

$$\alpha_p = \frac{1}{S_f} \int_{S_f} \psi_p(\mathbf{X}) \, dS.$$

We claim that with this approximation of the momentum flux, momentum conservation is ensured. Omitting the boundary conditions and summing the momentum equation on each cell, written with the previous approximation of the momentum flux, we get the global balance of momentum

$$\begin{aligned} \frac{d}{dt} \left(\sum_c m_c U_c \right) &= - \sum_c \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f N_f^c \alpha_p P_{pcf}^* \\ &= - \sum_c \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{F}_p(cp)} S_f N_f^c \alpha_p P_{pcf}^* \quad \text{thanks to (8)} \\ &= - \sum_p \alpha_p \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} S_f N_f^c P_{pcf}^* \\ &= \mathbf{0}. \end{aligned}$$

Here, we have replaced the global summation over cells by a global summation over points and, we have used the fact that P_{pcf}^* satisfies equation (31).

2.6.3 Total energy flux

We compute the total energy flux interpolation by combining the linear interpolations of the velocity and the pressure

$$(P\boldsymbol{\kappa})(\mathbf{X}, t) = \sum_{p \in \mathcal{P}(f)} \sum_{q \in \mathcal{P}(f)} \psi_q(\mathbf{X}) \varphi_p(\mathbf{X}) P_{qcf}^*(t) \boldsymbol{\kappa}_p(t), \quad \text{for } \mathbf{X} \in S_f.$$

Since the nodal pressure P_{pcf}^* is discontinuous across S_f we denote the total energy face flux $(P\boldsymbol{\kappa})_{cf}^*$ instead of $(P\boldsymbol{\kappa})_f^*$. After substitution of the linear interpolation we get

$$\begin{aligned} (P\boldsymbol{\kappa})_{cf}^* &= \frac{1}{S_f} \int_{S_f} (P\boldsymbol{\kappa})(\mathbf{X}, t) \, dS \\ &= \sum_{p \in \mathcal{P}(f)} \sum_{q \in \mathcal{P}(f)} \beta_{pq} P_{qcf}^* \boldsymbol{\kappa}_p, \end{aligned}$$

where the unknown coefficient β_{pq} is written

$$\beta_{pq} = \frac{1}{S_f} \int_{S_f} \varphi_p(\mathbf{X}) \psi_q(\mathbf{X}) dS.$$

We give a sufficient condition to ensure total energy conservation. We claim that if the coefficients β_{pq} are written under the form

$$\beta_{pq} = C_p \delta_{p,q}, \quad (40)$$

where C_p is an unknown coefficient and $\delta_{p,q}$ the Kronecker symbol, then the total energy conservation is ensured. The demonstration consists in writing the global balance of total energy

$$\begin{aligned} \frac{d}{dt} \left(\sum_c m_c E_c \right) &= - \sum_c \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} \sum_{q \in \mathcal{P}(f)} S_f \mathbf{N}_f^c \beta_{pq} P_{qc}^* \cdot \boldsymbol{\kappa}_p \\ &= - \sum_c \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c C_p P_{pc}^* \cdot \boldsymbol{\kappa}_p \quad \text{thanks to (40)} \\ &= - \sum_c \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{F}_p(cp)} S_f \mathbf{N}_f^c C_p P_{pc}^* \cdot \boldsymbol{\kappa}_p \quad \text{thanks to (8)} \\ &= - \sum_p C_p \left(\sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} S_f \mathbf{N}_f^c P_{pc}^* \right) \cdot \boldsymbol{\kappa}_p \\ &= \mathbf{0}. \end{aligned}$$

We note that the term between the parenthesis is null because of the momentum conservation.

2.6.4 Definition of the ψ_p functions

Finally, it remains to determine the linear function ψ_p for $p \in \mathcal{P}(f)$ in order to compute the coefficients α_p and C_p . This determination is performed using the following decomposition of the unknown function over the basis $\{\varphi_p\}$ for $p \in \mathcal{P}(f)$

$$\psi_p = \sum_{q \in \mathcal{P}(f)} A_{pq} \varphi_q, \quad p \in \mathcal{P}(f), \quad (41)$$

where A_{pq} are the d^2 unknown coordinates of ψ_q function. These coordinates are computed using the two following conditions

- Consistency condition: $\{\psi_p\}$ for $p \in \mathcal{P}(f)$ must preserve constant functions, that is,

$$\sum_{p \in \mathcal{P}(f)} \psi_p = 1.$$

The substitution of ψ_p in terms of its coordinates gives the equivalent condition

$$\sum_{p \in \mathcal{P}(f)} A_{pq} = 1, \quad q \in \mathcal{P}(f). \quad (42)$$

- Energy conservation condition: ψ_q must satisfy the condition

$$\frac{1}{S_f} \int_{S_f} \varphi_p(\mathbf{X}) \psi_q(\mathbf{X}) dS = C_p \delta_{p,q}.$$

By substituting ψ_q in terms of its coordinates and using the result

$$\frac{1}{S_f} \int_{S_f} \varphi_p(\mathbf{X}) \varphi_r(\mathbf{X}) dS = \frac{1 + \delta_{p,r}}{d(d+1)},$$

we obtain

$$\frac{1}{S_f} \int_{S_f} \varphi_p(\mathbf{X}) \psi_q(\mathbf{X}) dS = \sum_{r \in \mathcal{P}(f)} A_{qr} \frac{(1 + \delta_{p,r})}{d(d+1)}.$$

Hence, we find that the coordinates must satisfy

$$\sum_{r \in \mathcal{P}(f)} A_{qr} (1 + \delta_{p,r}) = 0, \quad (p, q) \in \mathcal{P}^2(f) \text{ and } p \neq q. \quad (43)$$

The combination of (42) and (43) gives us a linear system of d^2 equations for the d^2 unknown coordinates A_{pq} . We can show that the determinant of this system is always non null, thus this system admits a unique solution which is written

$$A_{pq} = \begin{cases} d & \text{if } p = q, \\ -1 & \text{if } p \neq q. \end{cases}$$

Therefore the functions ψ_p are written

$$\psi_p = (d+1)\varphi_p - 1. \quad (44)$$

We notice that the basis $\{\psi_p\}$ can be viewed as the dual basis of $\{\varphi_p\}$.

Finally, knowing the ψ_p functions, we get $\alpha_p = \frac{1}{d}$ and $\beta_{pq} = \frac{\delta_{p,q}}{d}$.

2.6.5 Summary

With the previous linear interpolations the face fluxes are written

$$\boldsymbol{\kappa}_f^* = \frac{1}{d} \sum_{p \in \mathcal{P}(f)} \boldsymbol{\kappa}_p, \quad (45a)$$

$$P_{cf}^* = \frac{1}{d} \sum_{p \in \mathcal{P}(f)} P_{pcf}^*, \quad (45b)$$

$$(P\boldsymbol{\kappa})_{cf}^* = \frac{1}{d} \sum_{p \in \mathcal{P}(f)} P_{pcf}^* \boldsymbol{\kappa}_p. \quad (45c)$$

We notice that this formulation of the face fluxes leads to a numerical scheme that conserves momentum and total energy.

2.7 The semi-discrete evolution equations

We give in this section the summary of the semi-discrete evolution equations that constitute a closed set of equations for the unknowns $(\frac{1}{\rho_c}, \mathbf{U}_c, E_c)$:

$$m_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) - \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c \cdot \boldsymbol{\kappa}_p = 0, \quad (46a)$$

$$m_c \frac{d}{dt} (\mathbf{U}_c) + \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c P_{pcf}^* = \mathbf{0}, \quad (46b)$$

$$m_c \frac{d}{dt} (E_c) + \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c \cdot P_{pcf}^* \boldsymbol{\kappa}_p = 0. \quad (46c)$$

We recall that the point velocity $\boldsymbol{\kappa}_p$ and the point pressure P_{pcf}^* are written

$$\boldsymbol{\kappa}_p = \mathbf{M}_p^{-1} \sum_{c \in \mathcal{C}(p)} \sum_{f \in \mathcal{F}_p(cp)} [S_f P_c \mathbf{N}_f^c + \mathbf{M}_{pcf} \mathbf{U}_c], \quad (47a)$$

$$P_{pcf}^* = P_c - Z_c(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c, \quad \text{for } f \in \mathcal{F}_p(cp), \quad (47b)$$

where the $d \times d$ matrices \mathbf{M}_{pcf} and \mathbf{M}_p are defined by (34). The motion of the mesh is ruled by the semi-discrete trajectory equation

$$\frac{d}{dt} \mathbf{X}_p = \boldsymbol{\kappa}_p, \quad \mathbf{X}_p(0) = \mathbf{x}_p.$$

Comment 4. *In the Lagrangian formalism we have to consider two types of boundary conditions on the border of the domain \mathcal{D} : either the pressure is prescribed or the normal component of the velocity. Here, we do not detail the implementation of these boundary conditions. Let us notice that they are consistent with our nodal solver. For a detailed presentation about this topic the reader can refer to [22].*

2.8 Entropy inequality for the semi-discrete scheme

We show that our scheme in its semi-discrete form satisfies a local entropy inequality. We compute the time variation of the specific entropy σ_c in cell c using the Gibbs formula

$$m_c T_c \frac{d\sigma_c}{dt} = m_c \left[\frac{d\varepsilon_c}{dt} + P_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) \right], \quad (48)$$

where T_c denotes the mean temperature of the cell. Thanks to the definition of the internal energy this equation is rewritten

$$m_c T_c \frac{d\sigma_c}{dt} = m_c \left[\frac{dE_c}{dt} - \mathbf{U}_c \cdot \frac{d\mathbf{U}_c}{dt} + P_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) \right].$$

We dot-multiply equation (46b) and subtract it from the total energy equation (46c) and we get

$$m_c \left[\frac{dE_c}{dt} - \mathbf{U}_c \cdot \frac{d\mathbf{U}_c}{dt} \right] = -\frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c P_{pcf}^* \cdot (\boldsymbol{\kappa}_p - \mathbf{U}_c).$$

The pressure work is computed by multiplying (46b) by P_c and it is written

$$\begin{aligned} P_c \frac{d}{dt} \left(\frac{1}{\rho_c} \right) &= \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c P_c \cdot \boldsymbol{\kappa}_p \\ &= \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c P_c \cdot (\boldsymbol{\kappa}_p - \mathbf{U}_c). \end{aligned}$$

The last line of the previous equation comes from the fact that for a closed polyhedron we have

$$\sum_{f \in \mathcal{F}(c)} S_f \mathbf{N}_f^c = \mathbf{0}.$$

Finally, the combination of the previous results yields

$$m_c T_c \frac{d\sigma_c}{dt} = \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f \mathbf{N}_f^c (P_c - P_{pcf}^*) \cdot (\boldsymbol{\kappa}_p - \mathbf{U}_c). \quad (49)$$

Now, using the definition of the pressure flux (47b) we get

$$m_c T_c \frac{d\sigma_c}{dt} = \frac{1}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f Z_c [(\boldsymbol{\kappa}_p - \mathbf{U}_c) \cdot \mathbf{N}_f^c]^2.$$

This equation represents a local entropy inequality for the semi-discrete scheme since its right-hand side is always positive.

Comment 5. *The entropy production of our semi-discrete centered scheme has a structure very similar to the artificial viscosity term used in staggered scheme [9]. But, we must admit that our entropy production term is always active even in the case of isentropic flows. For such flows our scheme does not conserve entropy. This property is typical from Godunov-type schemes. However, this extra entropy production can be dramatically decreased by using a second order extension of the scheme.*

3 Spatial second order extension

The spatial second order extension is obtained by a piecewise linear monotonic reconstruction of the pressure and velocity, given by their mean values over mesh cells [4, 3].

3.1 Piecewise linear reconstruction

Let $u \equiv u(\mathbf{X})$ denotes a fluid variable (pressure or velocity components), we assume a linear variation for u in cell c

$$u_c(\mathbf{X}) = u_c + \nabla u_c \cdot (\mathbf{X} - \mathbf{X}_c). \quad (50)$$

Here, u_c is the mean value of u in cell c and ∇u_c is the gradient of u that we are looking for.

We note that $\mathbf{X}_c = \frac{1}{V_c} \int_{\Omega_c} \mathbf{X} dV$ is the cell centroid so that the reconstruction is conservative.

The gradient in (50) is computed by imposing that

$$u_c(\mathbf{X}_{c'}) = u_{c'} \quad \text{for } c' \in \mathcal{C}(c),$$

where $\mathcal{C}(c)$ is the set of neighboring cells of cell c . This problem is generally overdetermined and thus the gradient is obtained by using a least squares procedure. Hence, it is the solution of the following minimization problem

$$\nabla u_c = \operatorname{argmin} \sum_{c' \in \mathcal{C}(c)} [u_{c'} - u_c - \nabla u_c \cdot (\mathbf{X}_{c'} - \mathbf{X}_c)]^2.$$

A straightforward computation shows that this solution is written

$$\nabla u_c = \mathbf{M}_c^{-1} \sum_{c' \in \mathcal{C}(c)} (u_{c'} - u_c) (\mathbf{X}_{c'} - \mathbf{X}_c), \quad (51)$$

where \mathbf{M}_c is the $d \times d$ matrix given by

$$\mathbf{M}_c = \sum_{c' \in \mathcal{C}(c)} (\mathbf{X}_{c'} - \mathbf{X}_c) \otimes (\mathbf{X}_{c'} - \mathbf{X}_c),$$

we notice that \mathbf{M}_c is symmetric positive definite and thus always invertible. The main feature of this least squares procedure is that it is valid for any type of unstructured mesh and moreover it preserves the linear fields.

3.2 Monotonicity

To preserve monotonicity, we limit the value that the gradient is allowed to take, using the Barth Jespersen multidimensional extension [4] of the van Leer's classical method. For each cell, we introduce the slope limiter $\phi_c \in [0, 1]$ and the limited reconstructed field

$$u_c^{\text{lim}}(\mathbf{X}) = u_c + \phi_c \nabla u_c \cdot (\mathbf{X} - \mathbf{X}_c), \quad (52)$$

where ∇u_c denotes the approximated gradient given by (51). The coefficient ϕ_c is determined by enforcing the following local monotonicity criterion

$$u_c^{\min} \leq u_c^{\text{lim}}(\mathbf{X}) \leq u_c^{\max}, \forall \mathbf{X} \in c. \quad (53)$$

Here, we have set $u_c^{\min} = \min(\min_{c' \in \mathcal{C}(c)}, u_c)$ and $u_c^{\max} = \max(\max_{c' \in \mathcal{C}(c)}, u_c)$. Since the reconstructed field is linear we note that it is sufficient to enforce the following conditions at the points

$$u_c^{\min} \leq u_c^{\text{lim}}(\mathbf{X}_p) \leq u_c^{\max}, \forall p \in \mathcal{P}(c). \quad (54)$$

so that the quantity u in the cell c does not lie outside the range of the average quantities in the neighboring cells. Thanks to this formula we can define the slope limiter as

$$\phi_c = \min_{p \in \mathcal{P}(c)} \phi_{c,p}$$

knowing that

$$\phi_{c,p} = \begin{cases} \mu\left(\frac{u_c^{\max} - u_c}{u_c(\mathbf{X}_p) - u_c}\right) & \text{if } u_c(\mathbf{X}_p) - u_c > 0, \\ \mu\left(\frac{u_c^{\min} - u_c}{u_c(\mathbf{X}_p) - u_c}\right) & \text{if } u_c(\mathbf{X}_p) - u_c < 0, \\ 1 & \text{if } u_c(\mathbf{X}_p) - u_c = 0. \end{cases}$$

Here, μ denotes a real function that characterizes the limiter. By setting $\mu(x) = \min(1, x)$ we recover the Barth Jespersen limiter. We can also define a smoother -in the sense that it is more differentiable- limiter by setting $\mu(x) = \frac{x^2 + 2x}{x^2 + x + 2}$. This limiter has been introduced by Vankatakrishnan [31] in order to improve the convergence towards steady solutions for the Euler equations.

Finally, instead of using the mean values of the pressure and the velocity in our nodal solver, we use their nodal extrapolated values deduced from the linear monotonic reconstruction.

4 Time discretization

For the time discretization of the semi-discrete evolution equations (46a)-(46c) we use a second order Runge-Kutta scheme. We assume to know the physical properties in the cell c and its geometrical characteristics at the beginning of the time step t^n , i.e., $\rho_c^n, \mathbf{U}_c^n, E_c^n, P_c^n$ and \mathbf{X}_p^n , for $p \in \mathcal{P}(c)$. We want to compute their values at time t^{n+1} and we introduce the time step $\Delta t = t^{n+1} - t^n$. We describe hereafter the predictor and the corrector steps of the second order Runge-Kutta time discretization.

4.1 Predictor step

We start with the following predictor step. We denote with the superscript $n+1, 1$ the values at the end of this predictor step.

- We compute the nodal values with the nodal solver: knowing the physical variables and the geometry at time t^n we compute the nodal velocity κ_p^n by solving the linear system (33), then we deduce the nodal pressure related to face f and cell c , $P_{pcf}^{\star, n}$, thanks to the equation (32).

- point motion: we advance the points position using the trajectory equation

$$\mathbf{X}_p^{n+1,1} = \mathbf{X}_p^n + \Delta t \boldsymbol{\kappa}_p^n$$

.

- We update the geometry and density: knowing $\mathbf{X}_p^{n+1,1}$ we compute the geometry and particularly the volume of the cell $V_c^{n+1,1}$ and deduce from it the new density $\rho_c^{n+1,1}$ from mass conservation.
- We update the momentum and the total energy by solving

$$m_c(\mathbf{U}_c^{n+1,1} - \mathbf{U}_c^n) + \frac{\Delta t}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f^n \mathbf{N}_f^{c,n} P_{pcf}^{*,n} = \mathbf{0},$$

$$m_c(E_c^{n+1,1} - E_c^n) + \frac{\Delta t}{d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} S_f^n \mathbf{N}_f^{c,n} \cdot P_{pcf}^{*,n} \boldsymbol{\kappa}_p^n = 0.$$

- We update the specific internal energy and the pressure

$$\varepsilon_c^{n+1,1} = E_c^{n+1,1} - \frac{1}{2} \|\mathbf{U}_c^{n+1,1}\|^2,$$

$$P_c^{n+1,1} = P(\rho_c^{n+1,1}, \varepsilon_c^{n+1,1}).$$

4.2 Corrector step

We use the predict values in order to complete the time discretization

- We compute the nodal values with the nodal solver: knowing the physical variables and the geometry at the end of the predictor step we compute the nodal velocity $\boldsymbol{\kappa}_p^{n+1,1}$ by solving the linear system (33), then we deduce the nodal pressure related to face f and cell c , $P_{pcf}^{*,n+1,1}$, thanks to the equation (32).
- point motion: we advance the points position using the trajectory equation with the centered velocity

$$\mathbf{X}_p^{n+1} = \mathbf{X}_p^n + \frac{\Delta t}{2} (\boldsymbol{\kappa}_p^n + \boldsymbol{\kappa}_p^{n+1,1})$$

.

- We update the geometry and density: knowing \mathbf{X}_p^{n+1} we compute the geometry and particularly the volume of the cell V_c^{n+1} and deduce from it the new density ρ_c^{n+1} from mass conservation.
- We update the momentum and the total energy by using the centered fluxes

$$m_c(\mathbf{U}_c^{n+1} - \mathbf{U}_c^n) + \frac{\Delta t}{2d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} (S_f^n \mathbf{N}_f^{c,n} P_{pcf}^{*,n} + S_f^{n+1,1} \mathbf{N}_f^{c,n+1,1} P_{pcf}^{*,n+1,1}) = \mathbf{0},$$

$$m_c(E_c^{n+1} - E_c^n) + \frac{\Delta t}{2d} \sum_{f \in \mathcal{F}(c)} \sum_{p \in \mathcal{P}(f)} (S_f^n \mathbf{N}_f^{c,n} \cdot P_{pcf}^{*,n} \boldsymbol{\kappa}_p^n + S_f^{n+1,1} \mathbf{N}_f^{c,n+1,1} \cdot P_{pcf}^{*,n+1,1} \boldsymbol{\kappa}_p^{n+1,1}) = 0.$$

- We update the specific internal energy and the pressure

$$\varepsilon_c^{n+1} = E_c^{n+1} - \frac{1}{2} \|\mathbf{U}_c^{n+1}\|^2,$$

$$P_c^{n+1} = P(\rho_c^{n+1}, \varepsilon_c^{n+1}).$$

Comment 6. We note that we could also use the time discretization of the GCL (46a) to compute the updated value of the density. However, since the geometrical part of the volume variation flux is a quadratic function of time, one needs to perform the time integration exactly in order to ensure the consistency between the discretized GCL and the cell volume variation [23, 24].

Comment 7. The geometrical part of the momentum and total energy fluxes is discretized in an explicit manner in order to preserve the compatibility with the nodal solver. Thus, the scheme conserves exactly momentum and total energy.

4.3 Time step limitation

The time step is evaluated following two criteria. The first one is a standard CFL criterion which guaranties heuristically the monotone behavior of the entropy. The second is more intuitive, but reveals very useful in practice: we limit the variation of the volume of cells over one time step.

4.3.1 CFL criterion

We propose a CFL like criterion in order to ensure a positive entropy production in cell c during the time step. At time t_n , for each cell c we denote by λ_c^n the minimal value of the distance between two points of the cell. We define

$$\Delta t_E = C_E \min_c \frac{\lambda_c^n}{a_c^n},$$

where C_E is a strictly positive coefficient and a_c is the sound speed in the cell. The coefficient C_E is computed heuristically and we provide no rigorous analysis which allows such formula. However, extensive numerical experiments show that $C_E = 0.25$ is a value which provides stable numerical results. We have also checked that this value is compatible with a monotone behavior of entropy. The rigorous derivation of this criterion could be obtained by computing the time step which ensures a positive entropy production in cell c from time t^n to t^{n+1} .

4.3.2 Criterion on the variation of volume

We estimate the volume of the cell c at $t = t^{n+1}$ with the Taylor expansion

$$V_c^{n+1} = V_c^n + \frac{d}{dt} V_c(t^n) \Delta t.$$

Here, the time derivative $\frac{d}{dt} V_c$ is computed by using (11). Let C_V be a strictly positive coefficient, $C_V \in]0, 1[$. We look for Δt such that

$$\frac{|V_c^{n+1} - V_c^n|}{V_c^n} \leq C_V.$$

To do so, we define

$$\Delta t_V = C_V \min_c \left\{ \frac{V_c^n}{\left| \frac{d}{dt} V_c(t^n) \right|} \right\}.$$

For numerical applications, we choose $C_V = 0.1$.

Last, the estimation of the next time step Δt^{n+1} is given by

$$\Delta t^{n+1} = \min(\Delta t_E, \Delta t_V, C_M \Delta t^n), \quad (55)$$

where Δt^n is the current time step and C_M is a multiplicative coefficient which allows the time step to increase. We generally set $C_M = 1.01$.

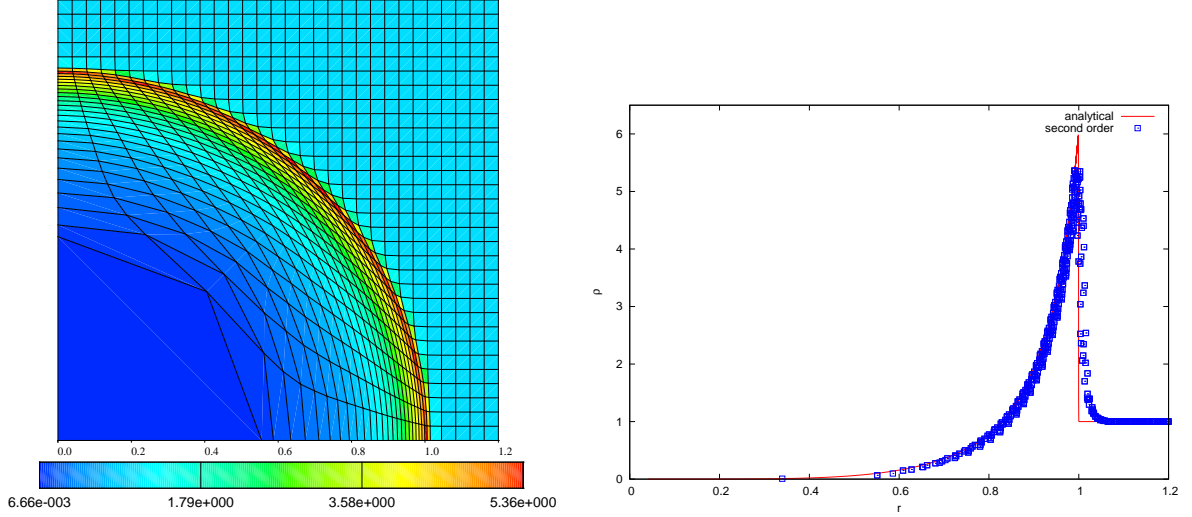


Figure 5: 2D Sedov problem on a Cartesian grid: density map (left) and density in all the cells (right) at $t = 1$.

5 Numerical results

In this section, we present several test cases in order to validate our numerical scheme. For each problems, we use a perfect gas equation of state which is taken to be of the form $P = (\gamma - 1)\rho\varepsilon$, where γ is the polytropic index.

5.1 2D Sedov problem

This test case describes the evolution of a blast wave in a point symmetric explosion for a gas characterized by $\gamma = \frac{7}{5}$. An exact solution with cylindrical symmetry is derived with self-similarity arguments in [20]. The initial density has a uniform unit distribution, and, the pressure is 10^{-6} everywhere, except in the cell containing the origin. For this cell we define $P = (\gamma - 1)\rho\frac{\mathcal{E}_0}{V}$ where $\mathcal{E}_0 = 0.244816$ is the total amount of released energy. For this value, it is shown in [20] that the exact solution is a cylindrically symmetric diverging shock whose front is at radius $r = \sqrt{x^2 + y^2} = 1$ and has a peak density of 6.

First, we perform a computation on the $[0, 1.2] \times [0, 1.2]$ quadrant, subdivided into 30×30 squares. The results obtained in Figure 5 are quite good and they assess the ability of the method to respect the cylindrical symmetry. We notice that these results are very close to those obtained in [21] for the same setup but using a staggered scheme. For the same problem, we have also displayed in Figure 6 the results obtained on a polygonal grid which is defined in [21]. Once more the result shows a good agreement with the analytical solution. The numerical method preserves very well the one-dimensional cylindrical solution. This demonstrates the ability of our solver to handle unstructured mesh.

5.2 2D Noh problem

This test case has been introduced by Noh in [25]. A gas with $\gamma = \frac{5}{3}$ is given an initial unit inward velocity. A circular shock wave is generated which at time $t = 0.6$ has a radius of 0.2. The initial thermodynamic state is given by $(\rho, P) = (1, 10^{-6})$. The initial domain is defined by $[R, \theta] \in [0; 1] \times [0, \frac{\pi}{2}]$ where the polar coordinates are given by $r = \sqrt{x^2 + y^2}$ and $\theta = \arctan(\frac{y}{x})$.

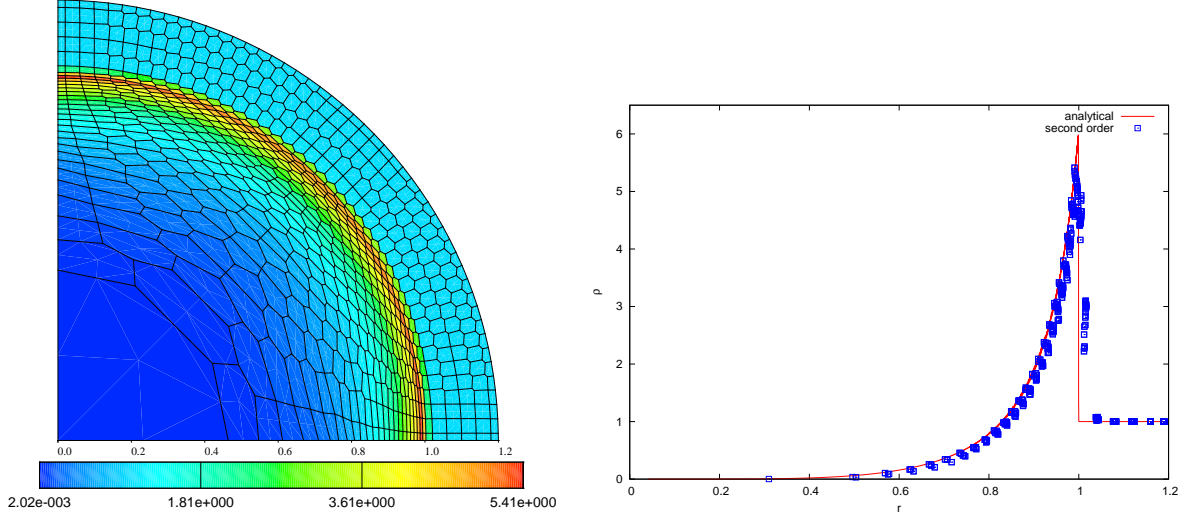


Figure 6: 2D Sedov problem on a polygonal grid: density map (left) and density in all the cells (right) at $t = 1$.

We use an non-conformal mesh with two levels of refinement. This non-conformal grid contains a mixture of triangles, quadrangles and pentagons as it can be seen in Figure 7. In Figure 7, we observe the good quality of the mesh after shock reflection and the good agreement with the analytical solution for the density profile. These numerical results show the ability of our Lagrangian scheme to handle non-conformal grid without any specific modifications.

5.3 3D Saltzmann problem

We consider now the motion of a planar shock wave on a Cartesian grid that has been skewed. This test case has been initially defined for two-dimensional flows in [15]. This is a well known difficult test case that enables to validate the robustness of a Lagrangian scheme when the mesh is not aligned with the fluid flow. Here, we consider the three-dimensional extension of this test that has been proposed in [11]. The computational domain is the volume defined by $(x, y, z) \in [0, 1] \times [0, 0.1] \times [0, 0.1]$. The initial mesh is obtained by transforming a uniform $100 \times 10 \times 10$ Cartesian grid with the mapping

$$\begin{aligned} x_{\text{sk}} &= x + (0.1 - z)(1 - 20y) \sin(x\pi), \quad \text{for } 0 \leq y \leq 0.05 \\ x_{\text{sk}} &= x + z(20y - 1) \sin(x\pi), \quad \text{for } 0.05 \leq y \leq 0.1 \\ y_{\text{sk}} &= y, \\ z_{\text{sk}} &= z. \end{aligned}$$

We notice that this skewed grid is based on generalizing the two-dimensional Saltzmann grid in the following manner: the $y = 0$ surface is the original two-dimensional skewed grid, this grid is additionally skewed with respect to y coordinate. We note that the $y = 0.05$ surface is not skewed at all. For this problem we use the polytropic index ($\gamma = 5/3$). The initial state is $(\rho^0, P^0, \mathbf{V}^0) = (1, 10^{-6}, 0)$. At the plane $x = 0$, a unit normal velocity is specified. On all the other boundaries, we set reflective boundary conditions. The exact solution is a planar shock wave that moves at speed $4/3$ in the x direction. The shock wave hits the face $x = 1$ at time $t = 0.75$, the density should be equal to 4 in the shocked region.

We perform two computations with our three-dimensional scheme. The first one is done by using a 3D grid which is 2D skewed, that is, we set only one cell in the y direction and rewrite

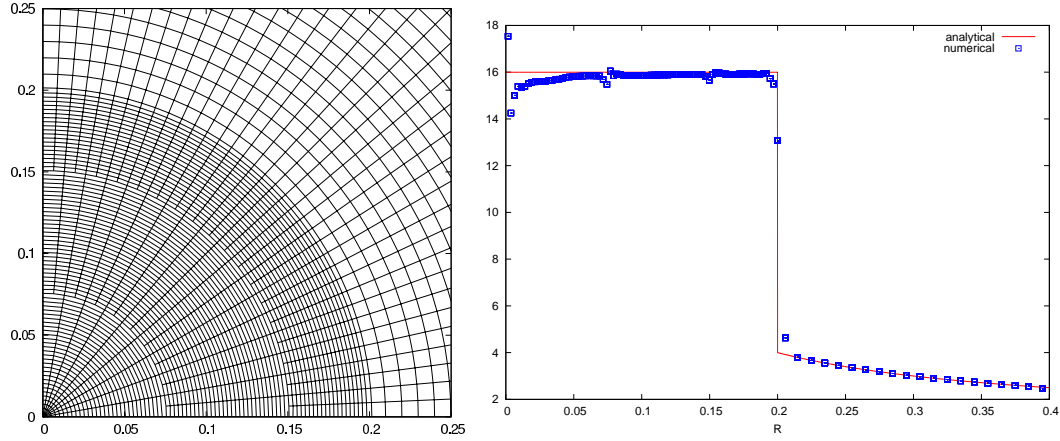


Figure 7: 2D Noh problem on a non-conformal grid: grid (left) and density in all the cells (right) at $t = 0.6$.

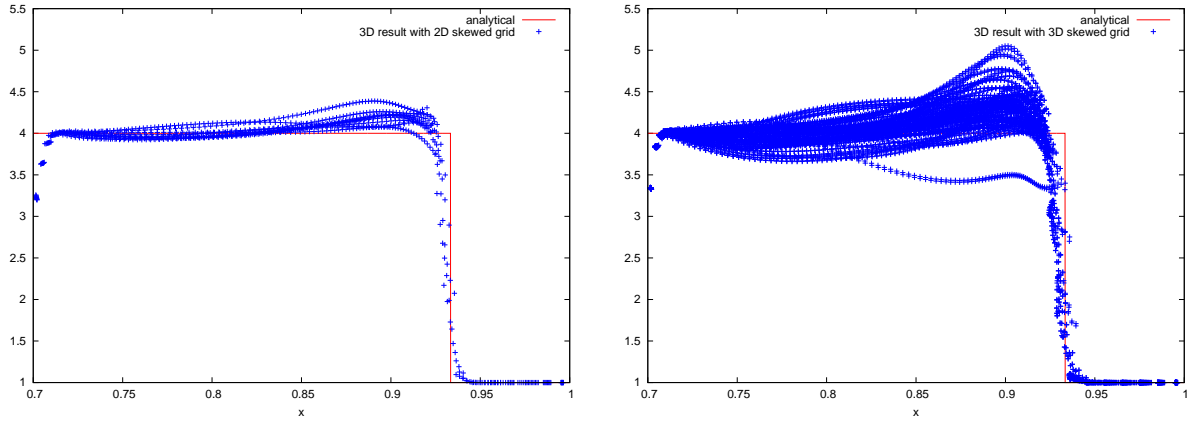


Figure 8: 3D Saltzman problem: density in all the cells at time $t = 0.7$ for 2D (left) and 3D (right) deformation.

$x_{sk} = x + (0.1 - z) \sin(x\pi)$. This problem is a sanity check that corresponds to the 2D Saltzman problem. The grid at time $t = 0.7$ is displayed in Figure 9 (left). We notice that our scheme preserves very well the one-dimensional solution. Moreover, as it can be seen in Figure 8 (left), the location of the shock wave and shock plateau are in good agreement with the analytical solution. These results demonstrate that our 3D scheme behaves similarly with the 2D scheme derived in [22]. The second computation is performed by using the 3D skewed grid defined previously. The grid at time $t = 0.7$ is shown in Figure 9 (right). The grid over which the shock has propagated is less skewed than the original one. However, we note that some distortions have occurred in the vicinity of the shock front. These distortions will lead to a breakdown of the computation after the first bounce at time $t = 0.78$. After this time the computation stops because the time step becomes too small. This too small time step occurs when two points are too close to each other. The plot of the density profile in Figure 8 (right) assesses that the one-dimensional solution is quite well preserved. The shock level is not uniform but it oscillates slightly around the exact value.

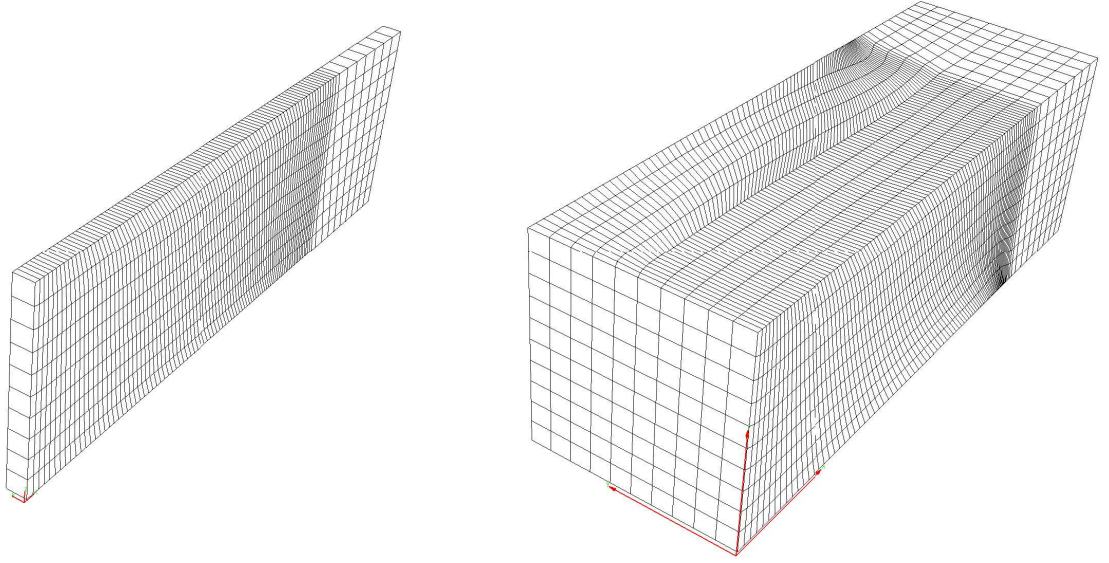


Figure 9: 3D Saltzmann problem: mesh at time $t = 0.7$ for 2D (left) and 3D (right) deformation.

5.4 3D Sedov problem

As it is said in [30], the Sedov test in three dimensions is much more challenging than its two-dimensional counterpart, due to the increased distortion undergone by the elements. Here, we are going to compute the Sedov blast test on the $[0, 1.2] \times [0, 1.2] \times [0, 1.2]$ octant. We use two grids made of $20 \times 20 \times 20$ and $40 \times 40 \times 40$ cubes. The initial conditions are the same than in the 2D Sedov test. The total amount of energy released in the cell containing the origin is computed so that the shock wave reaches the radius $r = 1$ at time $t = 1$. Following [20] we set $\mathcal{E}_0 = 0.106384$. We successfully run to completion computations for both grids with such an initialization. The density in all the cells as a function of the radius of the center of the cell is displayed in Figure 10. We note a very good agreement with the analytical solution. Moreover, for the coarser grid our result seems to reach the same level of accuracy than the one obtained in [30] with almost the same grid. For the finer grid we observe the convergence of the numerical solution toward the analytical one. The grid and the density contour are displayed in perspective view in Figure 11 for the coarse grid and in Figure 12 for the fine grid.

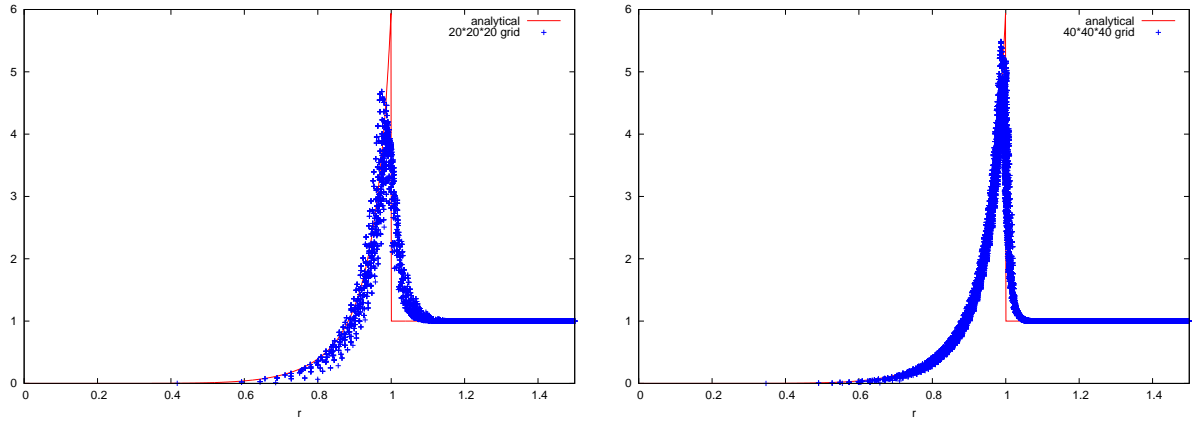


Figure 10: 3D Sedov problem: density in all the cells at time $t = 1$ as a function of the radius of the cell center, $20 \times 20 \times 20$ mesh (left) and $40 \times 40 \times 40$ mesh (right).

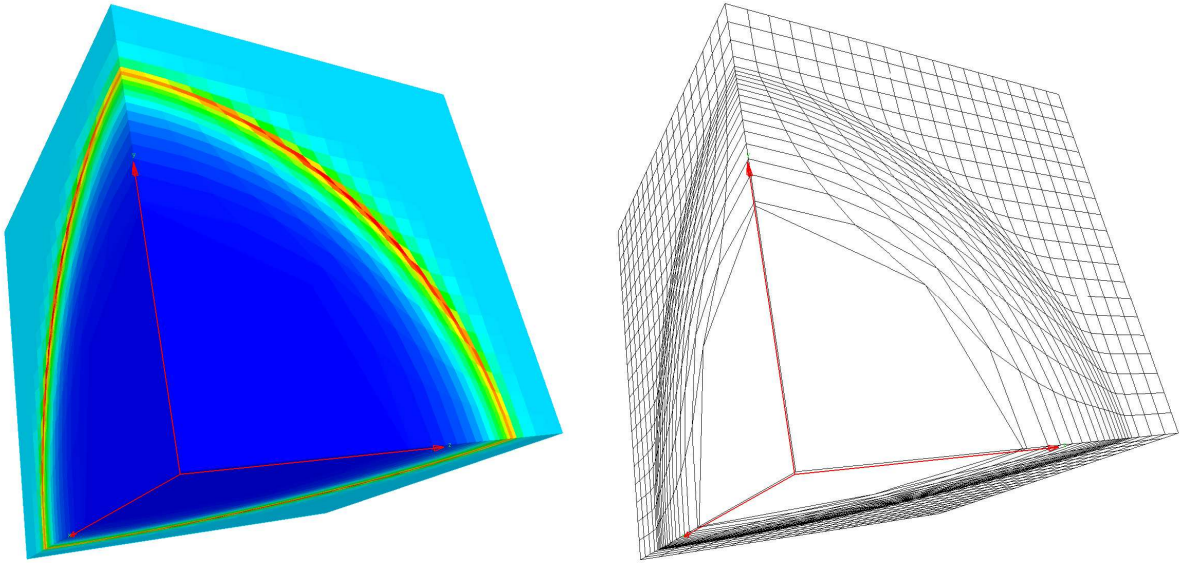


Figure 11: 3D Sedov problem: density contour (left) and mesh (right) at time $t = 1$ for the $20 \times 20 \times 20$ mesh

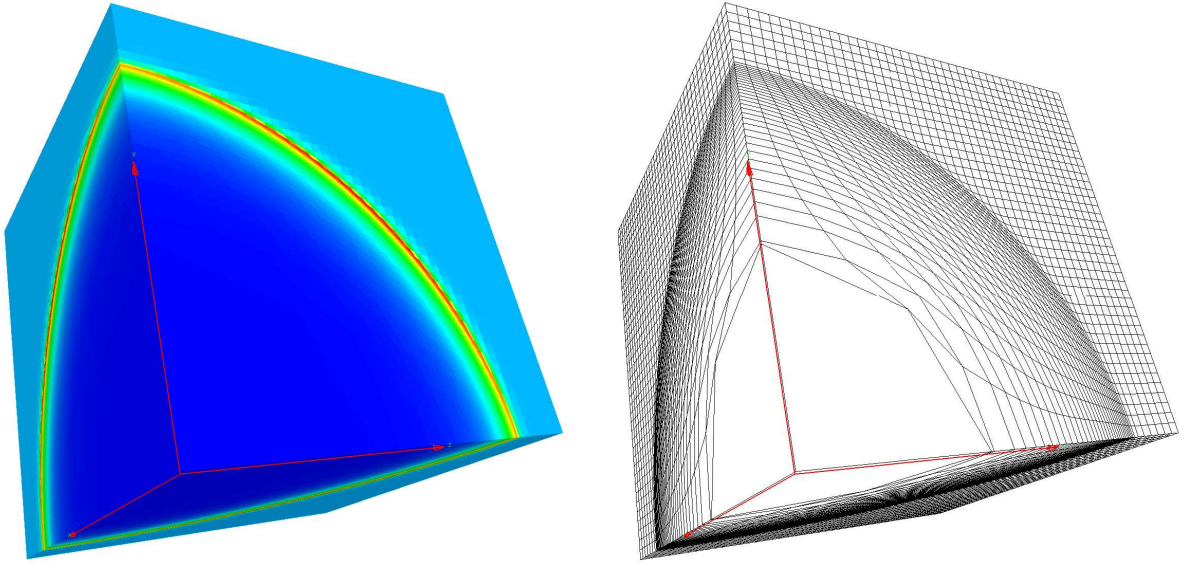


Figure 12: 3D Sedov problem: density contour (left) and mesh (right) at time $t = 1$ for the $40 \times 40 \times 40$ mesh

6 Conclusion

In this paper, we have presented a multidimensional finite volume scheme for Lagrangian hydrodynamics. We have shown how to derive the discretization of the Geometric Conservation Law (GCL) consistently with the nodes displacement. Then, to ensure consistency with the GCL discretization and continuity of the kinematic velocity around each node, we have introduced $2d$ face pressures located at each node for each simplicial face. These pressures are linked to the kinematic velocity thanks to half-Riemann problems around each node. The local balance of momentum around each node allows to compute the kinematic velocity by solving a local $d \times d$ system and then deduce from it the nodal pressures. Finally, by using the previous quantities we design linear mappings over each simplicial faces that provides a consistent approximation of the face fluxes. In this framework, we have derived a scheme that ensures momentum and total energy conservation and satisfies a local entropy inequality. Numerical results obtained, in 2D and 3D, with this scheme are in good agreement with the analytical solutions of the proposed test cases.

In the future, we intend to investigate improvements of the nodal Riemann solver formulation, particularly its dependency to the geometry of mesh. This point is crucial since it is related to the spatial distribution of entropy production.

Acknowledgment

This work was performed under the auspices of “L’Agence Nationale de la Recherche” under contract ANR-06-CIS6-009.

References

- [1] R. Abgrall, R. Loubère, and J. Ovadia. A Lagrangian Discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems. *Int. J. Numer. Meth. Fluids*, 44:645–663, 2004.
- [2] F. L. Adessio, D. E. Carroll, K. K. Dukowicz, J. N. Johnson, B. A. Kashiwa, M. E. Maltrud, and H. M. Ruppel. Caveat: a computer code for fluid dynamics problems with large distortion and internal slip. Technical Report LA-10613-MS, Los Alamos National Laboratory, 1986.
- [3] T. J. Barth. Numerical methods for conservation laws on structured and unstructured meshes. Technical report, VKI Lecture Series, 2003.
- [4] T. J. Barth and 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.
- [5] D. J. Benson. Computational methods in Lagrangian and Eulerian hydrocodes. *Comp. Meth. in Appl. Mech. and Eng.*, 99:235–394, 1992.
- [6] D.E. Burton. Multidimensional Discretization of Conservation Laws for Unstructured Polyhedral Grids. Technical Report UCRL-JC-118306, Lawrence Livermore National Laboratory, 1994.
- [7] J.C. Campbell and M.J. Shashov. A tensor artificial viscosity using a mimetic finite difference algorithm. *J. Comput. Phys.*, 172(4):739–765, 2001.
- [8] J.C. Campbell and M.J. Shashov. A compatible Lagrangian hydrodynamics algorithm for unstructured grids. *Selçuk J. Appl. Math.*, 4:53–70, 2003.

- [9] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of Artificial Viscosity for Multidimensional Shock Wave Computations. *J. Comput. Phys.*, 144:70–97, 1998.
- [10] E.J. Caramana, D.E. Burton, M.J. Shashov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, 1998.
- [11] E.J. Caramana, C.L. Rousculp, and D.E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *J. Comput. Phys.*, 157:89–119, 2000.
- [12] E.J. Caramana and M.J. Shashkov. Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures. *J. Comput. Phys.*, 142:521–561, 1998.
- [13] B. Després and C. Mazeran. Lagrangian gas dynamics in two dimensions and lagrangian systems. *Arch. Rational Mech. Anal.*, 178:327–372, 2005.
- [14] J. K. Dukowicz. A general, non-iterative Riemann solver for Godunov’s method. *J. Comput. Phys.*, 61:119–137, 1984.
- [15] J. K. Dukowicz and B. Meltz. Vorticity errors in multidimensional Lagrangian codes. *J. Comput. Phys.*, 99:115–134, 1992.
- [16] J.K. Dukowicz, M.C. Cline, and F.S. Adressio. A general topology method. *J. Comput. Phys.*, 82:29–63, 1989.
- [17] S.K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, and G. Prokopov. *Résolution numérique des problèmes multidimensionnels de la dynamique des gaz*. Mir, 1979.
- [18] C.W. Hirt, A. Amsden, and J.L. Cook. An arbitrary Lagrangian–Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14:227–253, 1974.
- [19] W. H. Hui, P. Y. Li, and Z. W. Li. A Unified Coordinate System for Solving the Two-Dimensional Euler Equations. *J. Comput. Phys.*, 153:596–637, 1999.
- [20] J.R. Kamm and F.X. Timmes. On efficient Generation of Numerically Robust Sedov Solutions. Technical Report LA-UR-07-2849, Los Alamos National Laboratory, 2007.
- [21] R. Loubère and M.J. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary-lagrangian-eulerian methods. *J. Comput. Phys.*, 23:155–160, 2004.
- [22] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for compressible flow problems. *SIAM J. Scien. Comp.*, 29(4):1781–1824, 2007.
- [23] B. Nkonga. On the conservative and accurate CFD approximations for moving meshes and moving boundaries. *Comput. Methods Appl. Mech. Eng.*, 190(13-14):1801–1825, 2000.
- [24] B. Nkonga and H. Guillard. Godunov type method on non-structured meshes for three-dimensional moving boundary problems. *Comput. Methods Appl. Mech. Eng.*, 113(1-2):183–204, 1994.
- [25] W. F. Noh. Errors for calculations of strong shocks using artificial viscosity and an artificial heat flux. *J. Comput. Phys.*, 72:78–120, 1987.
- [26] R. D. Richtmyer and K. W. Morton. *Difference methods for initial-value problems*. John Wiley, 1967.
- [27] G. Scovazzi. Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations. *Comput. Methods Appl. Mech. and Engrg.*, 196:966–978, 2007.
- [28] G. Scovazzi, M.A. Christon, T.J.R. Hughes, and J.N. Shadid. Stabilized shock hydrodynamics: I. A Lagrangian method. *Comput. Methods Appl. Mech. and Engrg.*, 196:923–966, 2007.

- [29] G. Scovazzi, E. Love, and M.J. Shashkov. Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: Theoretical framework and two-dimensional computations. *Comput. Methods Appl. Mech. and Engrg.*, 197:1056–1079, 2008.
- [30] G. Scovazzi, E.L. Love, and M. Shashkov. A multi-scale q1/p0 approach to lagrangian shock hydrodynamics. Technical Report SAND2007-1423, Sandia National Laboratories, 2007.
- [31] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J. Comput. Phys.*, 118:120–130, 1995.
- [32] J. von Neumann and R. D. Richtmyer. A method for the numerical calculations of hydrodynamical shocks. *J. Appl. Phys.*, 21:232–238, 1950.
- [33] M. L. Wilkins. Calculation of elastic plastic flow. *Methods in Computational Physics*, 3, 1964.