A node conservative cell-centered Finite Volume method for solving multidimensional Euler equations over general unstructured grids

Vincent Delmas, Raphaël Loubère, Pierre-Henri Maire

A node conservative cell-centered Finite Volume method for solving multidimensional Euler equations over general unstructured grids

Vincent Delmas[a,b], Raphaël Loubère[a], Pierre-Henri Maire[*b]

[a] *Institut de Mathématiques de Bordeaux (IMB), Université de Bordeaux, CNRS, Bordeaux INP, F33400, Talence, France*
[b] *CEA Cesta, 15 avenue des sablières, Le Barp, France*

Abstract

We are interested in the numerical simulation of hypersonic flows around vehicles characterized by complex geometry. As a first step to move in this direction, we present a robust and accurate cell-centered Finite Volume (FV) method for solving the three-dimensional compressible Euler equations over general unstructured grids. This FV approach relies on a novel positivity-preserving discretization of the multidimensional Euler equations, which leverages a partitioning of cell faces into subfaces impinging at the nodes. The subface flux approximation is derived from an approximate Riemann solver, which incorporates not only the mean values of the cells adjacent to the subface but also the velocity of the node from which the subface originates. The projection of the nodal velocity onto the unit normal vector of the subface can be interpreted as a parameter in this Riemann solver. Consequently, the resulting subface flux is not unique, leading to a lack of conservation in the classical sense. Conservation is restored by ensuring that the subface fluxes around a node sum to zero, which determines the nodal velocity. This innovative multipoint flux approximation approach seems to eliminate the numerical pathologies commonly encountered in classical face-based FV formulations. The space and time second-order extension of this FV approach is classically deduced by means of a monotonic piecewise linear reconstruction. The robustness and accuracy of this novel numerical method are assessed against various demanding representative test cases in 2D and 3D.

Keywords: Finite Volume method, Unstructured grid, 3D, Euler equations, Godunov-type methods, Approximate Riemann solver, Hypersonic flows

1. Introduction

The numerical simulation of hypersonic flows remains a domain of primary importance not only to design hypersonic flight vehicles but also to assess their aerodynamic and aerothermal characteristics [4]. This type of flows is characterized, among other things, by the presence of very strong shock and rarefaction waves. Sharp gradients of velocity and temperature are also concentrated in a very thin zone located in the vicinity of the surface and named the boundary layer. The hypersonic regime is thus particularly demanding in terms of robustness and accuracy with respect to the employed numerical methods. First of all, the numerical methods have to be able to cope with a very strong curved bow shock in front of the nose of the vehicle under consideration. This requires a sufficient amount of numerical dissipation related to the inviscid part of the flux to stabilize the shock profile. On the other hand, the high fidelity capture of the sharp temperature and velocity profiles in the boundary layer necessitates not only an adequate near wall

[*]Corresponding author
Email addresses: vincent.delmas@math.u-bordeaux.fr (Vincent Delmas), raphael.loubere@math.u-bordeaux.fr (Raphaël Loubère), pierre-henri.maire@cea.fr (Pierre-Henri Maire[*])

grid spacing but also a consistent and accurate discretization of the heat conducting and viscous fluxes of the Navier-Stokes equations [4]. In addition, one has to carefully control the numerical dissipation needed to stabilize the shock to prevent it from degrading the heat transfer in the boundary layer. This is the eternal trade-off between robustness (more dissipation) and accuracy (less dissipation) which becomes even more difficult in the context of hypersonic flows characterized by extreme gradients. Up to our knowledge, most[1] of the production codes used for intensive numerical simulations of hypersonic flows, in the continuum regime, rely on Finite Volume (FV) based numerical methods. They are at most nominally second-order accurate, refer for instance to the technical report [24], which describes the numerics of the NASA legacy code named LAURA, or to [12], which presents the numerical method underlying to the recent unstructured US3D code developed at the University of Minnesota.

Our ultimate motivation is to develop a robust and accurate cell-centered FV discretization of the three-dimensional compressible Navier-Stokes (NS) equations, specifically tailored to solving hypersonic flows around vehicles with complex geometries. The adoption of hybrid unstructured grids is indispensable for such applications, as they greatly simplify the meshing of complex configurations. However, these grids also impose significant demands on the robustness and accuracy of the numerical methods employed [56]. As a first step, we shall focus on the numerical discretization of the inviscid part of the flow which is governed by the compressible Euler equations. Namely, we aim at designing a cell-centered FV method for solving the three-dimensional compressible Euler equations over general unstructured grids composed of polyhedra (tetrahedra, hexahedra, prisms, pyramids...). Since its introduction at the beginning of the eighties, the shock capturing FV method has become the cornerstone of any modern aerodynamics code. A quite complete panorama of this classical approach might be found for instance into [57]. The cell-centered FV method consists in writing the integral formulation of the conservation laws of mass, momentum and total energy over each cell of the computational grid. The primary unknowns are the cell-averaged values of mass, momentum and total energy. For a given cell, the time increment of the cell-averaged value results from the summation of the normal fluxes located at each cell face. In this classical framework, the normal flux approximation is obtained in a unique manner at each face through the use of an approximate Riemann solver such as the Roe one [49], which is probably one of the most popular among the available ones [57]. Even if this type of method is characterized by a positive return of experience it is also admitted that it is plagued by several flaws such as the so-called odd-even decoupling and carbuncle numerical instabilities, refer to [46, 48]. There is thus still a strong need to design robust and accurate FV methods able to cope with production engineering simulations of hypersonic flows on any type of grids. As quoted by Candler and his co-authors [11]:

> There remain a number of outstanding numerical issues in the simulation of hypersonic flows. A particular difficulty is associated with the simulation of high Mach number blunt capsule geometries that have a very large region of subsonic flow near the stagnation point. This class of flow magnifies numerical error generated at the strong shock wave; this error then accumulates in the stagnation region and corrupts the solution. The main remedies for this problems are: The grid must be aligned with the bow shock; Eigenvalue limiters must be used judiciously; other forms of dissipation can also be used to counter-act the error generated by the strong gradients across the bow shock. None of these fixes actually solve the underlying problem, rather they reduce its magnitude and mask its effects with additional dissipation. Clearly, fundamental work needs to be done to reduce the sensitivity of the solution to the grid and specific details of the numerical method.

The present work is an attempt to address the foregoing issues. To this end, we construct the three-dimensional extension of the subface flux-based FV method introduced in [22] for solving the 2D compressible Euler equations on unstructured grids. This peculiar numerical method relies on the partition

[1] At the exception of the Dassault Aviation's Navier-Stoke code, named AETHER, which relies on an original Finite Element approach based on a symmetric form of the equations written in terms of entropy variables, refer to [15, 14].

of the cell interfaces into subfaces. The subface flux numerical approximation stems from the notion of simple Riemann solver introduced in the seminal work [21]. The approximate Riemann solver utilized in this work is constructed by decomposing the intermediate fluxes into a convection part plus a pressure part. This particular splitting of the numerical flux allows to define the wave speeds of this Riemann solver in a very natural manner contrarily to what is done for the HLLC solver [57]. Moreover, it is worth noting that this decomposition of the intermediate flux implies that the present Riemann solver has an underlying Lagrangian structure through a discrete Lagrange-to-Euler mapping introduced in [21]. This fundamental feature ensures the transfer of good properties such as positivity preservation and entropy stability [16]. The subface flux computation relies not only on the mean values of the cells located on both sides of the subface by means of the foregoing approximate Riemann solver but also on the velocity of the node from which the subface emanates. The normal component of this nodal velocity with respect to the unit normal of the subface might be viewed as a parameter in the aforementioned approximate Riemann solver. Thus, the interface flux is not uniquely defined and the FV method is not conservative anymore in the classical sense. Namely, the underlying approximate Riemann solver does not fulfill anymore the famous HLL consistency condition with the integral form of the conservation laws, refer to [27, 28]. Nonetheless, the conservation property is retrieved thanks to a local balance at each node written under the form of a vectorial equation, the solution of which provides the expression of the nodal velocity. This nodal balance equation might be interpreted as a multidimensional consistency condition with the integral form of the system of conservation laws written over the dual cell. This interpretation reveals the link of the present approach with the Residual Distribution method [1] which also relies on this type of consistency condition. Finally, we emphasize that the resulting numerical method is able to guarantee the positivity of mass density and specific energy while ensuring entropy stability provided an explicit time step condition is satisfied. We have thus obtained a novel multidimensional conservative and entropy-stable FV scheme wherein the numerical fluxes are computed through a nodal solver, which is exactly the one designed for compressible Lagrangian hydrodynamics [41]. The robustness and the accuracy of this novel FV scheme are assessed through various numerical tests run on various unstructured grids. This novel FV methods seems to be insensitive to the numerical pathologies (odd-even decoupling, carbuncle) occurring when utilizing the classical shock capturing methods such as the Roe or HLLC schemes.

The rest of the paper is organized as follows. After recalling the multidimensional Euler equations and the associated Riemann problem, we construct an original subface-based Riemann solver which does not fulfill the HLL condition. Then, we construct the flux approximation and the multipoint FV scheme induced by this Riemann solver and we show how to restore conservation in a node-based sense by means of a nodal solver. We also describe the two-point FV scheme induced by the Riemann solver satisfying the HLL condition for which the conservation is classically face-based. We propose also a variant of this Riemann solver by adding a controlled amount of numerical dissipation proportional to the jump between the tangential components of the velocity to increase its robustness. For the sake of completness, we also describe the boundary conditions discretization and the space/time second-order extension. Finally, we assess the robustness and the accuracy of these FV methods against various demanding and representative test cases.

2. Multidimensional Euler equations

2.1. Governing equations and main properties

Inviscid and non heat conducting multidimensional fluid flows are governed by the compressible Euler equations expressing the conservation of mass, momentum and total energy [40]. The corresponding system of conservation laws writes under the compact form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbb{F}(\mathbf{U}) = \mathbf{0}. \tag{1}$$

Here, $\mathbf{U} = \mathbf{U}(\mathbf{x}, t)$, where $\mathbf{x} \in \mathbb{R}^{\mathrm{d}}$, is the vector of conservative variables and d denotes the space dimension. This vector of size d+2 writes $\mathbf{U} = (\rho, \rho\mathbf{v}, \rho e)^t$ where ρ is the mass density, $\mathbf{v} \in \mathbb{R}^{\mathrm{d}}$ the velocity vector and e the specific total energy. The physical flux is the tensor

$$\mathbb{F}(\mathbf{U}) = \begin{pmatrix} \rho\mathbf{v}^t \\ \rho\mathbf{v} \otimes \mathbf{v} + p\mathbb{I}_d \\ \rho e\mathbf{v}^t + p\mathbf{v}^t \end{pmatrix},$$

where p denotes the thermodynamic pressure and $\mathbb{I}_d$ the d-dimensional identity matrix. The specific internal energy is given by $\varepsilon = e - \frac{1}{2}\mathbf{v}^2$. The domain of definition or admissibility set of Euler equations reads

$$\mathcal{D} = \left\{ \mathbf{U} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ \rho e \end{pmatrix} \in \mathbb{R}^{\mathrm{d}+2}, \quad \rho > 0, \quad \text{and,} \quad e - \frac{1}{2}\mathbf{v}^2 > 0 \right\}.$$

The specific physical entropy is denoted η and $\tau = \frac{1}{\rho}$ the specific volume, we make the fundamental assumption that $(\tau, \eta) \mapsto \varepsilon(\tau, \eta)$ is strictly convex which is equivalent to assume that $(\tau, \varepsilon) \mapsto \eta(\tau, \varepsilon)$ is strictly concave, refer to [25]. We work with the physical entropy, entropy flux pair $(\rho\eta, \rho\mathbf{v}\eta)$ and thus the gas dynamics system of conservation laws is equipped with the entropy inequality

$$\frac{\partial \rho\eta}{\partial t} + \nabla \cdot (\rho\eta\mathbf{v}) \geq 0. \tag{2}$$

This inequality is the mathematical expression of the second law of thermodynamics. Namely, for smooth flows entropy is conserved, whereas it is increasing across discontinuities such as shock waves. In addition, entropy inequality (2) ensures the selection of the physical solution. The thermodynamic closure of this system of conservation laws is ensured by means of the complete equation of state (EOS) [45]

$$p(\tau, \eta) = -\frac{\partial \varepsilon}{\partial \tau}, \quad \theta(\tau, \eta) = \frac{\partial \varepsilon}{\partial \eta}. \tag{3}$$

We make the classical assumption that the absolute temperature is strictly positive: $\theta > 0$. The convexity of the specific internal energy with respect to the specific volume allows us to define the isentropic sound speed

$$\frac{a^2}{\tau^2} = -\frac{\partial p}{\partial \tau} = \frac{\partial^2 \varepsilon}{\partial \tau^2}. \tag{4}$$

Let $\mathbf{n}$ be a unit normal vector, then one can show that the Jacobian matrix of the normal flux, *i.e.*, $\frac{\partial \mathbb{F}(\mathbf{U})\mathbf{n}}{\partial \mathbf{U}}$ is diagonalizable and admits the real eigenvalues

$$\Lambda_- = (\mathbf{v} \cdot \mathbf{n}) - a, \Lambda_0 = (\mathbf{v} \cdot \mathbf{n}) \text{ multiplicity d, and } \Lambda_+ = (\mathbf{v} \cdot \mathbf{n}) + a, \text{ for all } \mathbf{U} \in \mathcal{D} \text{ and } \mathbf{n},$$

associated with a complete set of eigenvectors, which shows the hyperbolicity of Euler equations, refer for instance to [58].

Remark 2.1 (Gamma gas law). *For the numerical applications, we shall utilize the perfect gas EOS which expresses the pressure in terms of the specific volume and the specific internal energy as follows*

$$p = (\gamma - 1)\rho\varepsilon, \tag{5}$$

where γ is the polytropic index. The corresponding isentropic sound speed writes

$$a^2 = \frac{\gamma p}{\rho}. \tag{6}$$

2.2. Associated Riemann problem

The integral form of the Euler equations is obtained by integrating them over the fixed control volume $\omega \subset \mathbb{R}^{\mathrm{d}}$ and applying the divergence theorem

$$\frac{d}{dt}\int_{\omega} \mathbf{U}\,\mathrm{d}v + \int_{\partial\omega} \mathbb{F}(\mathbf{U})\mathbf{n}\,\mathrm{d}s = \mathbf{0}.$$

Here, $\mathbf{n}$ denotes the outward unit normal to the control volume boundary $\partial\omega$. We introduce the normal flux, $\mathbf{F}_{\mathbf{n}}$, which is the projection of the tensor flux onto the unit normal

$$\mathbf{F}_{\mathbf{n}}(\mathbf{U}) = \mathbb{F}(\mathbf{U})\mathbf{n}. \tag{7}$$

The construction of the FV approximation of Euler equations relies on the foregoing integral form and on a consistent numerical approximation of the normal flux at the cell interface. The construction of such an approximation stems from an approximate Riemann solver, which is nothing but the approximate solution to the Riemann problem associated to the system of conservation laws under consideration [25]. We write this associated Riemann problem in a local coordinate system attached to the interface characterized by its unit normal $\mathbf{n}$. In this local frame, the one-dimensional abscissa along the normal direction reads $x_{\mathbf{n}} = \mathbf{x}\cdot\mathbf{n}$ and the velocity vector is decomposed into the sum of its normal component $v_{\mathbf{n}} = \mathbf{v}\cdot\mathbf{n}$ and remaining tangential vector $\mathbf{v}_t = \mathbf{v} - v_{\mathbf{n}}\mathbf{n}$, as follows

$$\mathbf{v} = v_{\mathbf{n}}\mathbf{n} + \mathbf{v}_t.$$

Obviously, $\mathbf{v}^2 = v_{\mathbf{n}}^2 + \mathbf{v}_t^2$ since $\mathbf{v}_t\cdot\mathbf{n} = 0$. Finally, the Riemann problem in the direction $\mathbf{n}$ associated to the Euler equations reads

$$(\mathcal{RP}) \quad \begin{cases} \dfrac{\partial \mathbf{U}}{\partial t} + \dfrac{\partial \mathbf{F}_{\mathbf{n}}}{\partial x_{\mathbf{n}}} = \mathbf{0} \\ \mathbf{U}(0, x_{\mathbf{n}}) = \begin{cases} \mathbf{U}_l \text{ if } x_{\mathbf{n}} < 0, \\ \mathbf{U}_r \text{ if } x_{\mathbf{n}} \geq 0. \end{cases} \end{cases}$$

Here, the vector of conservative variables and the normal flux are defined in the local frame as

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_{\mathbf{n}} \\ \rho \mathbf{v}_t \\ \rho e \end{pmatrix} \text{ and } \mathbf{F}_{\mathbf{n}} = \begin{pmatrix} \rho v_{\mathbf{n}} \\ \rho v_{\mathbf{n}}^2 + p \\ \rho v_{\mathbf{n}} \mathbf{v}_t \\ \rho v_{\mathbf{n}} e + p v_{\mathbf{n}} \end{pmatrix}.$$

The vectors $\mathbf{U}_l$ and $\mathbf{U}_r$ denote the constant left and right states located on both sides of the interface knowing that the unit normal $\mathbf{n}$ is pointing in the right direction.

We shall construct an approximate solution to the Riemann problem $(\mathcal{RP})$ in Section 3.

Remark 2.2 (Abuse of notation). *For the sake of notational simplicity, we keep the same notation for the vector of conservative variables,* $\mathbf{U}$*, and the normal flux vector,* $\mathbf{F}_{\mathbf{n}}(\mathbf{U})$*, although their momentum components are written in the local frame related to the unit normal* $\mathbf{n}$ *and its tangential plane.*

Remark 2.3 (Flux decomposition). *It is worth noting that the normal flux might be decomposed into an advective part plus a pressure/Lagrangian one*

$$\mathbf{F}_{\mathbf{n}} = v_{\mathbf{n}}\mathbf{U} + \mathbf{L}(\mathbf{U}), \quad \textit{where} \quad \mathbf{L}(\mathbf{U}) = \begin{pmatrix} 0 \\ p \\ \mathbf{0} \\ p v_{\mathbf{n}} \end{pmatrix}. \tag{8}$$

This particular decomposition shall be useful in the subsequent developments for constructing the numerical flux of our Finite Volume approach. Although quite natural, this decomposition of the flux into a convective plus a pressure part has been almost never used except in the context of Flux Vector Splitting [60].

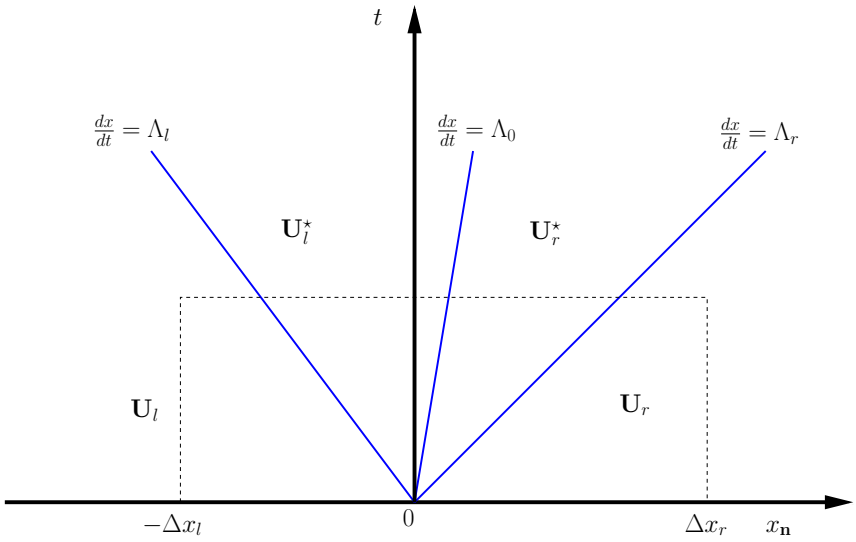


Figure 1: Representation of the simple approximate Riemann solver in the $x-t$ diagram.

3. Construction of the simple approximate Riemann solver

3.1. Notations

Here, we construct our approximate Riemann solver employing the notion of *simple Riemann solver* defined in the seminal work [21]. For the sake of clarity and ease of reading we follow the presentation described in [22]. The simple approximate Riemann solver $\mathbf{W}(\mathbf{U}_l, \mathbf{U}_r, \xi)$, where $\xi = \frac{x_\mathbf{n}}{t}$ consists of 4 constant states $\mathbf{U}_l$, $\mathbf{U}_l^\star$, $\mathbf{U}_r^\star$, $\mathbf{U}_r$ separated by 3 waves whose speeds are denoted Λ_l, Λ_0 and Λ_r, refer to Figure 1. More precisely, this approximate Riemann solver reads

$$\mathbf{W}(\mathbf{U}_l, \mathbf{U}_r, \xi) = \begin{cases} \mathbf{U}_l \text{ if } \xi < \Lambda_l, \\ \mathbf{U}_l^\star \text{ if } \Lambda_l \leq \xi < \Lambda_0, \\ \mathbf{U}_r^\star \text{ if } \Lambda_0 \leq \xi < \Lambda_r, \\ \mathbf{U}_r \text{ if } \Lambda_r \leq \xi. \end{cases}$$

To achieve the construction of this Riemann solver, we have to express the intermediate states $\mathbf{U}_l^\star$ and $\mathbf{U}_r^\star$ in terms of the wave speeds Λ_l, Λ_0, Λ_r and also in terms of the left $\mathbf{U}_l$ and right $\mathbf{U}_r$ states. For the moment, we consider the wave speeds as given parameters and we shall see further how to compute them in a systematic manner. First, we assume that the intermediate states write

$$\mathbf{U}_s^\star = \begin{pmatrix} \rho_s^\star \\ \rho_s^\star v_{\mathbf{n},s}^\star \\ \rho_s^\star \mathbf{v}_{t,s}^\star \\ \rho_s^\star e_s^\star \end{pmatrix}, \text{ for } s = l, r. \tag{9}$$

The intermediate specific total energy is defined by $e_s^\star = \varepsilon_s^\star + \frac{1}{2}((v_{\mathbf{n},s}^\star)^2 + (\mathbf{v}_{t,s}^\star)^2)$.

3.2. Characterization of the Riemann solver in terms of intermediate fluxes

To compute the intermediate states $\mathbf{U}_s^\star$ we introduce the intermediate fluxes $\mathbf{F}_{\mathbf{n},s}^\star$. Recalling that the left and right fluxes satisfy

$$\mathbf{F}_{\mathbf{n},s} = v_{\mathbf{n},s}\mathbf{U}_s + \begin{pmatrix} 0 \\ p_s \\ \mathbf{0} \\ p_s v_{\mathbf{n},s} \end{pmatrix}, \text{ for } s = l, r,$$

it is thus quite natural to assume that the intermediate fluxes mimick the decomposition of the physical normal flux (8) into a convective plus a pressure part

$$\mathbf{F}_{\mathbf{n},s}^\star = v_{\mathbf{n},s}^\star \mathbf{U}_s^\star + \begin{pmatrix} 0 \\ p_s^\star \\ \mathbf{0} \\ (pv_{\mathbf{n}})_s^\star \end{pmatrix}, \text{ for } s = l, r. \tag{10}$$

The intermediate states and fluxes represent a total of 2d+8 scalar unknowns. The supplementary equations needed to compute these unknowns are obtained writing the conservation relations across the Λ_l-wave and the Λ_r-wave as

$$-\Lambda_l(\mathbf{U}_l^\star - \mathbf{U}_l) + \mathbf{F}_{\mathbf{n},l}^\star - \mathbf{F}_{\mathbf{n},l} = \mathbf{0}, \tag{11a}$$
$$-\Lambda_r(\mathbf{U}_r - \mathbf{U}_r^\star) + \mathbf{F}_{\mathbf{n},r} - \mathbf{F}_{\mathbf{n},r}^\star = \mathbf{0}. \tag{11b}$$

These vectorial equations are Rankine-Hugoniot-like conditions written across the discontinuities of speeds Λ_l and Λ_r. They only represent 2d + 4 scalar equations. We close the problem making the following assumptions on the intermediate states and fluxes

- $v_{\mathbf{n},l}^\star = v_{\mathbf{n},r}^\star = v_{\mathbf{n}}^\star$,
- $(pv_{\mathbf{n}})_l^\star = p_l^\star v_{\mathbf{n}}^\star$ and $(pv_{\mathbf{n}})_r^\star = p_r^\star v_{\mathbf{n}}^\star$.

We note that the continuity of the normal velocity across the Λ_0-wave is a natural assumption consistent with the structure of the exact solution to the Riemann problem, refer to [58]. On the other hand, we decide to keep *a priori* $p_l^\star \neq p_r^\star$, which is rather unusual! Finally, we have reduced the number of scalar unknowns to 2d + 5, namely just one more than the number of equations given by (11a) and (11b). This will allow us to use one of the unknowns as a parameter that shall be expressed employing the conservation across the Λ_0-wave. We shall choose $v_{\mathbf{n}}^\star$ to be this parameter.

Substituting the decomposition (10) of the intermediate fluxes into the conservation relations (11a) and (11b) turns them into

$$(v_{\mathbf{n}}^\star - \Lambda_l)\mathbf{U}_l^\star - (v_{\mathbf{n},l} - \Lambda_l)\mathbf{U}_l + \mathbf{L}_l^\star - \mathbf{L}_l = \mathbf{0}, \tag{12a}$$
$$(v_{\mathbf{n},r} - \Lambda_r)\mathbf{U}_r - (v_{\mathbf{n}}^\star - \Lambda_r)\mathbf{U}_r^\star + \mathbf{L}_r - \mathbf{L}_r^\star = \mathbf{0}. \tag{12b}$$

We notice that the first components of the foregoing conservation equations collapse to

$$\rho_l^\star(v_{\mathbf{n}}^\star - \Lambda_l) - \rho_l(v_{\mathbf{n},l} - \Lambda_l) = 0, \quad \text{for the } \Lambda_l\text{-wave},$$
$$\rho_r(v_{\mathbf{n},r} - \Lambda_r) - \rho_r^\star(v_{\mathbf{n}}^\star - \Lambda_r) = 0, \quad \text{for the } \Lambda_r\text{-wave}.$$

It is thus quite natural to introduce the mass flux parameters λ_l and λ_r as follows

$$\lambda_l = \rho_l^\star(v_{\mathbf{n}}^\star - \Lambda_l) = \rho_l(v_{\mathbf{n},l} - \Lambda_l), \tag{13a}$$
$$\lambda_r = -\rho_r(v_{\mathbf{n},r} - \Lambda_r) = -\rho_r^\star(v_{\mathbf{n}}^\star - \Lambda_r). \tag{13b}$$

Then, (13a) and (13b) might be rewritten

$$\Lambda_l = v_{\mathbf{n},l} - \frac{\lambda_l}{\rho_l} = v_{\mathbf{n}}^\star - \frac{\lambda_l}{\rho_l^\star}, \tag{14a}$$

$$\Lambda_r = v_{\mathbf{n}}^\star + \frac{\lambda_r}{\rho_r^\star} = v_{\mathbf{n},r} + \frac{\lambda_r}{\rho_r}, \tag{14b}$$

which allows us to define the wave speeds of the simple approximate Riemann solver in terms of the mass flux paramaters. We note in passing that the definition of these wave speeds requires the fulfillment of the compatibility conditions

$$\lambda_l(\tau_l^\star - \tau_l) - (v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) = 0, \quad \text{and} \quad \lambda_r(\tau_r^\star - \tau_r) + v_{\mathbf{n}}^\star - v_{\mathbf{n},r} = 0, \tag{15}$$

where $\tau_s = \frac{1}{\rho_s}$ and $\tau_s^\star = \frac{1}{\rho_s^\star}$ for $s = l, r$.

Now, substituting the expressions (14a) and (14b) of the wave speeds into the conservation relations (12a) and (12b) transforms them into

$$\lambda_l\left(\frac{\mathbf{U}_l^\star}{\rho_l^\star} - \frac{\mathbf{U}_l}{\rho_l}\right) + \mathbf{L}_l^\star - \mathbf{L}_l = \mathbf{0},$$
$$\lambda_r\left(\frac{\mathbf{U}_r^\star}{\rho_r^\star} - \frac{\mathbf{U}_r}{\rho_r}\right) - \left(\mathbf{L}_r^\star - \mathbf{L}_r\right) = \mathbf{0}.$$

This system of vectorial equations completed by the compatibility conditions is written component-wise under the form of two systems of d + 2 equations. We have respectively for the leftmost wave

$$(\mathcal{S}_l)\begin{cases} \lambda_l(\tau_l^\star - \tau_l) - (v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) = 0, \\ \lambda_l(v_{\mathbf{n}}^\star - v_{\mathbf{n},l}) + p_l^\star - p_l = 0, \\ \lambda_l(\mathbf{v}_{t,l}^\star - \mathbf{v}_{t,l}) = \mathbf{0}, \\ \lambda_l(e_l^\star - e_l) + p_l^\star v_{\mathbf{n}}^\star - p_l v_{\mathbf{n},l} = 0, \end{cases}$$

and for the rightmost one

$$(\mathcal{S}_r)\begin{cases} \lambda_r(\tau_r^\star - \tau_r) + v_{\mathbf{n}}^\star - v_{\mathbf{n},r} = 0, \\ \lambda_r(v_{\mathbf{n}}^\star - v_{\mathbf{n},r}) - (p_r^\star - p_r) = 0, \\ \lambda_r(\mathbf{v}_{t,r}^\star - \mathbf{v}_{t,r}) = \mathbf{0}, \\ \lambda_r(e_r^\star - e_r) - (p_r^\star v_{\mathbf{n}}^\star - p_r v_{\mathbf{n},r}) = 0. \end{cases}$$

Let us recall that we have 2d + 4 equations for 2d + 5 unknowns. This means that 2d + 4 unknowns might be expressed in terms of one parameter that we choose to be the normal component of the intermediate velocity $v_{\mathbf{n}}^\star$. The constant mass flux parameters λ_l and λ_r shall be tuned to enforce the positivity of the intermediate specific volumes, *i.e.*, $\tau_s^\star > 0$ and the positivity of the intermediate specific internal energies, *i.e.*, $\varepsilon_s^\star > 0$ for $s = l, r$. Adapting the methodology introduced in [16] to the present Riemann solver leads to the positivity conditions

$$\lambda_l \geq \max\left(\frac{a_l}{\tau_l}, -\frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},l}}{\tau_l}\right), \tag{16a}$$

$$\lambda_r \geq \max\left(\frac{a_r}{\tau_r}, \frac{v_{\mathbf{n}}^\star - v_{\mathbf{n},r}}{\tau_r}\right). \tag{16b}$$

The study of the entropy production term [16] related to systems $(\mathcal{S}_l)$ and $(\mathcal{S}_r)$ shows that specific entropy increases across the discontinuity, *i.e.*, $\eta_s^\star - \eta_s \geq 0$ provided that the wave speed satisfies the condition

$$\lambda_s^2 \geq \frac{a^2(\overline{\tau}_s, \eta_s)}{\overline{\tau}_s^2} \quad \text{for all} \quad \overline{\tau}_s \in (\tau_s, \tau_s^\star), \quad \text{for } s = l, r. \tag{17}$$

3.3. Interface fluxes definition

It remains to deal with the central Λ_0-wave to give the expression of Λ_0 and also to assess the consistency property of the approximate Riemann solver with the integral form of the conservation law of $(\mathcal{RP})$. This amounts to check whether the famous HLL condition is satisfied, refer [27, 28]. We recall that this consistency condition plays a crucial role in the construction of HLL-like approximate Riemann solver since it expresses the conservation property of the Finite Volume discretization build upon the approximate Riemann solver.

To evaluate the balance across the Λ_0-wave, for $t > 0$, we integrate the conservation law of $(\mathcal{RP})$ over $[\Lambda_0 t - \delta, \Lambda_0 t + \delta]$, where $\delta > 0$, to get

$$\int_{\Lambda_0 t-\delta}^{\Lambda_0 t+\delta} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_{\mathbf{n}}(\mathbf{U})}{\partial x_{\mathbf{n}}} \right) \mathrm{d}x_{\mathbf{n}} =$$
$$\frac{d}{dt} \int_{\Lambda_0 t-\delta}^{\Lambda_0 t+\delta} \mathbf{U}\, \mathrm{d}x - \Lambda_0 \left[\mathbf{U}(\Lambda_0 t + \delta, t) - \mathbf{U}(\Lambda_0 t - \delta, t) \right] + \mathbf{F}_{\mathbf{n}}[\mathbf{U}(\Lambda_0 t + \delta, t)] - \mathbf{F}_{\mathbf{n}}[\mathbf{U}(\Lambda_0 t - \delta, t)],$$

letting δ tend to zero we arrive at

$$\lim_{\delta \to 0} \int_{\Lambda_0 t-\delta}^{\Lambda_0 t+\delta} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_{\mathbf{n}}(\mathbf{U})}{\partial x_{\mathbf{n}}} \right) \mathrm{d}x_{\mathbf{n}} = -\Lambda_0 (\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \mathbf{F}_{\mathbf{n},r}^\star - \mathbf{F}_{\mathbf{n},l}^\star.$$

Developing the right hand-side of the foregoing equation by substituting the decomposition of the intermediate fluxes (10) leads to

$$-\Lambda_0 (\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \mathbf{F}_{\mathbf{n},r}^\star - \mathbf{F}_{\mathbf{n},l}^\star = (v_{\mathbf{n}}^\star - \Lambda_0)(\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \mathbf{L}_r^\star - \mathbf{L}_l^\star, \tag{18}$$

where the difference between the pressure/Lagrangian fluxes writes

$$\mathbf{L}_r^\star - \mathbf{L}_l^\star = (p_r^\star - p_l^\star) \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^\star \end{pmatrix}.$$

We point out that contrarily to the classical framework, (18) is not a conservation relation across the Λ_0-wave due to the fact that the right hand side is not equal to zero since *a priori* $p_l^\star \neq p_r^\star$. Nonetheless, this does not prevent us from defining the wave speed by setting

$$\Lambda_0 = v_{\mathbf{n}}^\star. \tag{19}$$

With this definition, the balance relation across Λ_0-wave boils down to

$$-\Lambda_0 (\mathbf{U}_r^\star - \mathbf{U}_l^\star) + \mathbf{F}_{\mathbf{n},r}^\star - \mathbf{F}_{\mathbf{n},l}^\star = \mathbf{L}_r^\star - \mathbf{L}_l^\star = (p_r^\star - p_l^\star) \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^\star \end{pmatrix}. \tag{20}$$

Now, it remains to find how to compute the parameter $v_{\mathbf{n}}^\star$. To this end, we shall assess the consistency of our approximate Riemann solver with the integral form of the conservation of $(\mathcal{RP})$.

Following [28], let us recall the definition of the left and right-sided interface fluxes written in terms of the approximate Riemann solver $\mathbf{W}$ as follow

$$\mathbf{F}_{\mathbf{n}}^- = \mathbf{F}_{\mathbf{n},l} - \int_{-\infty}^{0} [\mathbf{W}(\mathbf{U}_l, \mathbf{U}_r, \xi) - \mathbf{U}_l]\, \mathrm{d}\xi,$$
$$\mathbf{F}_{\mathbf{n}}^+ = \mathbf{F}_{\mathbf{n},r} + \int_{0}^{+\infty} [\mathbf{W}(\mathbf{U}_l, \mathbf{U}_r, \xi) - \mathbf{U}_r]\, \mathrm{d}\xi.$$

These expressions stem from integrating $(\mathcal{RP})$ respectively over $[-\frac{\Delta x_l}{2}, 0] \times [0, \Delta t]$ and $[0, \frac{\Delta x_r}{2} \times [0, \Delta t]$ for a sufficientlty small Δt, refer to Figure 1. Substituting the expression of the simple approximate Riemann solver into these left and right-sided fluxes yields

$$\mathbf{F}_{\mathbf{n}}^{-} = \mathbf{F}_{\mathbf{n},l} - \Lambda_l^{-}(\mathbf{U}_l^{\star} - \mathbf{U}_l) - \Lambda_0^{-}(\mathbf{U}_r^{\star} - \mathbf{U}_l^{\star}) - \Lambda_r^{-}(\mathbf{U}_r - \mathbf{U}_r^{\star}), \tag{21a}$$

$$\mathbf{F}_{\mathbf{n}}^{+} = \mathbf{F}_{\mathbf{n},r} - \Lambda_l^{+}(\mathbf{U}_l^{\star} - \mathbf{U}_l) - \Lambda_0^{+}(\mathbf{U}_r^{\star} - \mathbf{U}_l^{\star}) - \Lambda_r^{+}(\mathbf{U}_r - \mathbf{U}_r^{\star}). \tag{21b}$$

Here, for any real value x we introduce its negative and positive parts as follows $x^{-} = \frac{1}{2}(|x| - x)$ and $x^{+} = \frac{1}{2}(|x| + x)$. We compute the difference between the right and the left-sided fluxes, noticing that $x^{+} - x^{-} = x$, and we obtain

$$\mathbf{F}_{\mathbf{n}}^{+} - \mathbf{F}_{\mathbf{n}}^{-} = -\Lambda_l(\mathbf{U}_l^{\star} - \mathbf{U}_l) - \Lambda_0(\mathbf{U}_r^{\star} - \mathbf{U}_l^{\star}) - \Lambda_r(\mathbf{U}_r - \mathbf{U}_r^{\star}) + \mathbf{F}_{\mathbf{n},r} - \mathbf{F}_{\mathbf{n},l}.$$

On the other hand, summing the conservation relations (11a) and (11b) across the Λ_l-wave and the Λ_r-wave plus the relation (20) across the Λ_0-wave we arrive at the fundamental result

$$\boxed{\mathbf{F}_{\mathbf{n}}^{+} - \mathbf{F}_{\mathbf{n}}^{-} = \mathbf{L}_r^{\star} - \mathbf{L}_l^{\star} = (p_r^{\star} - p_l^{\star}) \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^{\star} \end{pmatrix}}. \tag{22}$$

This result puts in evidence that the conservation property is completely linked to the value of the intermediate pressures difference $p_r^{\star} - p_l^{\star}$. This difference might be evaluated in terms of the normal velocity $v_{\mathbf{n}}^{\star}$ by combining the left and the right-sided momentum normal components of the conservation relations $(\mathcal{S}_l)$ and $(\mathcal{S}_r)$ that we recall hereafter

$$\lambda_l(v_{\mathbf{n}}^{\star} - v_{\mathbf{n},l}) + p_l^{\star} - p_l = 0,$$
$$\lambda_r(v_{\mathbf{n}}^{\star} - v_{\mathbf{n},r}) - (p_r^{\star} - p_r) = 0.$$

Summing the foregoing equations leads to

$$p_r^{\star} - p_l^{\star} = (\lambda_l + \lambda_r) \left\{ v_{\mathbf{n}}^{\star} - \left[\frac{\lambda_l v_{\mathbf{n},l} + \lambda_r v_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(p_r - p_l)}{\lambda_r + \lambda_l} \right] \right\}. \tag{23}$$

This incites us to introduce the normal velocity of the interface

$$\overline{v}_{\mathbf{n},lr} = \frac{\lambda_l v_{\mathbf{n},l} + \lambda_r v_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(p_r - p_l)}{\lambda_r + \lambda_l}, \tag{24}$$

which is formally similar to the velocity resulting from the one-dimensional approximate Riemann solver derived in [16]. Finally, the difference between the right and left-sided interface fluxes reads

$$\mathbf{F}_{\mathbf{n}}^{+} - \mathbf{F}_{\mathbf{n}}^{-} = (\lambda_l + \lambda_r) \left(v_{\mathbf{n}}^{\star} - \overline{v}_{\mathbf{n},lr} \right) \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^{\star} \end{pmatrix}. \tag{25}$$

Bearing this in mind and recalling that $v_{\mathbf{n}}^{\star}$ is still a free parameter, we arrive at the following alternative

- Either $v_{\mathbf{n}}^{\star} = \overline{v}_{\mathbf{n}}$, then the simple Riemann solver is consistent with its underlying conservation law. Thus, the simple Riemann solver induces a classical conservative Godunov-type Finite Volume scheme, which is characterized by a unique interface flux $\mathbf{F}_{\mathbf{n}}^{+} = \mathbf{F}_{\mathbf{n}}^{-}$;

- Or $v_{\mathbf{n}}^{\star} \neq \overline{v}_{\mathbf{n}}$, then the simple Riemann solver is not consistent with its underlying conservation law and thus not does not induce a conservative Godunov-type Finite Volume scheme in the classical sense since we cannot define a unique interface flux knowing that $\mathbf{F}_{\mathbf{n}}^{+} \neq \mathbf{F}_{\mathbf{n}}^{-}$

In what follows, we investigate further the second case for which in general $v_{\mathbf{n}}^{\star} \neq \overline{v}_{\mathbf{n}}$ and we shall demonstrate how to restore the conservation property for the Finite Volume scheme induced by such an approximate Riemann solver employing a *node-based conservation condition*.

We conclude this section by expressing more explicitly the left and right-sided interface fluxes. First, let us note that we have determined the expression of the difference between the right and the left-sided fluxes (25). Second, we can introduce the arithmetic average of the left and right-sided fluxes

$$\mathbf{F}_{\mathbf{n}}^{\star} = \frac{1}{2}\left(\mathbf{F}_{\mathbf{n},r}^{+} + \mathbf{F}_{\mathbf{n},l}^{-}\right) = \frac{1}{2}\left(\mathbf{F}_{\mathbf{n},r} + \mathbf{F}_{\mathbf{n},l}\right) - \frac{1}{2}\left[|\Lambda_l|(\mathbf{U}_l^{\star} - \mathbf{U}_l) + |\Lambda_0|(\mathbf{U}_r^{\star} - \mathbf{U}_l^{\star}) + |\Lambda_r|(\mathbf{U}_r - \mathbf{U}_r^{\star})\right]. \tag{26}$$

This is formally the same formula for the interface flux than the one we have obtained in the one-dimensional case. However, in general this does not induce a conservative Finite Volume method since the left and right-sided fluxes are *a priori* distinct! Combining the arithmetic average (26) of the fluxes with their difference (25) allows us to express them as follows

$$\mathbf{F}_{\mathbf{n}}^{-} = \mathbf{F}_{\mathbf{n}}^{\star} - \frac{1}{2}(\lambda_l + \lambda_r)\left(v_{\mathbf{n}}^{\star} - \overline{v}_{\mathbf{n},lr}\right)\begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^{\star} \end{pmatrix}, \tag{27a}$$

$$\mathbf{F}_{\mathbf{n}}^{+} = \mathbf{F}_{\mathbf{n}}^{\star} + \frac{1}{2}(\lambda_l + \lambda_r)\left(v_{\mathbf{n}}^{\star} - \overline{v}_{\mathbf{n},lr}\right)\begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v_{\mathbf{n}}^{\star} \end{pmatrix}. \tag{27b}$$

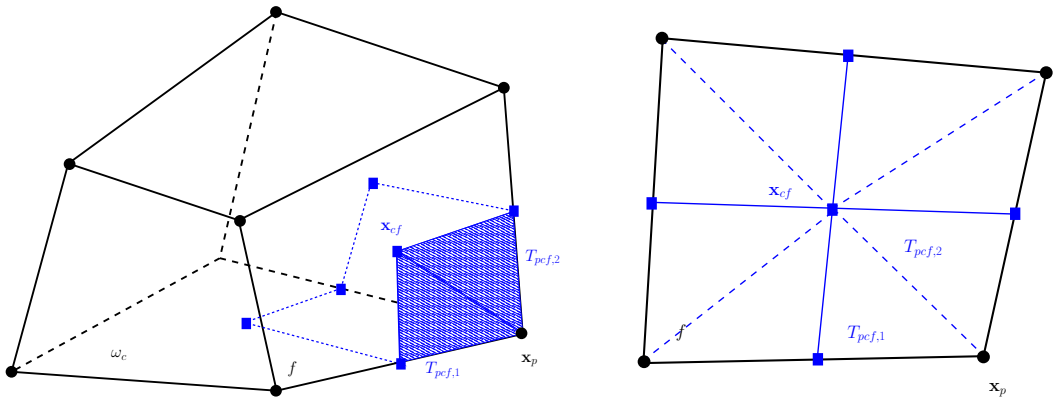
Remark 3.1. *We note that the positivity preserving conditions* (16a) *and* (16b) *not only determine* λ_l *and* λ_r *but also imply that* $\lambda_l > 0$ *and* $\lambda_r > 0$*. This in turn, by virtue of the waves speeds expressions* (14a), (19) *and* (14b)*, involves the waves speeds ordering* $\Lambda_l \leq \Lambda_0 \leq \Lambda_r$*. Let us emphasize that the waves speeds have been determined straightforwardly contrary to what is done for HLLC approximate Riemann solver for which an ad hoc procedure has to be settled to define them, refer for instance to [6, 57]. In addition, it is worth noting that all the foregoing developments have been undertaken with few assumptions on the equation of state.*

4. Unconventional Finite Volume discretization

In this section we derive an unconventional Finite Volume discretization for the multidimensional Euler equations. We call it unconventional in the sense that it relies on an interface flux which is not uniquely defined and it thus requires a novel way of taking into account conservation. This will be achieved through the introduction of a node-based conservation condition.

4.1. Notations

We shall construct the Finite Volume discretization of the Euler equations within the three-dimensional space $\mathbb{R}^3$, which is equipped with the orthonormal direct basis $(O, \mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z)$. The computational domain $\omega \subset \mathbb{R}^3$ is paved with N_c non-overlapping polyhedral cells denoted ω_c. In what follows, the label c always refers to the generic cell ω_c for $c = 1 \dots N_c$. The cell vertices (points) are characterized by the label p and their vector position $\mathbf{x}_p = (x_p, y_p, z_p)^t$ for $p = 1 \dots N_p$ where N_p is the total number of vertices. The construction of the spatial discretization shall rely on specific partitions of the computational domain which requires to introduce the following global geometrical sets



(a) Subface of face f attached to point p and cell c.

(b) Face partition into subfaces.

Figure 2: Geometrical entities attached to the hexaedral cell ω_c.

- $\mathcal{C}$: The set of cells of the mesh;
- $\mathcal{P}$: The set of vertices (points) of the mesh;
- $\mathcal{F}$: The set of faces of the mesh;
- $\mathcal{E}$: The set of edges of the mesh.

We also need the following local geometrical sets

- $\mathcal{P}(c)$: The set of vertices of cell $c \in \mathcal{C}$;
- $\mathcal{F}(c)$: The set of faces of cell $c \in \mathcal{C}$;
- $\mathcal{E}(f)$: The set of edges of face $f \in \mathcal{F}$;
- $\mathcal{C}(p)$: The set of cells around node $p \in \mathcal{P}$;
- $\mathcal{F}(p)$: The set of faces impinging at node $p \in \mathcal{P}$;
- $\mathcal{E}(p)$: The set of edges impinging at node $p \in \mathcal{P}$;

Finally, supplementary subsets might be created simply by intersecting existing ones

- $\mathcal{E}(fp) = \mathcal{E}(f) \cap \mathcal{E}(p)$: The set of edges belonging to face $f \in \mathcal{F}$ and attached to node $p \in \mathcal{P}$.

Each polygonal face of the polyhedral cells that pave the computational domain is split into subfaces in the following manner: for $f \in \mathcal{F}(c)$ we define its centroid $\mathbf{x}_{cf}$, refer to Figure 2. Then, the subface related to face f, attached to cell c and point p is defined as being the quadrangle obtained by joigning the point p, the midpoints of the two edges $e \in \mathcal{E}(fp)$ and the centroid $\mathbf{x}_{cf}$, refer to Figure 2a. As a consequence, we introduce the sets related to the subfaces:

- $\mathcal{SF}$: The set of subfaces of the mesh;
- $\mathcal{SF}(c)$: The set of subfaces of cell $c \in \mathcal{C}$, which is a partition of $\partial\omega_c$;
- $\mathcal{SF}(p)$: The set of subfaces impinging at node $p \in \mathcal{P}$;

- $\mathcal{SF}(pc) = \mathcal{SF}(c) \cap \mathcal{SF}(p)$: The set of subfaces of cell $c \in \mathcal{C}$ impinging at node $p \in \mathcal{P}$.

Regardless the type of the polyhedral cell, the subface is always a quadrangle, which in general might not be a planar surface. That is why, the quadrangular subface is decomposed into two triangles, $T_{pcf,1}$ and $T_{pcf,2}$, refer to Figure 2a and Figure 2b, which are planar and characterized by their unit outward normals $\mathbf{n}_{pcf,1}$ and $\mathbf{n}_{pcf,2}$. Then, the area, A_{pcf}, and the unit normal to the subface pcf are computed as follows

$$A_{pcf}\mathbf{n}_{pcf} = |T_{pcf,1}|\mathbf{n}_{pcf,1} + |T_{pcf,2}|\mathbf{n}_{pcf,2},$$

where $\mathbf{n}_{pcf}^2 = 1$. At last, we introduce the subcell, ω_{pc}, related to cell c and attached to p by joining the subfaces in $\mathcal{SF}(pc)$ to the centroid of cell $c \in \mathcal{C}$. The following sets are added:

- $\mathcal{SC}$: The set of subcells of the mesh;
- $\mathcal{SC}(c)$: The set of subcells of cell $c \in \mathcal{C}$, which is a partition of ω_c;
- $\mathcal{SC}(p)$: The set of subcells impinging at node $p \in \mathcal{P}$.

Now, gathering the subcells sharing point p allows us to introduce the dual cell attached to this point

$$\omega_p = \bigcup_{c \in \mathcal{C}(p)} \omega_{pc}. \tag{28}$$

Integrating the system of conservation laws (1) over ω_c and applying divergence theorem leads to

$$|\omega_c|\frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \int_{\partial\omega_c} \mathbb{F}(\mathbf{U})\mathbf{n}\,\mathrm{d}s = \mathbf{0}, \tag{29}$$

where $\mathbf{U}_c(t) = \frac{1}{|\omega_c|}\int_{\omega_c} \mathbf{U}(\mathbf{x}, t)\,\mathrm{d}v$ is the cell-averaged value of $\mathbf{U}$ over ω_c. Employing a classical first-order explicit time integration turns (29) into

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \mathbf{0}. \tag{30}$$

Here, $\mathbf{U}_c^n$ denotes the approximation of $\mathbf{U}_c(t)$ at time t^n, and, $t^{n+1} = t^n + \Delta t$ where $\Delta t > 0$ is the time step.

The design of the Finite Volume scheme (30) requires to construct an approximation of the normal flux integral. In what follows, we are going to define a node-based approximation of this integral term relying on the partition of $\partial\omega_c$ into subfaces, that is,

$$\int_{\partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{p \in \mathcal{P}(c)} \int_{\partial\omega_{pc} \cap \partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s. \tag{31}$$

The surface integral term at the right-hand side of (31) is approximated along the subfaces as follows

$$\int_{\partial\omega_{pc} \cap \partial\omega_c} \mathbb{F}(\mathbf{U}^n)\mathbf{n}\,\mathrm{d}s = \sum_{f \in \mathcal{SF}(pc)} A_{pcf}\overline{\mathbf{F}}_{pcf},$$

where $\overline{\mathbf{F}}_{pcf}$ is the subface flux related to the subface f attached to the corner pc. Substituting $\overline{\mathbf{F}}_{pcf}$ into (30) yields the subface-based generic Finite Volume scheme

$$\mathbf{U}_c^{n+1} - \mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\sum_{p \in \mathcal{P}(c)}\sum_{f \in \mathcal{SF}(pc)} A_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}, \tag{32}$$

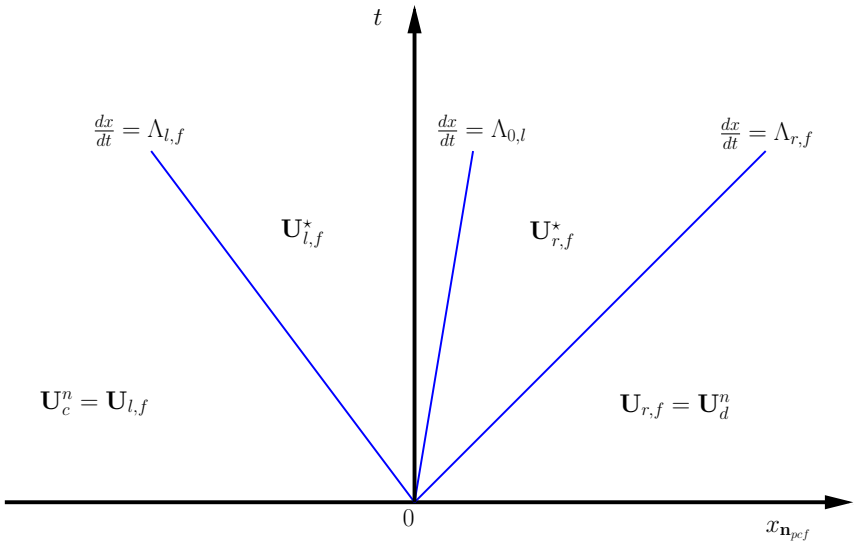


Figure 3: Representation of the approximate Riemann solver $\mathbf{W}(\mathbf{U}_c, \mathbf{U}_d, \xi_{pcf})$ in the $x_{\mathbf{n}_{pcf}} - t$ plane.

which is entirely characterized by the subface flux $\overline{\mathbf{F}}_{pcf}$. Viewed from cell ω_c, this peculiar Finite Volume discretization introduces as many subface fluxes than subface per cell face. In the three-dimensional case, the number of subface fluxes per face cell is at least equal to 3. In addition, if one consider the cell ω_d sharing the face f with the cell ω_c, we note that *a priori*, the subface fluxes $\overline{\mathbf{F}}_{pcf}$ and $\overline{\mathbf{F}}_{pdf}$ are distinct! We observe that this type of Finite Volume discretization, which consists in splitting the faces into subfaces, has been already utilized not only in the framework of cell-centered Lagrangian hydrodynamics [42, 41] but also for developing cell-centered diffusion schemes [44, 32] on unstructured grids. We also point out that this formalism encompasses the classical face-based Finite Volume discretization [25]. Now, it remains to provide a consistent numerical approximation of the subface flux.

4.2. Numerical approximation of the subface flux and convex combination property

We define the subface flux related to the subface f attached to cell ω_c, impinging at point p and characterized by its unit outward normal $\mathbf{n}_{pcf}$ as follows

$$\overline{\mathbf{F}}_{pcf} = \mathbf{F}^{-}_{\mathbf{n}_{pcf}}, \tag{33}$$

where $\mathbf{F}^{-}_{\mathbf{n}_{pcf}}$ is the left-sided interface flux, refer to section 3.3, computed from the approximate Riemann solver $\mathbf{W}(\mathbf{U}_c, \mathbf{U}_d, \xi_{pcf})$, refer to Figure 3, where d is the neighbor of cell c such that $f \subset (\omega_c \cap \omega_d)$ and $\xi_{pcf} = \frac{\mathbf{x}\cdot\mathbf{n}_{pcf}}{t}$. For the sake of completeness and with obvious notation adaptation, let us recall the expression of the left-sided interface flux, refer to (21a), in terms of the intermediate states and the wave speeds of our simple approximate Riemann solver

$$\mathbf{F}^{-}_{pcf} = \mathbb{F}(\mathbf{U}^{n}_{c})\mathbf{n}_{pcf} - \Lambda^{-}_{l,f}(\mathbf{U}^{\star}_{l,f} - \mathbf{U}^{n}_{c}) - \Lambda^{-}_{0,f}(\mathbf{U}^{\star}_{r,f} - \mathbf{U}^{\star}_{l,f}) - \Lambda^{-}_{r,f}(\mathbf{U}^{n}_{d} - \mathbf{U}^{\star}_{r,f}). \tag{34}$$

Substituting the expression of the subface flux into the discrete Finite Volume scheme (32) yields

$$\mathbf{U}^{n+1}_{c} - \mathbf{U}^{n}_{c} - \frac{\Delta t}{|\omega_c|} \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} A_{pcf} \left[\Lambda^{-}_{l,f}(\mathbf{U}^{\star}_{l,f} - \mathbf{U}^{n}_{c}) - \Lambda^{-}_{0,f}(\mathbf{U}^{\star}_{r,f} - \mathbf{U}^{\star}_{l,f}) - \Lambda^{-}_{r,f}(\mathbf{U}^{n}_{d} - \mathbf{U}^{\star}_{r,f})\right] = \mathbf{0}.$$

Here, we have used the geometrical identity $\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\mathbf{n}_{pcf} = \mathbf{0}$. Now, collecting the terms in factor of the intermediate states allows to express $\mathbf{U}_c^{n+1}$ as a linear combination of the intermediate states

$$\begin{aligned}\mathbf{U}_c^{n+1} = &\left[1 - \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\Lambda_{l,f}^-\right]\mathbf{U}_c^n + \frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\left(\Lambda_{l,f}^- - \Lambda_{0,f}^-\right)\mathbf{U}_{l,f}^\star \\ &+\frac{\Delta t}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\left[\left(\Lambda_{0,f}^- - \Lambda_{r,f}^-\right)\mathbf{U}_{r,f}^\star + \Lambda_{r,f}^-\mathbf{U}_d^n\right].\end{aligned} \tag{35}$$

Since $x \mapsto x^-$ is a decreasing function with respect to x and since the wave speeds are ordered, we get $\Lambda_{l,f}^- - \Lambda_{0,f}^- \geq 0$ and $\Lambda_{0,f}^- - \Lambda_{r,f}^- \geq 0$. This implies that $\mathbf{U}_c^{n+1}$ is a convex combination of $\mathbf{U}_c^n$, $\mathbf{U}_{l,f}^\star$, $\mathbf{U}_{r,f}^\star$ and $\mathbf{U}_d^n$ provided that the time step satisfies the condition

$$\Delta t \leq \frac{|\omega_c|}{\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\Lambda_{l,f}^-}. \tag{36}$$

This convex combination property under the time step condition (36) is very important since it allows to ensure that the Finite Volume scheme is domain preserving. This implies the preservation of positivity for the mass density and the specific internal energy provided that the approximate Riemann solver is positivity preserving. The notion of invariant domain is classical in the context of hyperbolic systems of conservation laws, refer for instance to [9]. A domain is invariant if for any initial condition $\mathbf{U}^0$ belonging to the domain, the solution of the hyperbolic system under consideration remains in the domain for all time $t > 0$. In the present case, introducing

$$\Delta t_c = \frac{|\omega_c|}{\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\Lambda_{l,f}^-}, \tag{37}$$

we claim that our Finite Volume scheme preserves the domain $\mathcal{D}$ under the global time-step condition

$$\Delta t \leq \min_c \Delta t_c. \tag{38}$$

This time step condition shall ensure not only the positivity preserving property of the numerical method but also its entropy stability, the interested reader might refer to [22] for more details about this topic.

Remark 4.1 (Pratical time step condition). *We give a practical formula for rewriting the time step condition* (36) *in a more explicit manner. By definition and with obvious notation adaptation*

$$\Lambda_{l,f}^- = \frac{1}{2}\left(\left|\mathbf{v}_c^n\cdot\mathbf{n}_{pcf} - \frac{\lambda_{pcf,l}}{\rho_c^n}\right| - \mathbf{v}_c^n\cdot\mathbf{n}_{pcf} + \frac{\lambda_{pcf,l}}{\rho_c^n}\right).$$

Since $\Lambda_{l,f}^- \leq |\mathbf{v}_c^n\cdot\mathbf{n}_{pcf}| + \frac{\lambda_{pcf,l}}{\rho_c^n}$ *we arrive at the practical time step condition*

$$\Delta t \leq \min_c\left\{\frac{|\omega_c|}{\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\left(|\mathbf{v}_c^n\cdot\mathbf{n}_{pcf}| + \frac{\lambda_{pcf,l}}{\rho_c^n}\right)}\right\}. \tag{39}$$

Now, since the left-sided and the right-sided interface fluxes are in general distinct, refer to section 3.3, the foregoing numerical flux approximation does not induce a conservative Finite Volume scheme in the classical sense. **Therefore, we need to provide a specific framework to enforce a conservation property for the present Finite Volume method.**

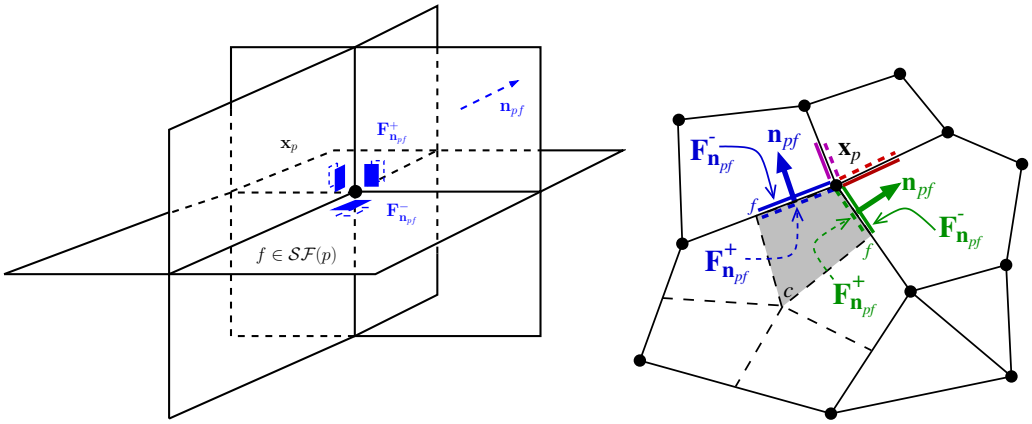


Figure 4: Fragment of the grid in the vicinity of point p. Left: Representation of the subface fluxes related to the front side octant. Right: 2D representation of the subface fluxes around a node p. The subcell is gray-shaded and the left/right-sided interface fluxes are represented by colored (dashed) lines. $\mathcal{E}(p)$ is constituted of the colored faces impinging at p.

4.3. Node-based conservation condition

This section aims at determining under which conditions the studied subface-based Finite Volume scheme is conservative. Assuming that the computational domain is the whole space $\mathbb{R}^2$, the subface-based Finite Volume scheme (32) is conservative if and only if $\sum_c |\omega_c|\mathbf{U}_c^{n+1} = \sum_c |\omega_c|\mathbf{U}_c^n$ for all $\Delta t > 0$. This amounts to write the global conservation as

$$\sum_c \sum_{p\in\mathcal{P}(c)} \sum_{f\in\mathcal{SF}(pc)} A_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}.$$

Now, exchanging the summation over the cells with the summation over the nodes, we get

$$\sum_p \sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(pc)} A_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0},$$

where $\mathcal{C}(p)$ is the set of cells sharing the point p. Finally, we claim that a sufficient condition to ensure conservation for the subface-based Finite Volume scheme writes

$$\sum_{c\in\mathcal{C}(p)} \sum_{f\in\mathcal{SF}(pc)} A_{pcf}\overline{\mathbf{F}}_{pcf} = \mathbf{0}. \tag{40}$$

This means that the summation over the cells c sharing point p of the fluxes attached to the subfaces impinging at p is equal to zero, where the subface fluxes have been displayed by means of blue patches on both sides of each subface emanating from point p, refer to Figure 4-left and Figure 4-right for a simpler 2D representation for a polygonal cell.. Notice that the sum over the cells c sharing p of the fluxes attached to the subfaces impinging at p is rigorously equal to the sum over the left-sided and the right-sided fluxes attached to the subfaces impinging at p. This amounts to rewrite condition (40) into

$$\sum_{f\in\mathcal{SF}(p)} A_{pf}(\mathbf{F}^+_{\mathbf{n}_{pf}} - \mathbf{F}^-_{\mathbf{n}_{pf}}) = \mathbf{0}. \tag{41}$$

Here, $\mathcal{SF}(p)$ denotes the set of subfaces impinging at point p. For $f \in \mathcal{SF}(p)$, A_{pf} is the length of subface f and $\mathbf{n}_{pf}$ is its unit normal. In the foregoing equation, $\mathbf{F}^-_{\mathbf{n}_{pf}}$ (resp. $\mathbf{F}^+_{\mathbf{n}_{pf}}$) denotes the left-sided (resp. right-sided) flux attached to the subface f, refer to Figure 4. Thanks to (25) the difference between the left and

the right-sided fluxes is expressed in terms of the approximate Riemann solver as

$$\mathbf{F}^{+}_{\mathbf{n}_{pf}} - \mathbf{F}^{-}_{\mathbf{n}_{pf}} = (\lambda_{l,pf} + \lambda_{r,pf}) \left(v^{\star}_{\mathbf{n}_{pf}} - \overline{v}_{\mathbf{n}_{pf},lr} \right) \begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v^{\star}_{\mathbf{n}_{pf}} \end{pmatrix},$$

where $\overline{v}_{\mathbf{n}_{pf},lr}$ is normal velocity of face f determined by (24) and $v^{\star}_{\mathbf{n}_{pf}}$ is the unknown normal velocity parameter attached to our approximate Riemann solver. This allows to write the node-based conservation condition (40) component-wise as follows

$$\sum_{f \in \mathcal{SF}(p)} A_{pf} (\lambda_{l,pf} + \lambda_{r,pf}) \left(v^{\star}_{\mathbf{n}_{pf}} - \overline{v}_{\mathbf{n}_{pf},lr} \right) \mathbf{n}_{pf} = \mathbf{0}, \tag{42a}$$

$$\sum_{f \in \mathcal{SF}(p)} A_{pf} (\lambda_{l,pf} + \lambda_{r,pf}) \left(v^{\star}_{\mathbf{n}_{pf}} - \overline{v}_{\mathbf{n}_{pf},lr} \right) v^{\star}_{\mathbf{n}_{pf}} = 0. \tag{42b}$$

We have $|\mathcal{SF}(p)|$ scalar unknowns, the normal velocity $\overline{v}_{\mathbf{n}_{pf},lr}$ of each subface impinging at p, for only d+1 scalar equations. For instance, in the case of a Cartesian grid, we have $|\mathcal{SF}(p)| = 12$ and $d + 1 = 4$. Therefore, to close this system of equations, we make the assumption that the parameter $v^{\star}_{\mathbf{n}_{pf}}$ is the projection of the unknown nodal vector $\mathbf{v}_p$ onto the unit normal $\mathbf{n}_{pf}$, that is

$$v^{\star}_{\mathbf{n}_{pf}} = \mathbf{v}_p \cdot \mathbf{n}_{pf}, \ \forall\, f \in \mathcal{SF}(p). \tag{43}$$

This fundamental assumption drastically reduces the number of unknowns to the vectorial unknown $\mathbf{v}_p$, which can be interpreted as an approximation of the nodal velocity. With this assumption the conservation condition (42b) is equivalent to the conservation condition (42a). We arrive at the conclusion that the node-based conservation condition (42a) boils down to the system

$$\sum_{f \in \mathcal{SF}(p)} A_{pf} (\lambda_{l,pf} + \lambda_{r,pf}) (\mathbf{n}_{pf} \otimes \mathbf{n}_{pf}) \mathbf{v}_p = \sum_{f \in \mathcal{SF}(p)} A_{pf} (\lambda_{l,pf} + \lambda_{l,pf}) \overline{v}_{\mathbf{n}_{pf}} \mathbf{n}_{pf}. \tag{44}$$

This system always admits a unique solution which provides an approximation of the nodal velocity $\mathbf{v}_p$. It is thus called a nodal solver. We point out that the foregoing system has been already obtained when constructing a cell-centered Finite Volume discretization of multidimensional Lagrangian hydrodynamics [42]. It has been also retrieved in [51] for designing a Finite Volume scheme for Eulerian gas dynamics.

Remark 4.2 (Entropy inequality). *We are also able to derive an entropy inequality for our unconventional Finite Volume method. This might be done employing the convex combination property of the Finite Volume scheme and applying the Jensen's inequlity. The interested reader might refer to [22] for a detailed derivation of this entropic inequality.*

4.4. Summary

We have derived a peculiar Finite Volume discretization of the Euler equations

$$\mathbf{U}^{n+1}_c - \mathbf{U}^{n}_c + \frac{\Delta t}{|\omega_c|} \sum_{p \in \mathcal{P}(c)} \sum_{f \in \mathcal{SF}(pc)} A_{pcf} \overline{\mathbf{F}}_{pcf} = \mathbf{0}.$$

whose main ingredients are

- A subface-based flux approximation computed as the left-sided flux, *i.e.*, $\overline{\mathbf{F}}_{pcf} = \mathbf{F}^{-}_{\mathbf{n}_{pcf}}$, resulting from the simple approximate Riemann solver $\mathbf{W}_{pcf}$, refer to section 3

$$\begin{aligned} \mathbf{F}^{-}_{\mathbf{n}_{pcf}} =& \mathbb{F}(\mathbf{U}^{n}_c) \mathbf{n}_{pcf} - \int_{-\infty}^{0} [\mathbf{W}_{pcf}(\mathbf{U}^{n}_c, \mathbf{U}^{n}_d, \xi) - \mathbf{U}^{n}_c] \, \mathrm{d}\xi, \\ =& \mathbb{F}(\mathbf{U}^{n}_c) \mathbf{n}_{pcf} - \Lambda^{-}_{l,f} (\mathbf{U}^{\star}_{l,f} - \mathbf{U}^{n}_c) - \Lambda^{-}_{0,f} (\mathbf{U}^{\star}_{r,f} - \mathbf{U}^{\star}_{l,f}) - \Lambda^{-}_{r,f} (\mathbf{U}^{n}_d - \mathbf{U}^{\star}_{r,f}), \end{aligned}$$

where $\Lambda_{l,f}$, $\Lambda_{0,f}$, $\Lambda_{r,f}$ are the wave speeds and $\mathbf{U}^{\star}_{l,f}$, $\mathbf{U}^{\star}_{r,f}$ are the intermediate states of the simple Riemann solver.

- A positivity preserving and entropy stable simple approximate Riemann solver $\mathcal{W}_{pcf}$ for which the intermediate states and the wave speeds are function of a unique scalar parameter which corresponds to the intermediate normal velocity $v^{\star}_{\mathbf{n}_{pcf}}$. The wave speeds are given explicitly in terms of the mass flux parameters $\lambda_{l,f}$ and $\lambda_{r,f}$

$$\Lambda_{l,f} = \mathbf{v}^{n}_{\mathbf{n}_{pcf},c} - \frac{\lambda_{l,f}}{\rho^{n}_{c}} = \mathbf{v}^{\star}_{\mathbf{n}_{pcf}} - \frac{\lambda_{l,f}}{\rho^{\star}_{l,f}}, \quad \Lambda_{0,f} = v^{\star}_{\mathbf{n}_{pcf}}, \quad \Lambda_{r,f} = \mathbf{v}^{\star}_{\mathbf{n}_{pcf}} + \frac{\lambda_{r,f}}{\rho^{\star}_{r,f}} = \mathbf{v}^{n}_{\mathbf{n}_{pcf},d} + \frac{\lambda_{r,f}}{\rho^{n}_{d}}.$$

  The mass flux parameters might be adjusted, refer to (16) and (17), to guarantee not only the positivity of the mass density and the specific internal energy of the intermediate states but also to ensure the entropy control. This in turns provides the ordering of the wave speeds, *i.e.*, $\Lambda_{l,f} \leq \Lambda_{0,f} \leq \Lambda_{r,f}$. However, this Riemann solver does not satisfy the classical HLL consistency condition. Therefore, the subface flux approximation is not unique and the resulting Finite Volume discretization is not conservative in the classical sense, *i.e.* in the face-based sense.

- A node-based condition conservation condition provides the restoration of the conservation property locally at the node, refer to Figure 4 writing that the sum over the subfaces impinging at the node of the jumps between the left and right-sided fluxes times the measure of the subface is equal to zero

$$\sum_{f \in \mathcal{SF}(p)} A_{pf}(\mathbf{F}^{+}_{\mathbf{n}_{pf}} - \mathbf{F}^{-}_{\mathbf{n}_{pf}}) = \mathbf{0}.$$

  This gives a system of equations to compute the intermediate normal velocity $v^{\star}_{\mathbf{n}_{pcf}}$ assuming that it is the projection of the nodal velocity onto the subface unit normal, *i.e.*, $v^{\star}_{\mathbf{n}_{pcf}} = \mathbf{v}_p \cdot \mathbf{n}_{pcf}$

- A nodal solver computes the nodal velocity $\mathbf{v}_p$ in a unique manner, refer to (44), in terms of all the states surrounding the node p.

- A subface-based flux expression is derived in terms of the "classical one-dimensional flux" plus a contribution originated from the discrepancy between the projection of the nodal velocity onto the subface unit normal and the normal velocity resulting from the one-dimensional Riemann solver, as

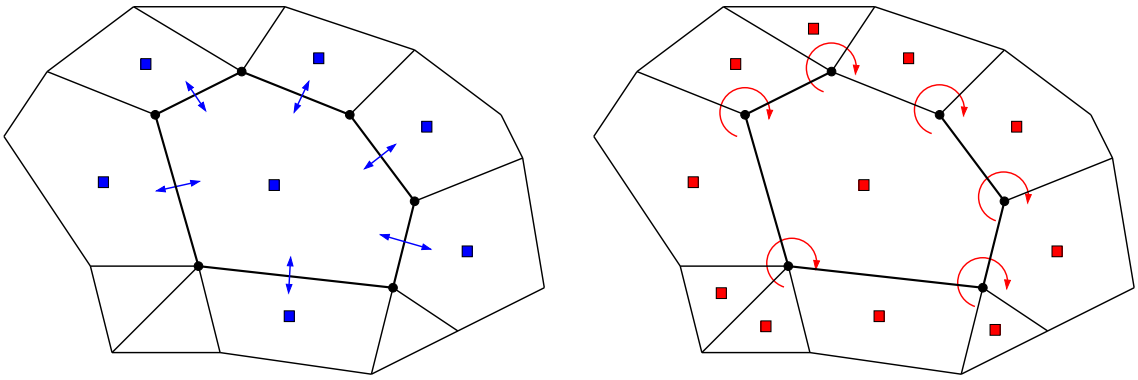
$$\mathbf{F}^{-}_{\mathbf{n}_{pcf}} = \mathbf{F}^{\star}_{\mathbf{n}_{pcf}} - \frac{1}{2}(\lambda_{l,f} + \lambda_{r,f})\left(v^{\star}_{\mathbf{n}_{pcf}} - \overline{v}_{\mathbf{n}_{pcf},lr}\right)\begin{pmatrix} 0 \\ 1 \\ \mathbf{0} \\ v^{\star}_{\mathbf{n}_{pcf}} \end{pmatrix},$$

  where the "classical one-dimensional flux" reads

$$\mathbf{F}^{\star}_{\mathbf{n}_{pcf}} = \frac{1}{2}\left(\mathbf{F}_{\mathbf{n}_{pcf},c} + \mathbf{F}_{\mathbf{n}_{pcf},d}\right) - \frac{1}{2}\left[|\Lambda_{l,f}|(\mathbf{U}^{\star}_{l,f} - \mathbf{U}^{n}_{c}) + |\Lambda_{0}|(\mathbf{U}^{\star}_{r,f} - \mathbf{U}^{\star}_{l,f}) + |\Lambda_{r,f}|(\mathbf{U}^{n}_{d} - \mathbf{U}^{\star}_{r,f})\right].$$

  Here, $\mathbf{F}_{\mathbf{n}_{pcf},c} = \mathbb{F}(\mathbf{U}^{n}_{c})\mathbf{n}_{pcf}$ and $\mathbf{F}_{\mathbf{n}_{pcf},d} = \mathbb{F}(\mathbf{U}^{n}_{d})\mathbf{n}_{pcf}$. Strictly speaking $\mathbf{F}^{\star}_{\mathbf{n}_{pcf}}$ is not the classical one-dimensional classical flux since the values of the intermediate states and the wave speeds depend directly on the nodal velocity through $v^{\star}_{\mathbf{n}_{pcf}} = \mathbf{v}_p \cdot \mathbf{n}_{pcf}$.

- Finally, an explicit time step condition, refer to (37) and (38) ensures that $\mathbf{U}^{n+1}_{c}$ is a convex combination of the intermediate states of the subface-based approximate Riemann problem. This in turn guarantees the positivity preserving and entropy stability properties of the Finite Volume method provided that the approximate Riemann solver ensures also these properties locally, which is indeed the case thanks to (16).



(a) Face-based stencil for classical FV.

(b) Node-based stencil for unconventional FV.

Figure 5: Fragement of a two-dimensional grid; Stencil of classical *two-point* and unconventional *multipoint* Finite Volume discretizations.

We conclude this summary of main features of the subface-based Finite Volume discretization by observing that the domain of influence of the cell ω_c, displayed in Figure 5b, consists of its neighboring cells sharing a node. This is the consequence of the direct contribution of the nodal velocity to the numerical approximation of the subface flux since the nodal velocity is computed from the nodal solver which depends on all the states surrounding the node under consideration. For this reason, the subface flux approximation is multidimensional in the sense that it is not only a function of the left and right states on both sides of the subface but also a function of all the states surrounding the node. This is why we shall refer to it as the **multipoint Finite Volume method**. On the other hand, the classical Finite Volume discretization relies on a one-dimensional interface flux approximation which is computed from the states located on both sides of the interface, *i.e.*, the left and the right-sided states, by means of an approximate Riemann solver. For this reason, we shall refer to it as the **two-point Finite Volume method**. In this case, the domain of influence of ω_c, displayed in Figure 5a, is composed of the neighboring cells sharing a face.

4.5. Variation around classical Finite Volume discretizations

For the sake of completeness, this subsection aims at describing the classical two-point Finite Volume discretization. Then, we shall present a robust variant of this solver relying on the introduction of a viscous dissipation on the tangential velocity component.

4.5.1. The classical two-point Finite Volume discretization

We derive this numerical method utilizing the simple Riemann solver which corresponds to the first case of the alternative presented at the end of section 3.3. Namely, in this case, the $v_{\mathbf{n}}^\star$ parameter is equal to the normal velocity of the interface $\overline{v}_{\mathbf{n},lr} = \frac{\lambda_l v_{\mathbf{n},l} + \lambda_r v_{\mathbf{n},r}}{\lambda_l + \lambda_r} - \frac{(p_r - p_l)}{\lambda_l + \lambda_r}$. Then, the structure of the approximate Riemann solver is completely determined in terms of the left and the right states on both sides of the interface. For the sake of completeness we write explicitly the main components of this solver. The wave speeds expressions are

$$\Lambda_l = \mathbf{v}_{\mathbf{n},l} - \frac{\lambda_l}{\rho_l} = \overline{v}_{\mathbf{n},lr} - \frac{\lambda_l}{\rho_l^\star}, \quad \Lambda_{0,f} = \overline{v}_{\mathbf{n},lr}, \quad \Lambda_r = \overline{v}_{\mathbf{n},lr} + \frac{\lambda_r}{\rho_r^\star} = \mathbf{v}_{\mathbf{n},r} + \frac{\lambda_r}{\rho_r}.$$

The intermediate states and fluxes of the Riemann solver write

$$\mathbf{U}_s^\star = \begin{pmatrix} \rho_s^\star \\ \rho_s^\star \overline{v}_{\mathbf{n},lr} \\ \rho_s^\star \mathbf{v}_{t,s}^\star \\ \rho_s^\star e_s^\star \end{pmatrix} \quad \text{and} \quad \mathbf{F}_{\mathbf{n},s}^\star = \overline{v}_{\mathbf{n},lr} \mathbf{U}_s^\star + \begin{pmatrix} 0 \\ p^\star \\ \mathbf{0} \\ p^\star \overline{v}_{\mathbf{n},lr} \end{pmatrix}, \text{ for } s = l, r.$$

The components of the intermediate states and fluxes are computed by solving the two following systems

$$(\mathcal{S}_l)\begin{cases}\lambda_l(\tau_l^\star-\tau_l)-(\overline{v}_{\mathbf{n},lr}-v_{\mathbf{n},l})=0,\\ \lambda_l(\overline{v}_{\mathbf{n},lr}-v_{\mathbf{n},l})+(p^\star-p_l)=0,\\ \lambda_l(\mathbf{v}_{t,l}^\star-\mathbf{v}_{t,l})=\mathbf{0},\\ \lambda_l(e_l^\star-e_l)+(p^\star\overline{v}_{\mathbf{n},lr}-p_lv_{\mathbf{n},l})=0,\end{cases}\quad\text{and}\quad(\mathcal{S}_r)\begin{cases}\lambda_r(\tau_r^\star-\tau_r)+(\overline{v}_{\mathbf{n},lr}-v_{\mathbf{n},r})=0,\\ \lambda_r(\overline{v}_{\mathbf{n},lr}-v_{\mathbf{n},r})-(p^\star-p_r)=0,\\ \lambda_r(\mathbf{v}_{t,r}^\star-\mathbf{v}_{t,r})=\mathbf{0},\\ \lambda_r(e_r^\star-e_r)-(p^\star\overline{v}_{\mathbf{n},lr}-p_rv_{\mathbf{n},r})=0.\end{cases}$$

The mass flux parameters are determined employing the conditions derived in [16] to ensure positivity and entropy stability. Finally, gathering all these informations, we take note that the HLL consistency condition is satisfied, which means that the left and the right-sided fluxes are rigorously equal, *i.e.*, $\mathbf{F}_{\mathbf{n}}^-=\mathbf{F}_{\mathbf{n}}^+$, and thus the numerical approximation of the cell-interface flux, refer to (26), is uniquely defined by

$$\mathbf{F}_{\mathbf{n}}^\star(\mathbf{U}_l,\mathbf{U}_r)=\frac{1}{2}\left(\mathbf{F}_{\mathbf{n}}^-+\mathbf{F}_{\mathbf{n}}^+\right)=\frac{1}{2}\left(\mathbf{F}_{\mathbf{n},l}+\mathbf{F}_{\mathbf{n},r}\right)-\frac{1}{2}\left[|\Lambda_l|(\mathbf{U}_l^\star-\mathbf{U}_l)+|\Lambda_0|(\mathbf{U}_r^\star-\mathbf{U}_l^\star)+|\Lambda_r|(\mathbf{U}_r-\mathbf{U}_r^\star)\right],$$

where $\mathbf{F}_{\mathbf{n},l}=\mathbb{F}(\mathbf{U}_l)\mathbf{n}$ and $\mathbf{F}_{\mathbf{n},r}=\mathbb{F}(\mathbf{U}_r)\mathbf{n}$. The resulting Finite Volume discretization reads

$$\mathbf{U}_c^{n+1}-\mathbf{U}_c^n+\frac{\Delta t}{|\omega_c|}\sum_{d\in\mathcal{NF}(c)}A_{cd}\mathbf{F}_{\mathbf{n}_{cd}}^\star(\mathbf{U}_c^n,\mathbf{U}_d^n)=\mathbf{0},\tag{45}$$

where $\mathcal{NF}(c)$ is the set of cells neighboring ω_c and sharing a face with it, A_{cd} is the measure of the face area shared by cells c and d. The unit normal outward to this face is denoted by $\mathbf{n}_{cd}$. This numerical scheme belongs to the class of two-point Finite Volume methods.

In this case, since $\mathbf{F}_{\mathbf{n}_{cd}}^\star=\mathbf{F}_{\mathbf{n}_{cd}}^-$, it is possible as in section 4.2 to derive an explicit condition on the time step, refer to (37) and (38) to ensure that $\mathbf{U}_c^{n+1}$ is a convex combination of $\mathbf{U}_c^n$ and the set of intermediate states of the Riemann problems attached to each face of the cell under consideration. The conjunction of this time step condition plus the conditions on the mass flux parameters to ensure the positivity preserving and the entropy stability of the Riemann solver results in a positivity preserving and entropy stable Finite Volume scheme.

4.5.2. The modified two-point Finite Volume discretization

The two-point Finite Volume discretization is characterized by a numerical dissipation that is face-based and essentially governed by the jumps of the pressure and the normal velocity between the left and right states on both sides of a given interface. This might be a reason to explain its sensitivity to the odd-even decoupling and to the carbuncle instabilities [46], whereas the multipoint method seems to be insensitive to these instabilities, as we shall see in section 8.2 and in section 8.5. The main difference between those two approaches lies in the node-based feature of the numerical dissipation embedded in the multipoint method, which is thus multidimensional by construction. Following, Rodionov [48], we might conjecture that the multidimensional numerical dissipation should be a pertinent remedy to eradicate these spurious numerical instabilities. However, we do not have any theoretical arguments to explain this conjecture which is uniquely based on numerical experiments.

In this paragraph, we present a modification of the two-point method, which appears to enhance its robustness by simply adding a supplementary viscous dissipation proportional to the shearing and we call it the modified two-point method. We describe hereafter the main ingredients of this modification. The modified two-point method relies on the variant of the approximate Riemann solver described in the foregoing paragraph, which consists in assuming the continuity of the tangential velocity for the intermediate state and in adding a viscous force in the tangential velocity and the total energy components for the intermediate fluxes. We acknowledge that similar modifications have been already proposed for instance in [52, 8]. Here, the modification we propose relies on the simple approximate Riemann solver formalism and is thus characterized by a straightforward wave speeds definition and also the positivity and entropy stability properties obtained by tuning the mass flux parameters.

Now, it remains to construct the proposed modification. First, the intermediate states and fluxes write

$$\mathbf{U}_s^\star = \begin{pmatrix} \rho_s^\star \\ \rho_s^\star \overline{v}_{\mathbf{n},lr} \\ \rho_s^\star \mathbf{v}_t^\star \\ \rho_s^\star e_s^\star \end{pmatrix} \quad \text{and} \quad \mathbf{F}_{\mathbf{n},s}^\star = \overline{v}_{\mathbf{n},lr}\mathbf{U}_s^\star + \begin{pmatrix} 0 \\ p^\star \\ \mathbf{S}^\star \\ p^\star \overline{v}_{\mathbf{n},lr} + \mathbf{S}^\star \cdot \mathbf{v}_t^\star \end{pmatrix}, \text{ for } s = l, r.$$

Here, the novel ingredients have been written using red color. These are the viscous force $\mathbf{S}^\star$ which acts on the tangential component of the velocity and its corresponding work rate, $\mathbf{S}^\star \cdot \mathbf{v}_t^\star$, in the total energy component. As usual, the conservation relations written respectively for the Λ_l-wave and the Λ_r-wave provide us 2 systems of equations which allow to compute the intermediate states. These systems read

$$(\mathcal{S}_l) \begin{cases} \lambda_l(\tau_l^\star - \tau_l) - (\overline{v}_{\mathbf{n},lr} - v_{\mathbf{n},l}) = 0, \\ \lambda_l(\overline{v}_{\mathbf{n},lr} - v_{\mathbf{n},l}) + p^\star - p_l = 0, \\ \lambda_l(\mathbf{v}_t^\star - \mathbf{v}_{t,l}) + \mathbf{S}^\star = \mathbf{0}, \\ \lambda_l(e_l^\star - e_l) + p^\star \overline{v}_{\mathbf{n},lr} - p_l v_{\mathbf{n},l} + \mathbf{S}^\star \cdot \mathbf{v}_t^\star = 0, \end{cases} \qquad (\mathcal{S}_r) \begin{cases} \lambda_r(\tau_r^\star - \tau_r) + \overline{v}_{\mathbf{n},lr} - v_{\mathbf{n},r} = 0, \\ \lambda_r(\overline{v}_{\mathbf{n},lr} - v_{\mathbf{n},r}) - (p^\star - p_r) = 0, \\ \lambda_r(\mathbf{v}_t^\star - \mathbf{v}_{t,r}) - \mathbf{S}^\star = \mathbf{0}, \\ \lambda_r(e_r^\star - e_r) - (p^\star \overline{v}_{\mathbf{n},lr} - p_r v_{\mathbf{n},r}) - \mathbf{S}^\star \cdot \mathbf{v}_t^\star = 0. \end{cases}$$

Summing the equations related to the left and the right-sided tangential components of the velocity leads to

$$\mathbf{v}_t^\star = \frac{\lambda_l \mathbf{v}_{t,l} + \lambda_r \mathbf{v}_{t,r}}{\lambda_l + \lambda_r}, \text{ and } \mathbf{S}^\star = -\frac{\lambda_l \lambda_r}{\lambda_l + \lambda_r}(\mathbf{v}_{t,r} - \mathbf{v}_{t,l}).$$

The viscous force $\mathbf{S}^\star$ is proportional to the jump of the tangential velocities between the left and the right states and thus boils down to zero in the case of a uniform flow, which ensures the consistency of the modification with the underlying physical model. It remains to assess the impact of the presence of the viscous force into the energy balance. To this end, we need to compute the internal energy equation for both the left and the right states. Recalling $\varepsilon_s^\star = e_s^\star - \frac{1}{2}\left[\overline{v}_{\mathbf{n},lr}^2 + (\mathbf{v}_t^\star)^2\right]$ and noticing that

$$\varepsilon_s^\star - \varepsilon_s = e_s^\star - e_s + \frac{1}{2}\left(\overline{v}_{\mathbf{n},lr} + v_{\mathbf{n},s}\right)\left(\overline{v}_{\mathbf{n},lr} - v_{\mathbf{n},s}\right) + \frac{1}{2}\left(\mathbf{v}_t^\star + \mathbf{v}_{t,s}\right)\left(\mathbf{v}_t^\star - \mathbf{v}_{t,s}\right),$$

we arrive at the internal energy equations

$$\varepsilon_l^\star - \varepsilon_l + \frac{1}{2}(p^\star + p_l)(\tau_l^\star - \tau_l) = \frac{1}{2}\frac{\lambda_r^2}{(\lambda_l + \lambda_r)^2}\left(\mathbf{v}_{t,r} - \mathbf{v}_{t,l}\right)^2,$$
$$\varepsilon_r^\star - \varepsilon_r + \frac{1}{2}(p^\star + p_r)(\tau_r^\star - \tau_r) = \frac{1}{2}\frac{\lambda_l^2}{(\lambda_l + \lambda_r)^2}\left(\mathbf{v}_{t,r} - \mathbf{v}_{t,l}\right)^2.$$

For both equations the right hand-side might be viewed as an extra viscous dissipation term resulting from the shearing. Denoting by Q_s the right hand-side for $s = l, r$, let us investigate the modification induced by the presence of this source term with respect to the condition ensuring the positivity of the intermediate specific internal energy. Proceeding similarly to what has been done in [16], we arrive at the condition

$$\lambda_s^2 \geq \frac{p_s^2}{2\left(\varepsilon_s + Q_s\right)}, \quad \text{for} \quad s = l, r. \tag{46}$$

The positivity of the intermediate internal energy, *i.e.*, $\varepsilon_s^\star > 0$, is ensured with a less restrictive value of λ_s since $Q_s \geq 0$.

5. Boundary conditions for the multipoint scheme

In this section, we investigate the treatment of the boundary conditions. To do so, let us first split the set of subfaces around a node $\mathcal{SF}(p)$, into two new sets, the set of interior subfaces around the node, $\mathcal{ISF}(p)$, and the set of boundary subfaces around the node, $\mathcal{BSF}(p)$, so that

$$\mathcal{SF}(p) = \mathcal{ISF}(p) \sqcup \mathcal{BSF}(p), \quad \forall p \in \mathcal{P}.$$

By convention, a boundary subface normal is always going outwards of the domain so that a boundary subface only has a left neighbor and is *missing* a right neighbor. Let us rewrite the node based conservation (44) as

$$\sum_{f\in\mathcal{ISF}(p)} A_{pf}\{\lambda_r+\lambda_l\}_{pf}(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\overline{v}_{\mathbf{n},pf})\mathbf{n}_{pf} + \sum_{f\in\mathcal{BSF}(p)} A_{pf}\{\lambda_r+\lambda_l\}_{pf}\left(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\left\{\frac{\lambda_r v_{\mathbf{n},r}+\lambda_l v_{\mathbf{n},l}}{\lambda_r+\lambda_l}-\frac{p_r-p_l}{\lambda_r+\lambda_l}\right\}_{pf}\right)\mathbf{n}_{pf}=\mathbf{0},$$

where we have freedom in the choice of the fictitious right states of the boundary faces, that is, $\lambda_{pf,r}$, $\rho_{pf,r}$, $\mathbf{v}_{pf,r}$, and $p_{pf,r}$ for all $f\in\mathcal{BSF}(p)$.

5.1. Slip-wall boundary condition

We show in this section how slip-wall boundary conditions are imposed in a simple manner. A different way of imposing slip-wall boundary conditions inspired from Lagrangian schemes is shown in [43]. The main difficulty comes from the definition of the star velocity on the faces around a node, that is

$$v^{\star}_{\mathbf{n},pf}=\mathbf{v}_p\cdot\mathbf{n}_{pf},\quad\forall f\in\mathcal{SF}(p).$$

From this equation, and realizing that the mass flux through a face depends directly on the sign of the star velocity $v^{\star}_{\mathbf{n},pf}$, one can see that the only way for the mass flux on boundary subfaces to be zero is that $\mathbf{v}_p=\mathbf{0}$. This would mean that adherence boundary conditions are used rather than slip-wall conditions, and also, would make the scheme non conservative for the other subfaces around the node because there is no reason that $\mathbf{v}_{pf}=\mathbf{0}$ is the solution to system (44). One way to solve this issue is to change the definition of the star velocities on boundary nodes in the following manner

$$v^{\star}_{\mathbf{n},pf}=\mathbf{v}_p\cdot\mathbf{n}_{pf},\quad\forall f\in\mathcal{ISF}(p),\quad\text{and}\quad v^{\star}_{\mathbf{n},pf}=\overline{v}_{\mathbf{n},pf},\quad\forall f\in\mathcal{BSF}(p).$$

In this way, the flux through wall boundary subfaces corresponds to the flux of the two-point scheme that can be imposed in the classical manner by setting

$$\rho_{pf,r}=\rho_{pf,l},\quad v_{\mathbf{n},pf,r}=-v_{\mathbf{n},pf,l},\quad\mathbf{v}_{\mathbf{t},pf,r}=\mathbf{v}_{\mathbf{t},pf,l},\quad\text{and}\quad,\quad p_{pf,r}=p_{pf,l},\quad\forall f\in\mathcal{BSF}(p).$$

The consequence is that $v^{GOD}_{\mathbf{n},pf}=0,\quad\forall f\in\mathcal{BSF}(p)$, and this ensures that there is no mass flux through boundary subfaces. Finally, the nodal system (44) can be written for a boundary node as

$$\sum_{f\in\mathcal{ISF}(p)} A_{pf}\{\lambda_r+\lambda_l\}_{pf}\left(\mathbf{v}_p\cdot\mathbf{n}_{pf}-\overline{v}_{\mathbf{n},pf}\right)\mathbf{n}_{pf}=\mathbf{0},\quad\forall p\in\mathcal{P},$$

which is a 3×3 system at node p, that can be written as

$$\mathbb{M}_p\mathbf{v}_p=\mathbf{R}_p,$$

with

$$\mathbb{M}_p=\sum_{f\in\mathcal{ISF}(p)}\{\lambda_r+\lambda_l\}_{pf}(\mathbf{n}_{pf}\otimes\mathbf{n}_{pf}),\quad\text{and}\quad\mathbf{R}_p=\sum_{f\in\mathcal{ISF}(p)}\{\lambda_r+\lambda_l\}_{pf}\,\overline{v}_{\mathbf{n},pf}\mathbf{n}_{pf}.$$

Notice that the only difference with system (44) is that now it is only written for the interior subfaces around a node. The consequence is that the matrix $\mathbb{M}_p$ is not always invertible. For example, on a Cartesian mesh of a cube, the interior subfaces around a boundary node in the middle of one of the faces would only span $\mathbb{R}^2$. This issue is however easily resolved as the nodal velocity $\mathbf{v}_p$ would only need to be projected onto this space. This means that we only need a pseudo inverse of matrix $\mathbb{M}_p$ to get a solution for $\mathbf{v}_p$. In this way, we will have a unique solution for the velocities $v^{\star}_{\mathbf{n},pf},\quad\forall f\in\mathcal{ISF}(p)$ that satisfies the node based conservation condition.

5.2. Freestream boundary condition

For an freestream boundary condition, a specific state $\mathbf{W}^{FS} = (\rho^{FS}, \mathbf{v}^{FS}, p^{FS})^t$ is prescribed on the boundary. This state is used for the *fictitious* right states of the boundary faces as

$$\mathbf{W}_{f,r} = \mathbf{W}^{FS}, \qquad \forall f \in \mathcal{BSF}(p).$$

5.3. Outflow boundary condition

We are only concerned here by supersonic outflows which are taken into account by using symmetric boundaries, that is

$$\mathbf{W}_{f,r} = \mathbf{W}_{f,l}, \qquad \forall f \in \mathcal{BSF}(p).$$

6. Second-order extension

This section briefly describes the second-order extension of our subface-based Finite Volume approach. Second-order accuracy in space is achieved reconstructing piece-wise polynomials within every mesh cell. The reconstructed solution $\tilde{\mathbf{U}}_c$ within cell ω_c write

$$\tilde{\mathbf{U}}_c(\mathbf{x}) = \mathbf{U}_c + (\nabla\tilde{\mathbf{U}})_c(\mathbf{x} - \mathbf{x}_c),$$

where $\mathbf{x}_c$ is the cell centroid and thus this reconstruction is conservative. The cell-centered gradients $(\nabla\tilde{\mathbf{U}})_c$ are approximated using a discrete gradient operator rather than using a classical least-squares reconstruction method. A comparison of the least-squares reconstruction method and a discrete operator approach similar to what we present can be found in [59]. Using the local gradient operators does not require the storage of any reconstruction matrix required by the least-squares approach, and thus results in a more memory efficient implementation, especially for three-dimensional unstructured grids. Once the cell-centered gradients have been computed, slope limiting is applied to ensure the monotonicity and the physical admissibility of the reconstructed states. The second-order flux at the subfaces is approximated employing the extrapolated values at the nodes resulting from the reconstructed variables. The subface based finite volumes scheme can then be written as

$$|\omega_c|\frac{\mathrm{d}\mathbf{U}_c}{\mathrm{d}t} + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\bar{\mathbf{F}}_{pcf}(\tilde{\mathbf{U}}_c(\mathbf{x}_p), \tilde{\mathbf{U}}_d(\mathbf{x}_p)) = \mathbf{0}. \tag{47}$$

Time discretization is performed as in the MUSCL-Hancock method integrating (47) in time to get

$$|\omega_c|\left(\mathbf{U}_c^{n+1} - \mathbf{U}_c^n\right) + \sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\int_{t^n}^{t^{n+1}} \bar{\mathbf{F}}_{pcf}(\tilde{\mathbf{U}}_c(\mathbf{x}_p), \tilde{\mathbf{U}}_d(\mathbf{x}_p))\,\mathrm{dt} = \mathbf{0}.$$

To obtain the second-order accuracy in time, we simply use the mid-point quadrature rule for approximating the flux integral

$$|\omega_c|\left(\mathbf{U}_c^{n+1} - \mathbf{U}_c^n\right) + \Delta t\sum_{p\in\mathcal{P}(c)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\bar{\mathbf{F}}_{pcf}(\tilde{\mathbf{U}}_c^{n+1/2}(\mathbf{x}_p), \tilde{\mathbf{U}}_d^{n+1/2}(\mathbf{x}_p)) = \mathbf{0}.$$

The reconstructed values $\tilde{\mathbf{U}}_c^{n+1/2}(\mathbf{x}_p)$ and $\tilde{\mathbf{U}}_d^{n+1/2}(\mathbf{x}_p)$ correspond to the values of the reconstructed solution from the left and right elements evaluated at $t^n + \frac{\Delta t}{2}$ *via* the Taylor expansion

$$\tilde{\mathbf{U}}_c^{n+1/2} = \tilde{\mathbf{U}}_c^n + \frac{\Delta t}{2}\left(\frac{\partial\tilde{\mathbf{U}}}{\partial t}\right)_c^n.$$

Regarding the multi-point scheme, the one-step time discretization is particularly advantageous since only one nodal system will be solved at each node for each time step contrarily to a two-step Runge-Kutta approach. In the remaining of this section, we describe the cell-centered gradients approximation using local discrete operators and the slope limiting of the reconstructed variables. We also describe the approximation of the solution discrete time derivative.

6.1. Construction of the cell-centered gradient approximation

First, utilizing the framework of compatible spatial discretization introduced by Shashkov and his collaborators [13], we define the discrete gradient operator of the scalar field ϕ at the node p

$$(\nabla\phi)_p = -\frac{1}{|\omega_p|}\sum_{c\in\mathcal{C}(p)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}\phi_c\mathbf{n}_{pcf}, \tag{48}$$

where ω_p defined by (28) denotes the dual cell attached to node p and $C(p)$ is the set of cells surrounding node p. This is a quite parsimonious formula since a single loop through the elements surrounding the node is needed for evaluating the discrete gradient and thus no linear system has to be solved.

Remark 6.1 (Expression of the nodal velocity in terms of the nodal discrete pressure gradient). *The local system* (44) *used to compute the nodal velocity in the multi-point scheme can be rewritten*

$$\left[\sum_{c\in\mathcal{P}(c)}\mathbb{M}_{pc}\right]\mathbf{v}_p = \sum_{c\in\mathcal{P}(c)}\mathbb{M}_{pc}\mathbf{v}_c + \sum_{c\in\mathcal{C}(p)}\sum_{f\in\mathcal{SF}(pc)} A_{pcf}p_c\mathbf{n}_{pcf},$$

where $\mathbb{M}_{pc} = \sum_{f\in\mathcal{SF}(pc)} A_{pcf}\lambda_{pcf}\mathbf{n}_{pcf}\otimes\mathbf{n}_{pcf}$ *is the viscous corner matrix which is symmetric positive definite. Employing the foregoing discrete gradient operator definition* (48)*, the nodal velocity expression turns into*

$$\left[\sum_{c\in\mathcal{P}(c)}\mathbb{M}_{pc}\right]\mathbf{v}_p = \sum_{c\in\mathcal{P}(c)}\mathbb{M}_{pc}\mathbf{v}_c - |\omega_p|(\nabla p)_p.$$

This shows that the nodal velocity expression is decomposed into two parts: the node-centered average of the surrounding cell-centered velocities and a term proportional to the nodal pressure gradient. This decomposition can be interpreted as a multidimensional extension of the classical Godunov interface velocity.

Finally, we define the cell-centered gradient approximation as the convex combination of the nodal discrete gradient operators defined at each node of the targeted cell

$$(\nabla\phi)_c = \frac{1}{|\tilde{\omega_c}|}\sum_{p\in\mathcal{P}(c)}|\omega_p|(\nabla\phi)_p, \tag{49}$$

where $|\tilde{\omega_c}| = \sum_{p\in\mathcal{P}(c)}|\omega_p|$. Replacing the expression of the nodal gradient operator (48) into (49) leads to the cell-centered discrete gradient operator

$$(\nabla\phi)_c = -\frac{1}{|\tilde{\omega_c}|}\sum_{p\in\mathcal{P}(c)}|\omega_p|\sum_{d\in\mathcal{C}(p)}\sum_{f\in\mathcal{SF}(pd)} A_{pdf}\phi_d\mathbf{n}_{pdf}. \tag{50}$$

The stencil of the cell-centered discrete gradient operator (50) is the set of the cells sharing at least a node with cell ω_c. From (49), it is clear that the cell-centered gradient is a linear combination of the nodal gradients which shall obviously result in an oscillatory solution if let unlimited. Slope limiting is required to ensure monotonicity and physical admissibility of the solution. In the next paragraph, we describe a classical monotonicity procedure based on slope limiting knowing that in the future, we intend to develop an ENO/WENO-based [53] limiting procedure which stems from a non-linear weighted combination of the nodal gradients.

6.2. Slope limiting to enforce both physical admissibility and monotonicity

Once the reconstructed gradients have been computed, we need to ensure that the reconstructed variables in the cells are physically admissible. At least, we want the nodal extrapolated values to remain

physically admissible, *e.g.*, the nodal extrapolated mass density and specific internal energy shall remain strictly positive which reads

$$\tilde{\rho}_c(\mathbf{x}_p) > 0, \quad \forall p \in \mathcal{P}(c), \quad \text{and} \quad \tilde{p}_c(\mathbf{x}_p) > 0, \quad \forall p \in \mathcal{P}(c). \tag{51}$$

This is a strict requirement to make sure that the Riemann solver shall be fed with a physical solution. However, this does not ensure in any way that the solution will be free of numerical oscillations. The conditions from equation (51) are easily imposed by limiting the reconstructed gradient by means of the scalar parameter $\alpha_c \in [0, 1]$

$$\tilde{\mathbf{U}}_c = \mathbf{U}_c + \alpha_c (\nabla \tilde{\mathbf{U}})_c (\mathbf{x}_d - \mathbf{x}_c), \tag{52}$$

Replacing the limited reconstruction (52) into (51) leads to

$$\rho_c + \alpha_c^{\rho} \nabla \tilde{\rho}_c \cdot (\mathbf{x}_i - \mathbf{x}_c) > 0, \quad \forall \mathbf{x}_i \in \omega_c,$$

from which we can easily find that

$$\alpha_c^{\rho} = \min_{p \in \mathcal{P}(c)} \alpha_{pc}^{\rho}, \quad \text{where} \quad \alpha_{pc}^{\rho} = \begin{cases} \min\left(1, \frac{-\rho_c}{\nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) \leq 0, \\ 1, & \text{if} \quad \nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) > 0. \end{cases}$$

If we choose to reconstruct the primitives variables, we can proceed in a similar manner for the reconstructed pressure

$$\alpha_c^{p} = \min_{p \in \mathcal{P}(c)} \alpha_{pc}^{p}, \quad \text{where} \quad \alpha_{pc}^{p} = \begin{cases} \min\left(1, \frac{-p_c}{\nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) \leq 0, \\ 1, & \text{if} \quad \nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) > 0. \end{cases}$$

The global parameter α_c is then simply found by taking the minimum between α_c^{ρ} and α_c^{p}. Now, to prevent the occurence of spurious oscillations, we shall impose that the reconstructed solution cannot be a local extremum for the density and pressure components

$$\alpha_c^{\rho} = \min_{p \in \mathcal{P}(c)} \alpha_{pc}^{\rho}, \quad \text{where} \quad \alpha_{pc}^{\rho} = \begin{cases} \min\left(1, \frac{\rho_p^{\min} - \rho_c}{\nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) \leq 0, \\ \min\left(1, \frac{\rho_p^{\max} - \rho_c}{\nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{\rho}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) > 0, \end{cases}$$

and

$$\alpha_c^{p} = \min_{p \in \mathcal{P}(c)} \alpha_{pc}^{p}, \quad \text{where} \quad \alpha_{pc}^{p} = \begin{cases} \min\left(1, \frac{p_p^{\min} - p_c}{\nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) \leq 0, \\ \min\left(1, \frac{p_p^{\max} - p_c}{\nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c)}\right), & \text{if} \quad \nabla \tilde{p}_c \cdot (\mathbf{x}_p - \mathbf{x}_c) > 0. \end{cases}$$

The minimum and maximum of density and pressure around the cell are found by going through every neighbour of the targeted cell as

$$\rho_c^{\min} = \min_{d \in C(c)} \rho_d, \quad \text{and} \quad \rho_c^{\max} = \max_{d \in C(c)} \rho_d, \tag{53}$$

and similarly for the pressure

$$p_p^{\min} = \min_{d \in C(c)} p_d, \quad \text{and} \quad p_p^{\max} = \max_{d \in C(c)} p_d. \tag{54}$$

Remark that even though we do not impose any explicit limitation criterion on the velocity, it will still be limited because we choose to multiply every gradient on cell ω_c by the same factor α_c. A similar process can be used while reconstructing the conservative variables to ensure that the density and internal energy remain positive to induce a limiting on the velocity, refer to [29].

6.3. Fully discrete MUSCL-Hancock scheme

We now turn our attention to the fully discrete MUSCL-Hancock scheme, where we need to evaluate the reconstructed solution at time $t^{n+1/2}$ as

$$\tilde{\mathbf{U}}_c^{n+1/2} \approx \tilde{\mathbf{U}}_c^n + \frac{\Delta t}{2}\left(\frac{\partial \tilde{\mathbf{U}}}{\partial t}\right)_c^n. \tag{55}$$

Following the usual method, we use the original multi-dimensional conservation law to exchange the time derivative in equation (55) for the divergence of the flux. Equation (55) is rewritten as

$$\tilde{\mathbf{U}}_c^{n+1/2} \approx \tilde{\mathbf{U}}_c^n - \frac{\Delta t}{2}(\nabla \cdot \mathbb{F}(\tilde{\mathbf{U}}))_c^n. \tag{56}$$

We now need to construct an approximation of the divergence of the flux inside cell ω_c. We will follow the same steps as for the derivation of the nodal gradient operators, by now integrating the divergence of the flux over cell ω_c to get

$$(\nabla \cdot \mathbb{F}(\tilde{\mathbf{U}}))_c^n = \frac{1}{|\omega_c|}\int_{\partial\omega_c} \mathbb{F}(\tilde{\mathbf{U}}^n)\mathbf{n}\,\mathrm{ds}. \tag{57}$$

The surface integral can be split using the subface formalism to get the final form of the discrete divergence of the flux over cell ω_c as

$$(\nabla \cdot \mathbb{F}(\tilde{\mathbf{U}}))_c \approx \frac{1}{|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{g\in\mathcal{SF}(pc)} A_{pcg}\mathbb{F}(\tilde{\mathbf{U}}_c^n(\mathbf{x}_p))\mathbf{n}_{pcg}. \tag{58}$$

Finally, the reconstructed value $\tilde{\mathbf{U}}_c^{n+1/2}(\mathbf{x}_p)$ from equation (56) writes

$$\tilde{\mathbf{U}}_c^{n+1/2}(\mathbf{x}_p) \approx \tilde{\mathbf{U}}_c^n(\mathbf{x}_p) - \frac{\Delta t}{2|\omega_c|}\sum_{p\in\mathcal{P}(c)}\sum_{g\in\mathcal{SF}(pc)} A_{pcg}\mathbb{F}(\tilde{\mathbf{U}}_c^n(\mathbf{x}_p))\mathbf{n}_{pcg}. \tag{59}$$

As usual with fully discrete methods, the update (59) is entirely contained inside cell ω_c and is fully explicit. This scheme achieves second-order of accuracy in both space and time, without the global intermediate state used in the semi-discrete Runge-Kutta method. In the multi-dimensional setting, we need to compute the discrete divergence of the flux inside each cell of the mesh as defined in equation (58) which can be costly, but has proven more robust than using multi-slope one-dimensional approximations.

7. Mesh generation and parallel implementation

Unstructured mesh generation in 2D and 3D is performed using Gmsh a three-dimensional finite element mesh generator [23]. The meshes produced by Gmsh can be made of several polyhedra; tetrahedra, hexaedra, prism, pyramid in particular. For the 3D application, the computational power required is important both in terms of memory and computation time. These problems are overcome in the typical manner with the use of a domain decomposition and distributed memory parallelization through the Message-passing Interface (MPI) standard, specifically, Open MPI [20]. Gmsh [23] allows for a domain decomposition to be performed using the METIS library [35]. It remains to add ghost-cells to each subdomain in accordance to the stencil of our scheme. The second-order multipoint scheme needs two layers of ghost cells computed via the vertices (meaning that cells that have a vertex in common are considered neighbors). Although Gmsh allows for one layer of ghost cells to be computed in this way, a custom modification is needed to compute the second layer. At last, another modification allows Gmsh to write for each subdomain the list of ghost cells and their partition of origin, but also the list of interior cells of the subdomain that are ghost cells in other subdomains. With all this information, the MPI memory exchange can easily be carried on using non-blocking communications and send/receive buffers. Because we are dealing with 3D unstructured meshes,

and we need two layers of ghost cells computed via the vertices, there is a large amount of communication between the MPI processes. Although one could simply use one MPI process per CPU core, the experiment showed that the number of subdomains needed and the increase in communication greatly hinders performance. A solution to diminish the communication costs is to reduce the number of subdomains as much as possible by only using one MPI process per CPU socket or even per compute node. Then, some kind of shared-memory parallel programming model like Open MP [17] is used to fully leverage the compute nodes. The experiment showed that this hybrid MPI/Open MP approach used less memory than the pure MPI approach and was marginally faster in all test cases. Because fewer subdomains are used, it also made the post-processing of the results more manageable. For a discussion on hybrid MPI/Open MP approaches, see [47]. This hybrid approach can also be modified in the future by using Open ACC or CUDA to make use of GPUs.

8. Numerical test cases

In this section, we run a set of representative test cases to demonstrate the performance of the three-dimensional version of the subface-based Finite Volume scheme. Comparisons are done between the usual *two-point scheme*, the *modified two-point scheme* and the multi-dimensional *multi-point scheme*. Recall that the *two-point scheme* is obtained by setting the contact discontinuity velocity to the usual Godounov solvers $v_{\mathbf{n}}^{GOD}$. The *two-point scheme* corresponds to an HLLC scheme with different wave speeds than those in [57], and is used to compute the flux through the subfaces instead of the faces. The fact that the two-point scheme is used on the subfaces instead of on the faces does not make any difference in the quality of the solution at first-order but takes longer to compute than the usual face based version. At second-order however, because the variables are reconstructed at the nodes of the cells, the solution obtained with the two-point scheme on the subfaces is not the same as the face based version. In fact the second-order solution of the subface version is more accurate than the face version because the flux through a face is approximated using different reconstructed values. The *modified two-point scheme* is essentially the HLLCM scheme from [52] but with different wave speeds estimation and is also run using the subface version with second-order of accuracy achieved through nodal reconstruction. The first- and second-order accurate versions of the schemes are compared for a set of relevant test cases. The CFL number is set to 0.5 for all cases otherwise noticed.

8.1. Stationary isentropic vortex

The isentropic vortex problem [53] tests the accuracy of numerical methods since an exact, smooth and analytic solution exists. The ambient flow is characterized by $\rho_\infty = 1$, $u_\infty = 0$, $v_\infty = 0$, $w_\infty = 0$ and $p_\infty = 1$, with a normalized ambient temperature $\theta_\infty = 1$. At the initial time $t = 0$, onto this ambient flow is superimposed a vortex centered at $(0, 0)$ with the following state: $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $\theta = \theta_\infty + \delta\theta$, where the increments are given by

$$
\begin{aligned}
\delta u &= -y' \frac{\beta}{2\pi} \exp\left(\frac{1 - r^2}{2}\right), \\
\delta v &= x' \frac{\beta}{2\pi} \exp\left(\frac{1 - r^2}{2}\right), \\
\delta w &= 0, \\
\delta\theta &= -\frac{(\gamma - 1)\beta^2}{8\gamma\pi^2} \exp\left(1 - r^2\right),
\end{aligned}
$$

with $r = \sqrt{x^2 + y^2 + z^2}$. The so-called strength of the vortex is set to $\beta = 5.0$, the density is given by $\rho = \rho_\infty \, (\theta/\theta_\infty)^{\frac{1}{\gamma - 1}}$, and the pressure by $p = \rho^\gamma$. At final time $t = t_{\text{final}} = 1$ the vortex should be identical to the initial solution. Because the solution is smooth one removes the slope limiter in order to observe

the optimal orders of convergence. In the next sections, we run the isentropic vortex test case with first- and second-order extension for both two-point and multi-point schemes on structured and unstructured 3-dimensionnal meshes. The density error of the solution $(\rho_c^{n+1})_c$ is computed in L^2 norm as

$$\varepsilon_2^N = \sum_c |\omega_c| \, \|\rho_c^{n+1} - \rho(x_c, y_c, z_c)\|^2,$$

where $\rho(x, y, z)$ is the initial density given in the previous subsection. The experimental order of convergence is computed between two successive refinements as $\log(\varepsilon_2^N/\varepsilon_2^{N'})/\log(N/N')$.

8.1.1. Emulating 2D case with one layer of hexahedra

The computational domain is $\Omega = [-5, 5] \times [-5, 5] \times [0, 0.025]$. The results of the convergence analysis for both two-point and multi-point schemes are displayed in Table 1 and show that the expected orders of convergence is reached. Notice that the error of the multi-point schemes is about 1.2 (resp. 5) times larger than the error produced by the modified two-point scheme for the hexedral (resp. tetrahedral) mesh.

		Two-point			**Modified Two-point**			**Multi-point**		
	h	L^2 error	L^2 order	Time (s)	L^2 error	L^2 order	Time (s)	L^2 error	L^2 order	Time (s)
1^{st}-order	$6.0e-2$	$8.72e-3$	–	6	$1.20e-2$	–	6	$1.38e-2$	–	10
	$3.0e-2$	$4.57e-3$	0.93	24	$6.36e-3$	0.92	26	$7.44e-3$	0.89	41
	$2.0e-2$	$3.09e-3$	0.96	55	$4.34e-3$	0.94	60	$5.08e-3$	0.93	94
	$1.5e-2$	$2.34e-3$	0.97	98	$3.28e-3$	0.97	105	$3.85e-3$	0.96	167
2^{nd}-order	$6.0e-2$	$2.95e-4$	–	13	$3.06e-4$	–	14	$4.28e-4$	–	17
	$3.0e-2$	$7.61e-5$	1.96	54	$7.61e-5$	2.01	54	$9.38e-5$	2.19	69
	$2.0e-2$	$3.42e-5$	1.97	120	$3.40e-5$	1.98	126	$3.98e-5$	2.11	160
	$1.5e-2$	$1.93e-5$	1.98	208	$1.92e-5$	1.98	220	$2.19e-5$	2.08	282

Table 1: Isentropic vortex – L^2 norm errors on density ρ between the numerical solution and the exact solution of the **stationary** isentropic vortex problem at $t_{\text{final}} = 1$ on **a mesh with one layer of hexahedra in the z direction**.

8.1.2. Genuine 3D case using tetrahedra

The same test case is simulated on increasingly refined meshes consisting of only tetrahedra. The results of the convergence analysis for both two-point and multi-point schemes are displayed in Table 2 and show that the expected rates of convergence are attained. Notice that the error of the multi-point schemes is about 1.2 (resp. 5) times larger than the error produced by the modified two-point scheme for the hexedral (resp. tetrahedral) mesh.

		Two-point			**Modified Two-point**			**Multi-point**		
	h	L^2 error	L^2 order	Time (s)	L^2 error	L^2 order	Time (s)	L^2 error	L^2 order	Time (s)
1^{st}-order	$7.2e-2$	$1.40e-1$	–	644	$2.12e-1$	–	643	$4.93e-1$	–	814
	$5.4e-2$	$1.05e-1$	0.98	1800	$1.63e-1$	0.90	1877	$4.12e-1$	0.61	2318
	$3.6e-2$	$7.13e-2$	0.97	10765	$1.12e-1$	0.93	10828	$3.09e-1$	0.72	13746
	$2.9e-2$	$5.74e-2$	0.97	25212	$9.11e-2$	0.94	26030	$2.60e-1$	0.77	27735
2^{nd}-order	$7.2e-2$	$1.71e-2$	–	2708	$1.75e-2$	–	3373	$9.21e-2$	–	3380
	$5.4e-2$	$1.01e-2$	1.83	10016	$1.01e-2$	1.92	10847	$5.22e-2$	1.95	11434
	$3.6e-2$	$4.72e-3$	1.9	60260	$4.75e-3$	1.88	60433	$2.43e-2$	1.91	60359
	$2.9e-2$	$3.06e-3$	1.94	127005	$3.11e-3$	1.9	128402	$1.60e-2$	1.89	128440

Table 2: Isentropic vortex – L^2 norm errors on density ρ between the numerical solution and the exact solution of the **stationary** isentropic vortex problem at $t_{\text{final}} = 1$ on **a mesh with only tetrahedra**.

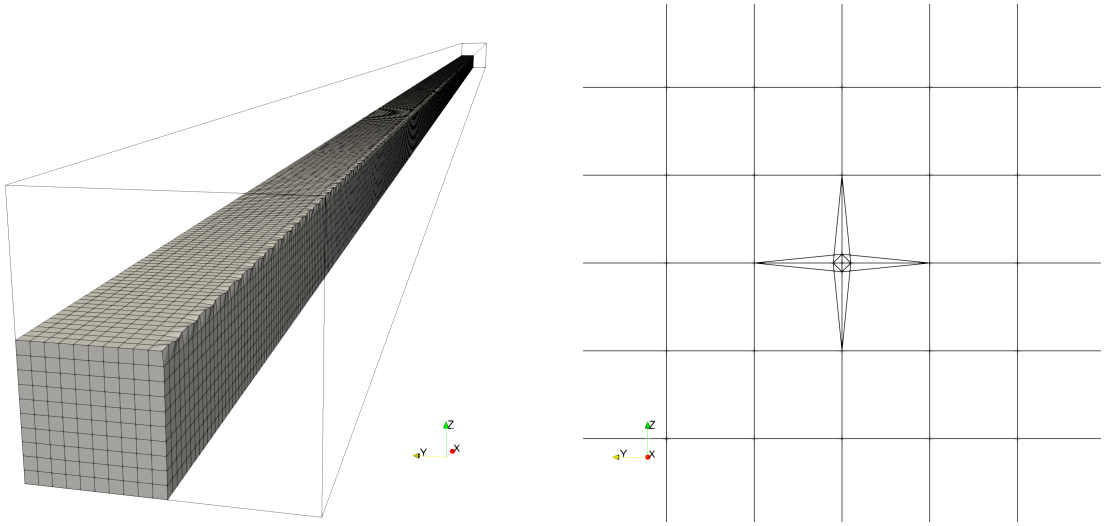
8.2. 3D extension of the odd-even decoupling test

Here, we propose the three-dimensional extension of the famous odd-even decoupling test case initially proposed by Quirk [46] to assess the sensitivity of Godunov-type methods to infinitesimal perturbations.

More precisely, this test case consists in computing the propagation of a planar shock wave on a pointwisely perturbed Cartesian grid for which only the center line is slightly perturbed. The computational domain is $\Omega = \{(x, y, z) \in [0, 800] \times [-10, 10] \times [-10, 10]\}$. The mesh is a regular hexaedral grid made of 800 cells in the x direction, and 20 cells in the y and z directions. A grid perturbation is introduced at the center line ($y = 0$ and $z = 0$) displacing the nodes position of the center line as follows

$$\tilde{\mathbf{x}}_p = \mathbf{x}_p + \begin{pmatrix} 0 \\ a_0 cos(\phi) \\ a_0 sin(\phi) \end{pmatrix},$$

where a_0 is the amplitude of the pertubation and ϕ the angle defined by $\phi = (\mathbf{x}_p \cdot \mathbf{e}_x)\frac{\pi}{2}$. Remark that this perturbation is coiling the center line of the mesh around the center axis but there is no movement in the x direction. Figure 6a shows the bottom left quadrant of the grid with a perturbation amplitude of $a_0 = 0.1$. Figure 6b shows the same grid when looking from the left side towards $+\mathbf{e}_x$. Remark that the angle ϕ has been chosen to alternatively move the nodes in the y and z direction.



(a) View of the bottom left quadrant of the grid.

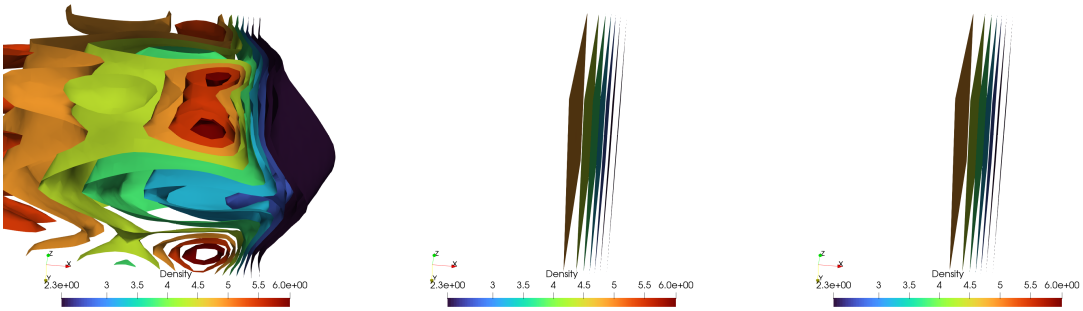
(b) View of the grid when looking towards $+\mathbf{e}_x$.

Figure 6: Grid fragments for a perturbation of amplitude $a_0 = 0.1$.

A right-going shock wave of Mach 6 is traveling through the domain initially filled with a diatomic gas in the state $(\rho^0, \mathbf{v}, p^0, \gamma) = (1, \mathbf{0}, 1, \frac{7}{5})$. The Rankine-Hugoniot relations provides the inflow state

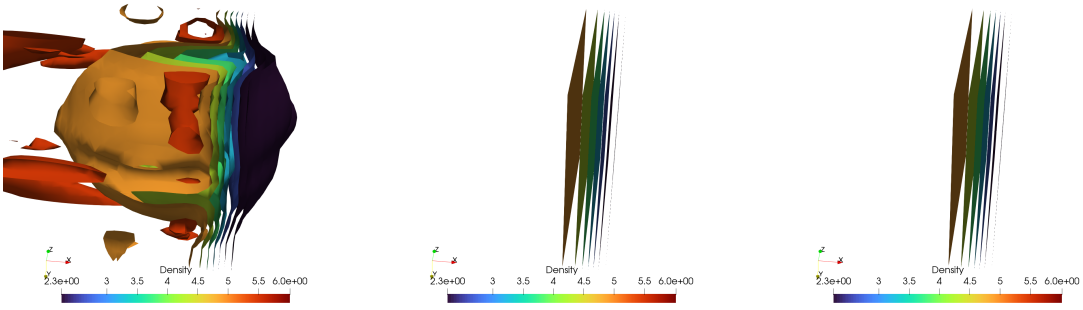
$$u_{\text{shock}} = \text{Ma}\sqrt{\gamma}, \quad \rho_\infty = \frac{(\gamma+1)\text{Ma}^2}{(\gamma-1)\text{Ma}^2+2}, \quad u_\infty = u_{\text{shock}}\frac{2(\text{Ma}^2-1)}{(\gamma+1)\text{Ma}^2}, \quad p_\infty = \frac{2\gamma\text{Ma}^2-(\gamma-1)}{(\gamma+1)},$$

where Ma denotes the Mach number. The left boundary condition is an inflow boundary condition characterized by the foregoing inflow state, wall boundary conditions are imposed on the top and bottom boundaries, while an outflow boundary condition has been prescribe on the right boundary. The final time of the computation is $t_{\text{final}} = 50$. The exact solution to this problem corresponds to a one-dimensional Mach 6 shock wave propagating at speed u_{shock} in the x-direction. Let us point out that since the imposed condition at the left side of the domain is uniform in the yz plane, **no matter** the amplitude of the perturbation of the mesh, **the solution should remain uniform** in the yz plane.



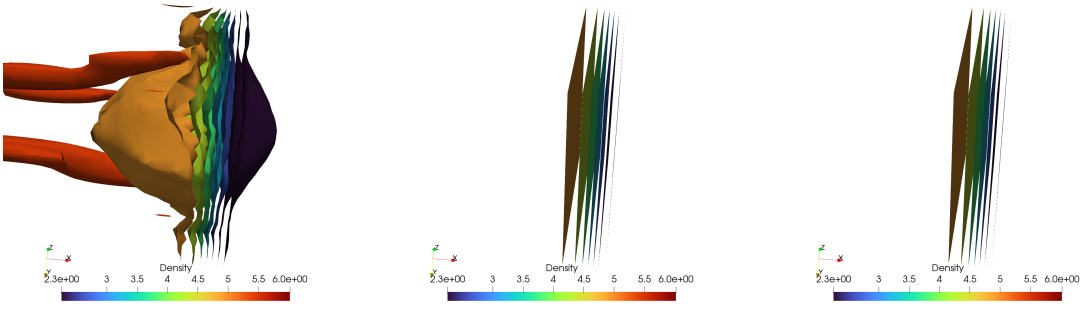
(a) Two-point scheme. (b) Modified two-point scheme. (c) Multi-point scheme.

Figure 7: Density isosurfaces viewed from the side at time $t_{final} = 50$ with the pertubation amplitude $a_0 = 10^{-3}$.



(a) Two-point scheme. (b) Modified two-point scheme. (c) Multi-point scheme.

Figure 8: Density isosurfaces viewed from the side at time $t_{final} = 50$ with a pertubation amplitude $a_0 = 10^{-6}$.



(a) Two-point scheme. (b) Modified two-point scheme. (c) Multi-point scheme.

Figure 9: Density isosurfaces viewed from the side at time $t_{final} = 50$ with a pertubation amplitude $a_0 = 10^{-9}$.

We have run the 3D odd-even test case using the two-point, modified two-point and multi-point schemes with the 3 grids respectively characterized by the pertubation amplitudes $a_0 = 10^{-3}$, $a_0 = 10^{-6}$ and $a_0 = 10^{-9}$. The corresponding numerical results are displayed in Figures 7, 8 and 9 wherein one cab observe the density contours plotted at the final time $t_{final} = 50$. Whatever the level of amplitude of the

initial infinitesimal pertubation, the classical two-point scheme exhibits the development of a significative numerical pertubation and is not able to compute correctly the analytical solution. One the other hand, both the modified two-point and multi-point schemes are able to capture the solution in a satisfactory manner and seem thus insensitive to the numerical perturbations. Following Rodionov [48], we infer that the good behavior of the modified two-point and multi-point schemes might come from their inherent multi-dimensional numerical dissipation. The theoretical investigation of this conjecture remains for the moment an open problem.

8.3. Supersonic jet

This test case, taken from [30], corresponds to a Riemann problem for the steady two-dimensional Euler equations. The interested reader might refer to [39] for the construction of the analytical solution to this type of Riemann problem in the case of a perfect gas equation of state. The computational domain $(x, y) \in [0, 1]^2$ is paved using a 200×200 uniform grid. The initial configuration corresponds to the juxtaposition of a lower and an upper fluids separated by the interface $y = 0.5$. Their respective states are expressed in terms of the primitive variables

$$(\rho, u, v, p) = \begin{cases} (\rho, u, v, p)_b = (1, 2.4, 0, 0.5) \text{ if } y < 0.5, \\ (\rho, u, v, p)_t = (0.25, 4.0, 0, 0.25) \text{ if } y \geq 0.5. \end{cases}$$

Here, u and v are the x and y components of the velocity field. A supersonic inflow boundary condition is prescribed at the left boundary $x = 0$, whereas an outflow boundary condition is imposed on the remaining boundaries. This set up gives birth to an upper shock wave and a lower rarefaction wave, separated by an oblique slide line which is thus not aligned with the computational grid. This provides an interesting flow configuration to assess the ability of the numerical methods to capture sharply the slide lines, which is an essential property for accurately resolving viscous flows such as boundary layers. We have made a comparison between the first-order versions of the two-point, two-point modified and multi-point Finite Volume methods to assess the performance of the underlying Riemann solver in resolving the shear line. The density contours are displayed in Figure 10. We have plotted the density, the x and y velocity components with respect to y along the line $x = 0.5$ in Figure 11. We take note that the two-point method exhibits the best results with a sharp description of the shear line. The two-point modified and the multi-point methods perform in a similar manner with a shear line which is a little bit smeared compared to the two-point method. However, the results obtained with the two-point modified and multi-point method are far better than the ones got with the classical HLLE, refer to [30].

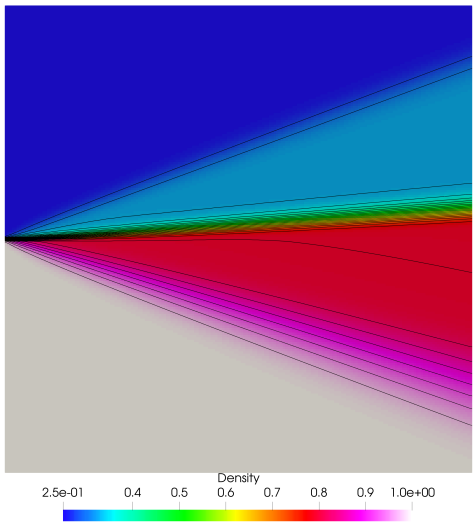
8.4. The Sedov test case

The famous Sedov problem [50] consists of a point-blast in a uniform cold medium with spherical symmetry. An exact solution based on self-similarity arguments is available, see for instance [34]. The initial conditions are characterized by $(\rho_0, \mathbf{v}_0, p_0) = (1, \mathbf{0}, 10^{-6})$ and the polytropic index is set equal to $\frac{7}{5}$. We set an initial delta-function energy source at the origin prescribing the pressure in the cell containing the origin as follows

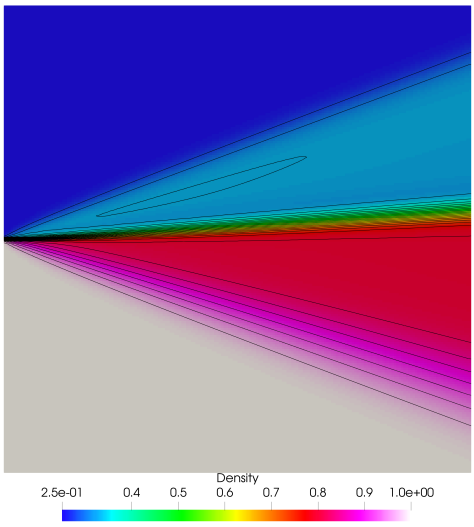
$$P_{\text{or}} = (\gamma - 1)\rho_{\text{or}} \frac{\mathcal{E}_0}{V_{\text{or}}},$$

where V_{or} denotes the volume of the cell and $\mathcal{E}_0$ is the total amount of released energy. Choosing $\mathcal{E}_0 = 0.851072$, as it is suggested in [34], the solution consists of a spherical diverging shock whose front is located at radius $R = 1$ at time $t = 1$. The peak density reaches the value 6.

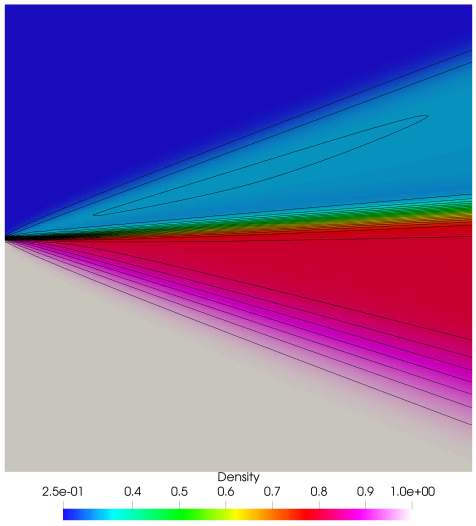
The computational domain is defined by the cubic region $(x, y, z) \in [-1.2, 1.2]^3$. Symmetry boundary conditions are prescribed on all boundaries. This test case has been run successively with a uniform Cartesian grid made of 64^3=262 144 hexaedras and an unstructured grid made of 287491 tetrahedras. The numerical simulations have been performer employing the two-point, the modified tow-point and the multi-point methods using both the first-order and the second-order space-time discretizations. The scattered plots



(a) Two-point scheme.

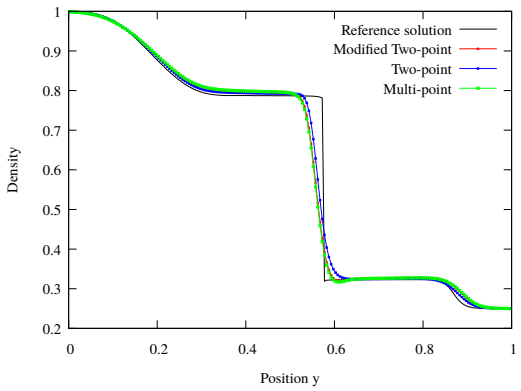


(b) Modified Two-point scheme.

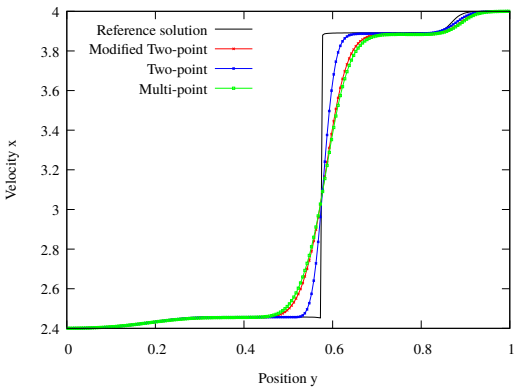


(c) Multi-point scheme.

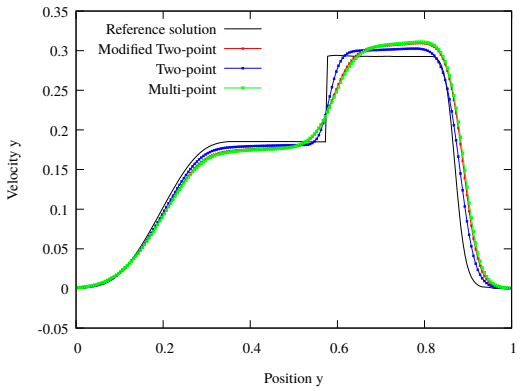
Figure 10: Comparison of the density contours.



(a) Density.

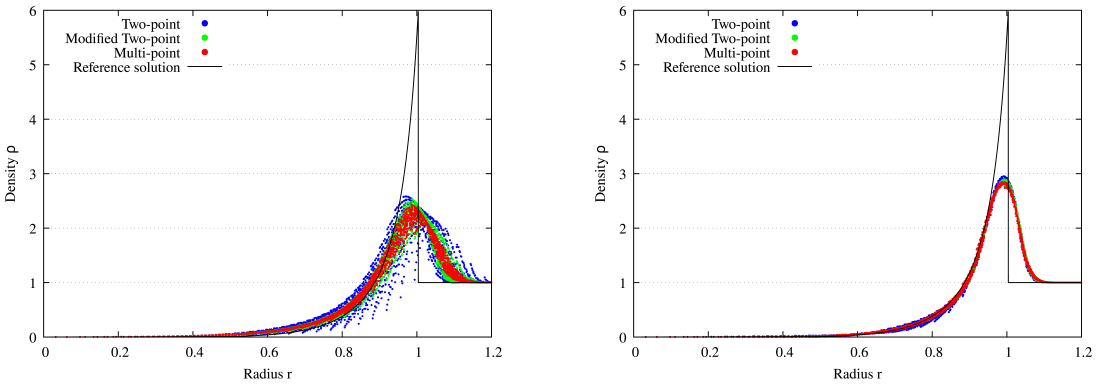


(b) Velocity u.



(c) Velocity v.

Figure 11: Comparison of the solution along $x = 1$.



(a) First-order.

(b) Second-order.

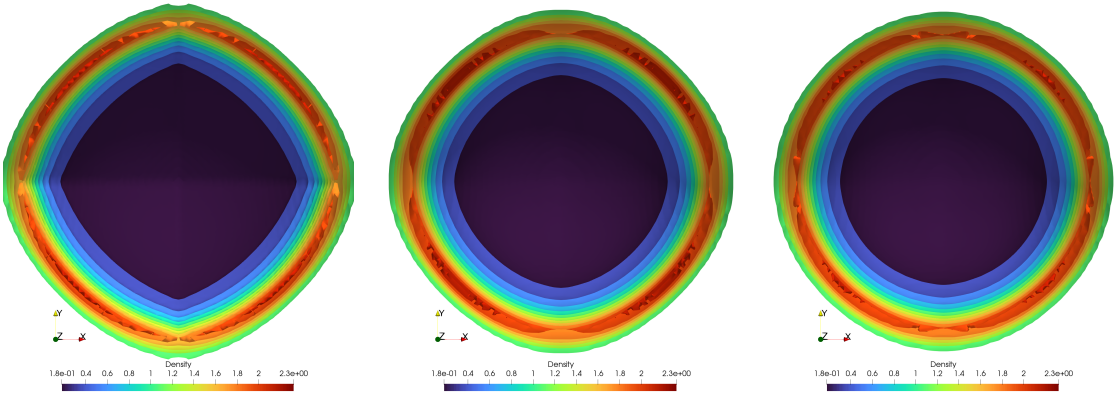
Figure 12: Sedov test case on the Cartesian grid made of 64^3 hexahedra. Scattered plots of the density in all the cells with respect to the radius of the cell at time $t = 1$. First and second-order space-time discretizations of the two-point, modified two-point and the multi-point Finite Volume methods.

of the density in all cells with respect to the cell radius have been displayed respectively in Figure 12 for the Cartesian grid and in Figure 14 for the unstructured grid. Figure 12a (resp. Figure 12b) shows the ability of the multi-point to better preserve the spherical symmetry of the numerical solution even on a non-aligned grid at first order (resp. second order). This observation is also confirmed by Figure 13 which shows that the two-point schemes badly preserve the spherical shape of the shock wave. In addition, we observe that the two-point and modified two-point methods are a little less dissipative than the multi-point method for the unstructured grid, refer to Figure 14. Let us point out that the Cartesian and the unstructured grids employed for this test case are rather coarse for an Eulerian computation, since there are only 32 zones over the interval [0, 1.2]. This explains the discrepancy between the numerical results and the analytical solution with respect to the peak value of the mass density at the shock wave.

8.5. Blunt-body test case

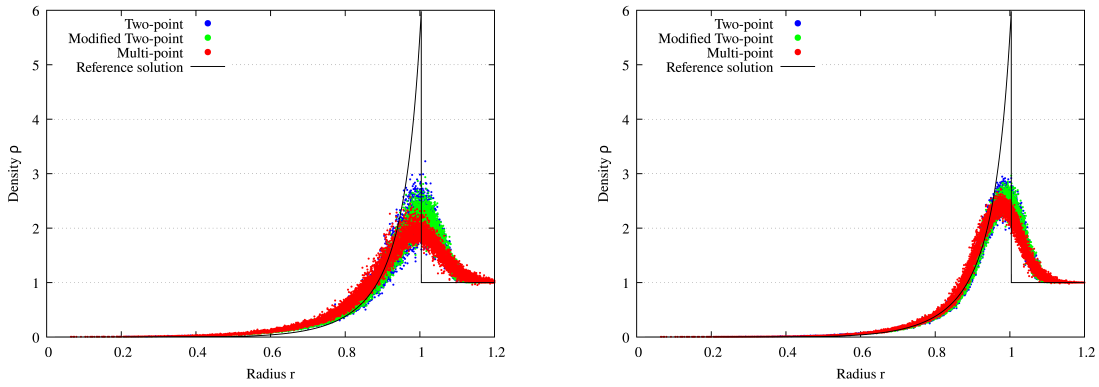
This test case is taken from the works of Candler *et al.* [10], wherein they assess the capability of their numerical method to cope with the numerical solution of a Mach 12 flow over a cylinder normal to the flow. Due to the high value of the Mach number, the structure of the steady solution consists of a bow shock that envelops the cylinder. The shock layer, *i.e.*, the portion of the flow located between the shock and the surface, is the location of complex physical phenomena: vorticity and strong entropy gradient both induced by the curved shock wave. As underlined by [11] the stagnation streamline is singular, with weak gradients in the streamwise direction and a low flow speed. In addition, the shock layer consists of the juxtaposition of a subsonic zone, in the vicinity of the stagnation point, and supersonic one elsewhere. We recall, that the steady Euler equations are elliptic in the subsonic zones whereas hyperbolic in the supersonic ones. All these facts, render the computation of such flows particular challenging for numerical methods. This problem has been run using 3 types of grids to assess the effect of the grid on the numerical solution:

- Two structured hexaedral grids: the first one has been generated with the constraint of being aligned with the bow shock as in [10], refer to Figure 15a, whereas the second one is a radial hexaedral grid *a priori* non aligned with the bow shock;
- One unstructured prismatic grid obtained by extruding the triangular paving the plane of symmetry of the cylinder, refer to Figure 15b;
- One unstructured tetrahedral grid, refer to Figure 15c.



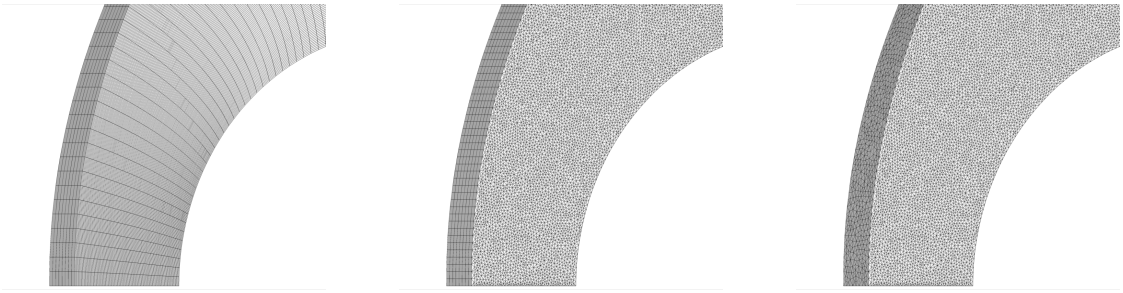
(a) Two-point scheme. (b) Modified Two-point scheme. (c) Multi-point scheme.

Figure 13: Sedov test case on the Cartesian grid made of 64^3 hexaedras. Density contours at time $t = 1$: 15 equally spaced iso surfaces over [0, 2.5]. View of the domain $z < 0$. First -order space-time discretizations of the two-point, modified two-point and the multi-point Finite Volume methods.



(a) First-order. (b) Second-order.

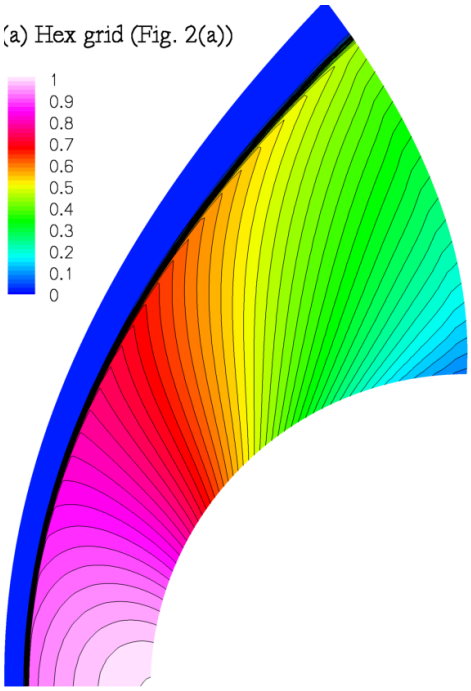
Figure 14: Sedov test case on an unstructured grid made of 287 491 tetrahedras. Scattered plots of the density in all the cells with respect to the radius of the cell at time $t = 1$. First and second-order space-time discretizations of the two-point, modified two-point and the multi-point Finite Volume methods.



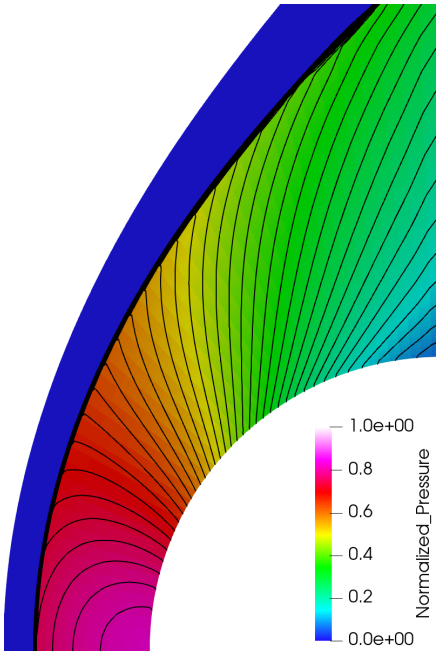
(a) Hexaedral grid. (b) Prismatic grid. (c) Tetrahedral grid.

Figure 15: Blunt-body test case: The different types of grids used after [10].

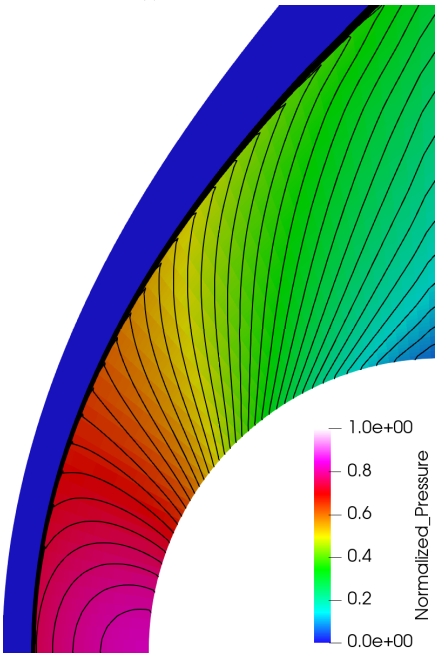
The size of the grids we have employed is similar to what has been prescribed in [10]. More precisely, for the hexaedral grid aligned with the bow shock there are 44 equally-spaced zones along the circumference of the quarter cylinder and 140 equally-space layers in the direction normal to the surface. The non aligned hexaedral grid is still made of 44 equally-spaced zones along the circumference of the quarter cylinder and 320 equally-spaced layers in the normal direction to the surface. In this case, the computational domain is $(r, \theta, z) \in [0.5, 1.25] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 0.00625]$. For the prismatic grid, we use the same computational domain and fill it with 75000 prisms extruded from the plane of symmetry. For the tetrahedral grid, we generate a 3D unstructured grid of the computational domain $[(r, \theta, z) \in [0.5, 1.25] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \times [0, 0.05]$ with approximately 500000 elements. Compared to [10] we have run one supplementary computation in order to assess the impact of a non aligned hexaedral grid with the bow shock. Therefore, we have 4 sets of computations performed respectively on an aligned hexaedral grid, refer to Figures 16 and 17, a non aligned hexaedral grid, refer to Figures 18 and 19, a prismatic grid, refer to Figures 20 and 21 and a tetrahedral grid, refer to Figures 22 and 23. For each set, we have computed the steady state numerical solution utilizing the second-order space and time discretization version of the two-point, two-point modified and multi-point methods. Following [10], for each computation, we have post-processed the normalized pressure and temperature contours. For each physical quantity, the normalization has been done with respect to the corresponding theoretical value at the stagnation point, *i.e.*, the stagnation pressure has been evaluated with the Rayleigh-Pitot formula, refer to [4]. For each set of images, for the sake of comparison, we have also displayed the counterpart image taken from [10], when it was available. For the hexaedral grid aligned with the bow shock, we have obtained satisfying results which are in quite good agreement with those taken from [10], refer to Figure 16. Regarding the temperature iso-surfaces, the multi-point method exhibit quite good results, the two-point method produces a result characterized by the occurrence of pertubations in the vicinity of the stagnation line to Figure 17, which might be the premises of a carbuncle instability. The use of a non aligned grid with the bow shock has an impact on the quality of results, particularly on the temperature iso-surfaces, refer to Figure 19. With the prismatic grid, the pressure iso-surfaces obtained using the multi-point method seem to be the less perturbed, refer to Figure 20d. We observe the occurrence of the carbuncle instability for the two-point method which renders the computation completely useless, refer to Figure 20b. We take note that all the results regarding the temperature iso-surfaces are of low quality since the iso-surfaces of temperature are strongly perturbed, refer to Figure 21. We observe the occurrence of several peaks of temperature inside the shock layer, particularly for the US3D and two-point modified results, refer to 21a and 21c. Finally, with the tetrahedral grid, we have obtained numerical results which are quite similar to those obtained with the prismatic grid, namely of poor quality! We observe that the multi-point scheme provides a quite acceptable result in terms of pressure iso-surfaces, refer to Figure 22d. However, this is not the case anymore for the temperature iso-surfaces since all the methods produce quite non physical results. Once more the two-point method failed due to the occurrence of the carbuncle instability, refer to Figure 23b.



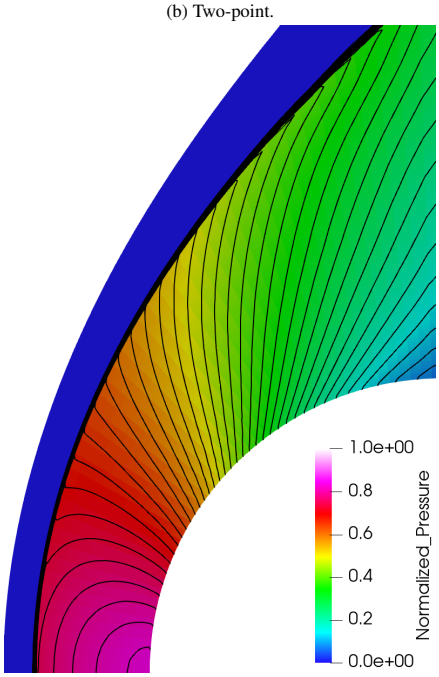
(a) US3D code from [10].



(b) Two-point.

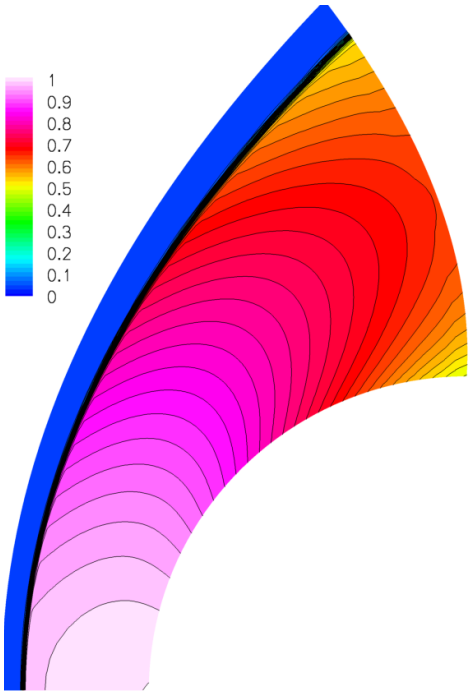


(c) Modified two-point.

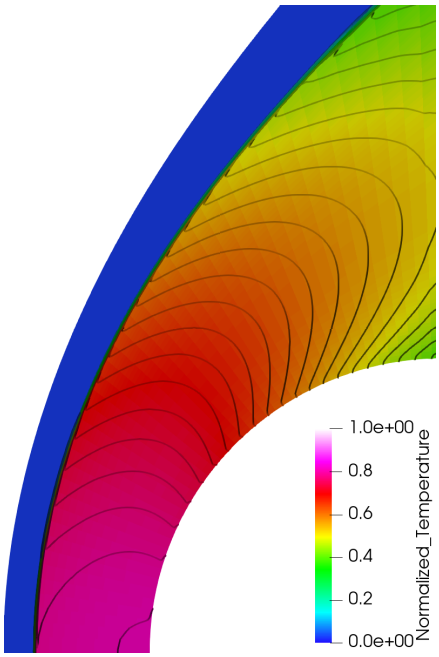


(d) multi-point.

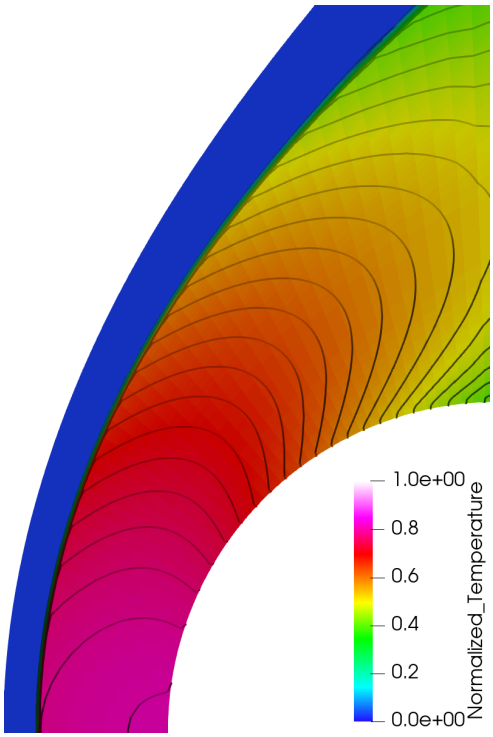
Figure 16: Blunt-body problem using a **hexahedral grid aligned** with the bow shock: Pressure contours.



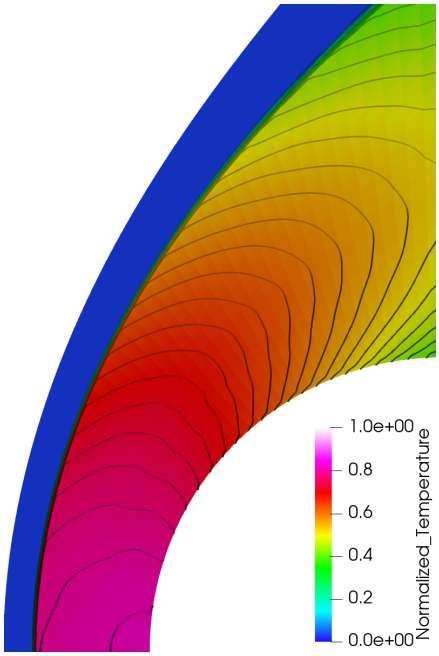
(a) US3D code from [10].



(b) Two-point.

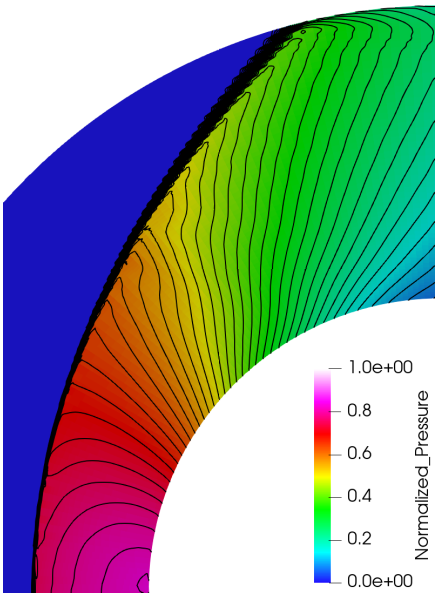


(c) Modified two-point.

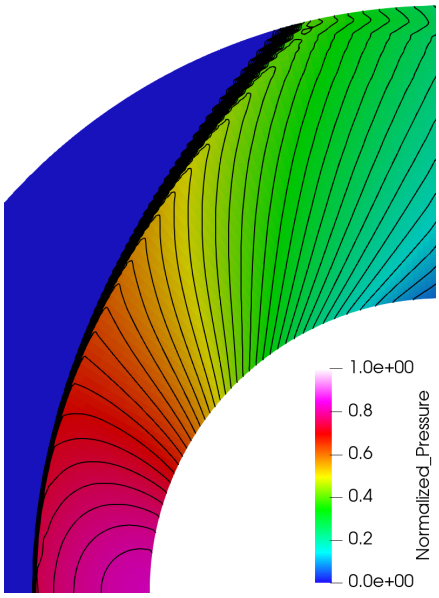


(d) Multi-point.

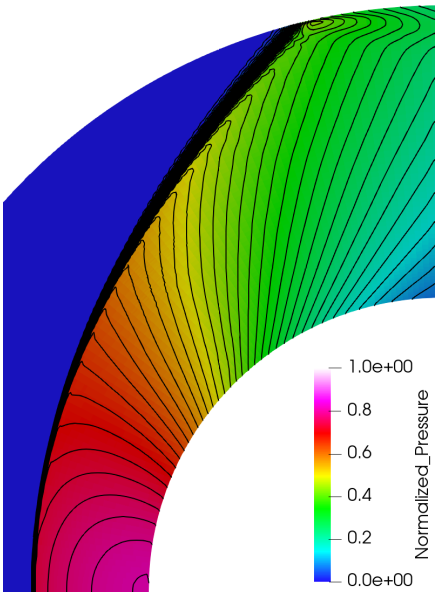
Figure 17: Blunt-body problem using a **hexahedral grid aligned** with the bow shock: Temperature contours.



(a) Two-point.

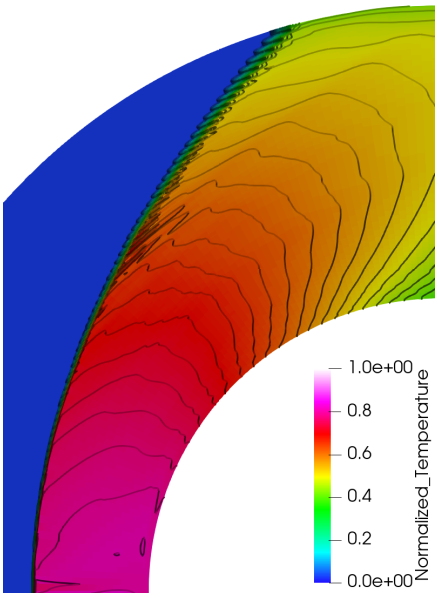


(b) Modified two-point.

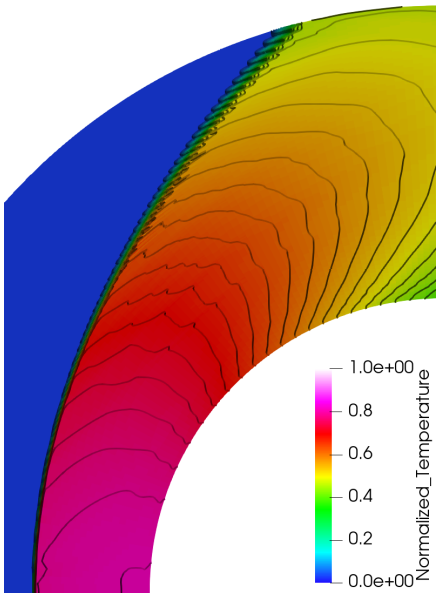


(c) Multi-point.

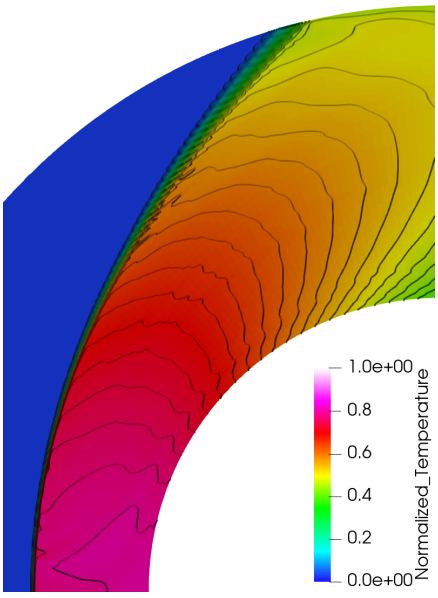
Figure 18: Blunt-body problem using a **hexahedral grid non aligned** with the bow shock: Pressure contours.



(a) Two-point.

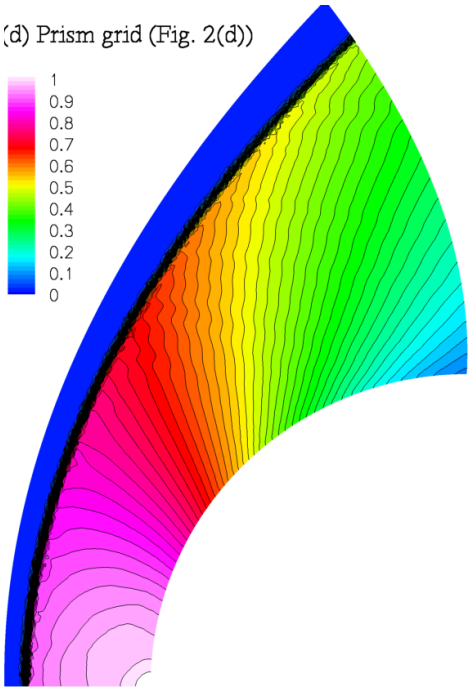


(b) Modified two-point.

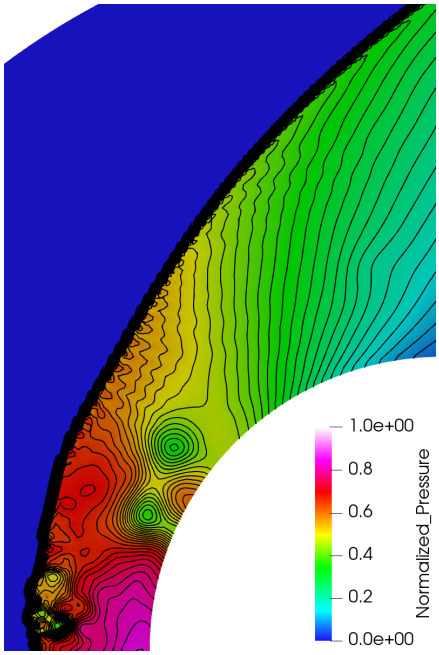


(c) Multi-point.

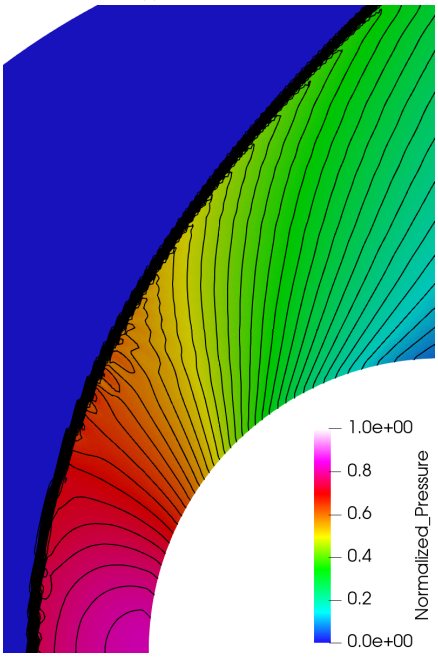
Figure 19: Blunt-body problem using a **hexahedral grid non aligned** with the bow shock: Temperature contours.



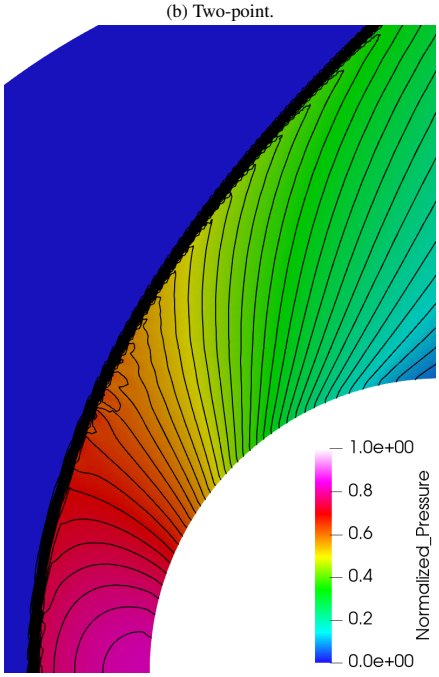
(a) US3D code from [10].



(b) Two-point.

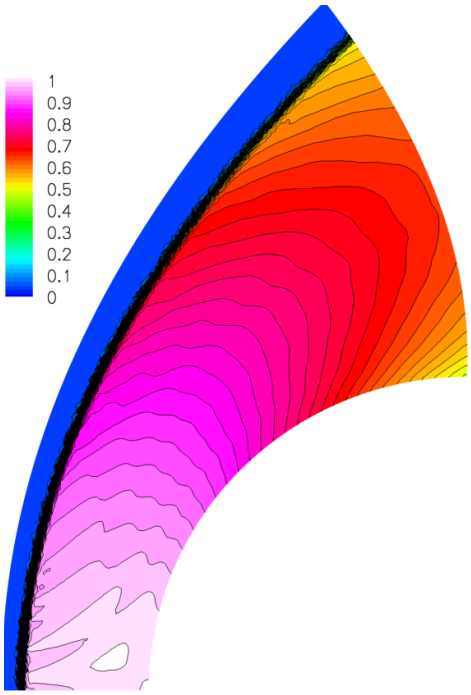


(c) Modified two-point.

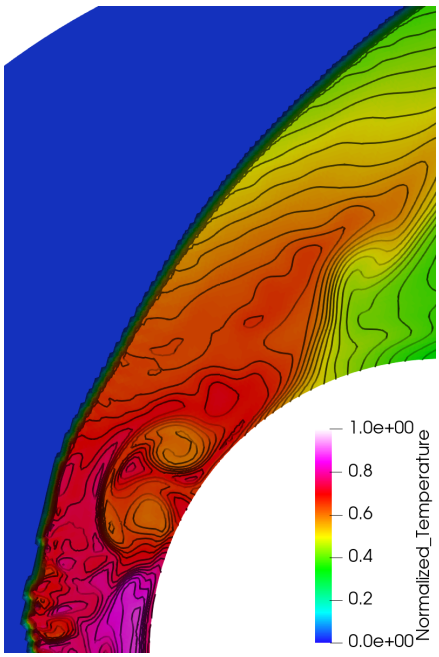


(d) Multi-point.

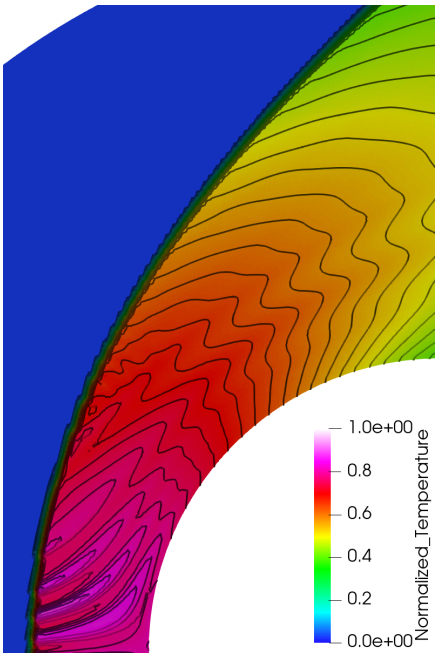
Figure 20: Blunt-body problem using a **prismatic grid**: Pressure contours.



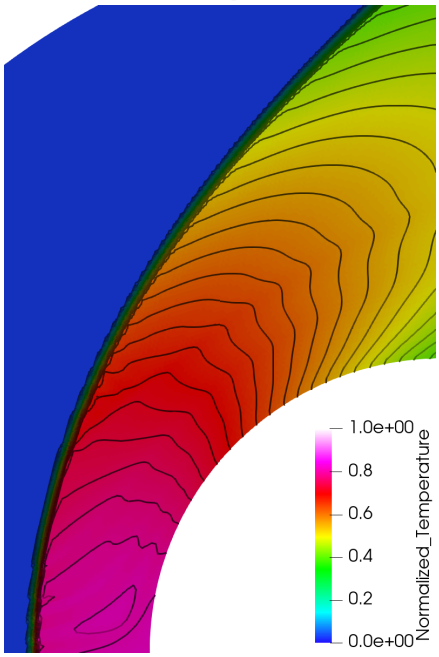
(a) US3D code from [10].



(b) Two-point.

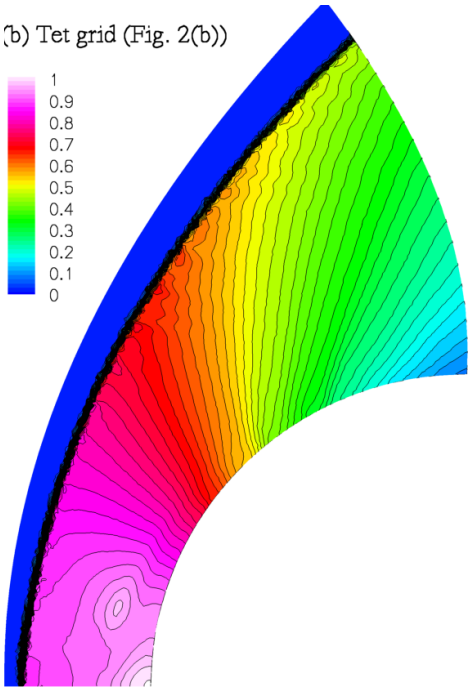


(c) Modified two-point.

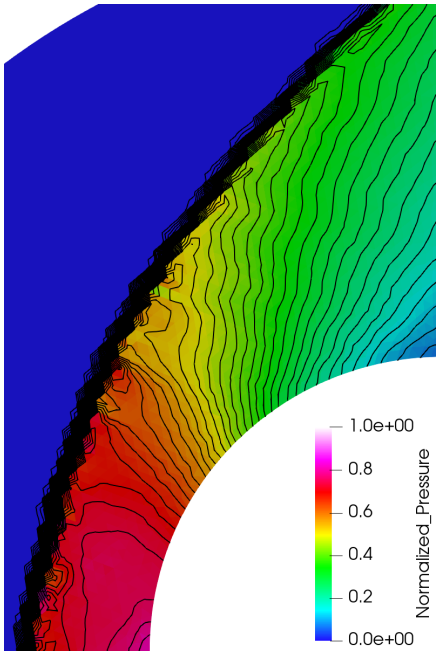


(d) Multi-point.

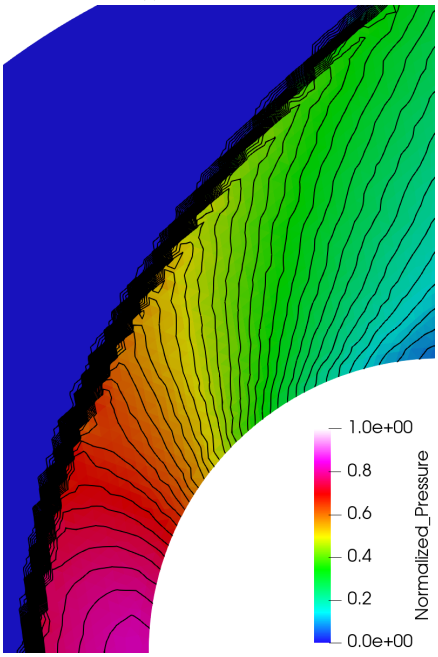
Figure 21: Blunt-body problem using a **prismatic grid**: Temperature contours.



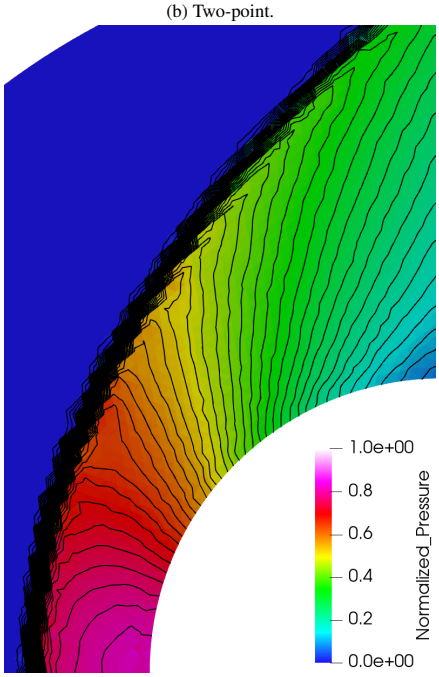
(a) US3D code from [10].



(b) Two-point.

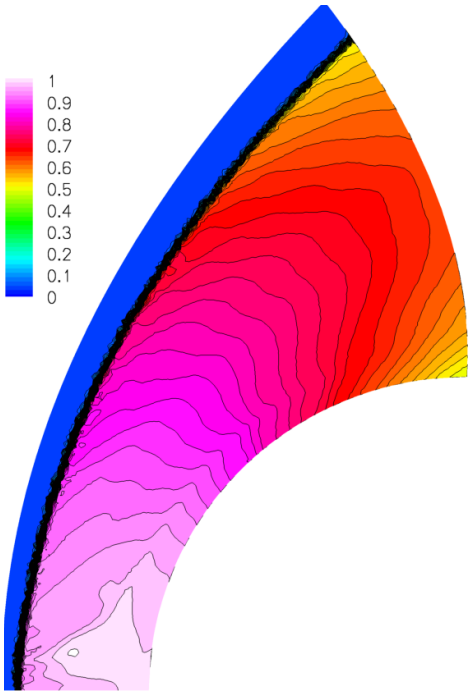


(c) Modified two-point.

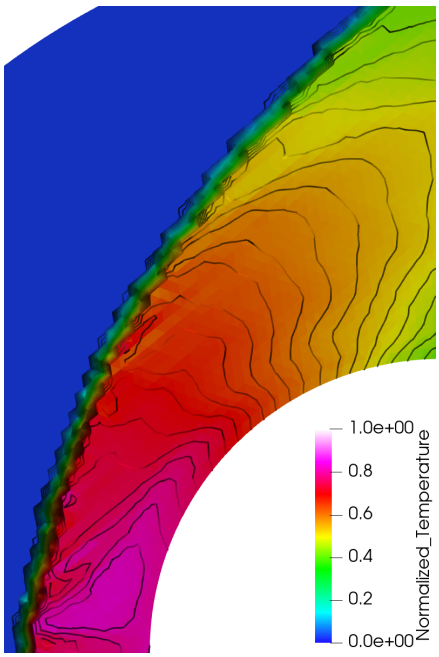


(d) Multi-point.

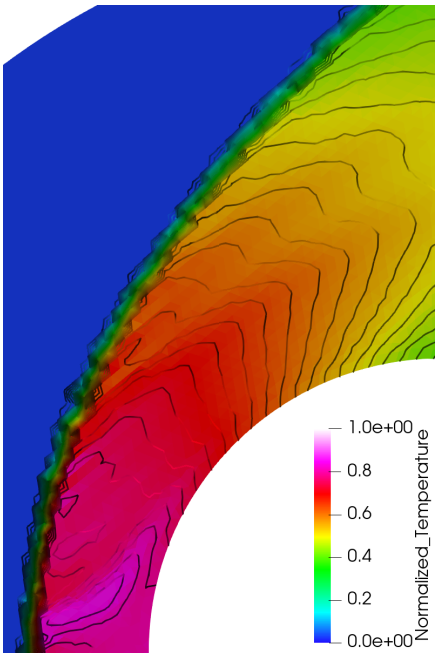
Figure 22: Blunt-body problem using a **tetrahedral grid**: Pressure contours.



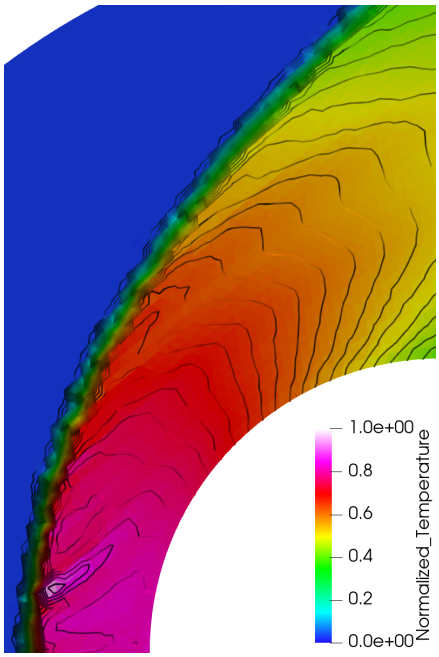
(a) US3D code from [10].



(b) Two-point.



(c) Modified two-point.

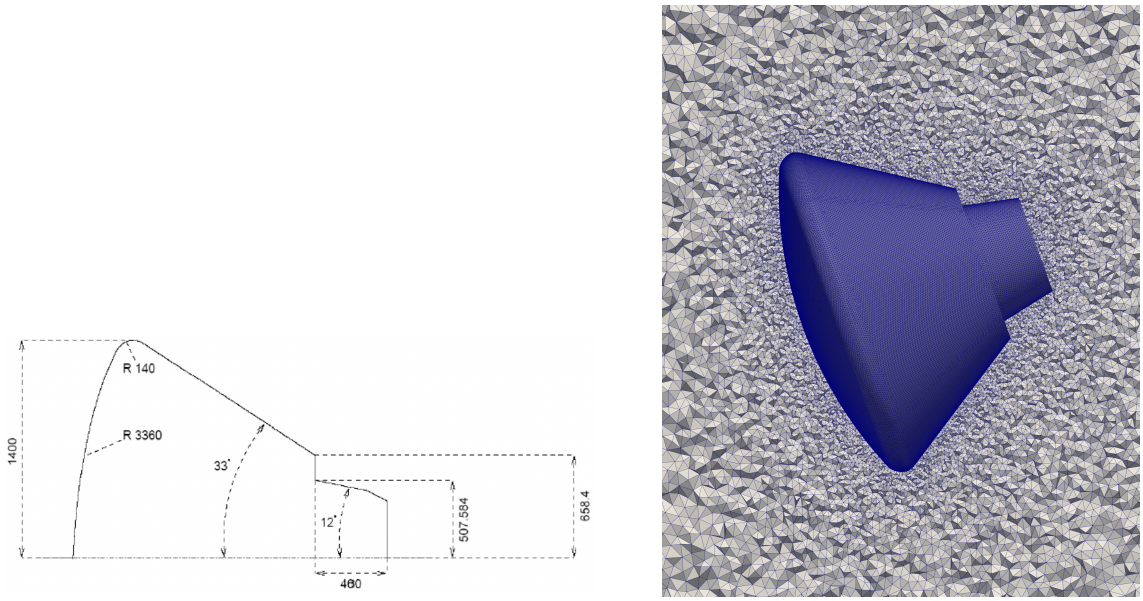


(d) Multi-point.

Figure 23: Blunt-body problem using a **tetrahedral grid**: Temperature contours.

Quantities	ARD	Pre-X
Mach	24	25
Altitude (km)	65.83	73.6
Velocity (ms^{-1})	7212.43	7205
Density (kgm^{-3})	$1.5869\ 10^{-4}$	$5.546\ 10^{-5}$
Temperature (K)	224.5	207
Pressure (Pa)	10.23	3.11
Wall temperature (K)	1500	1500
Angle of attack (^{o})	20	40

Table 3: Freestream conditions used for the ARD and the PRE-X test cases and reproduced from [5].



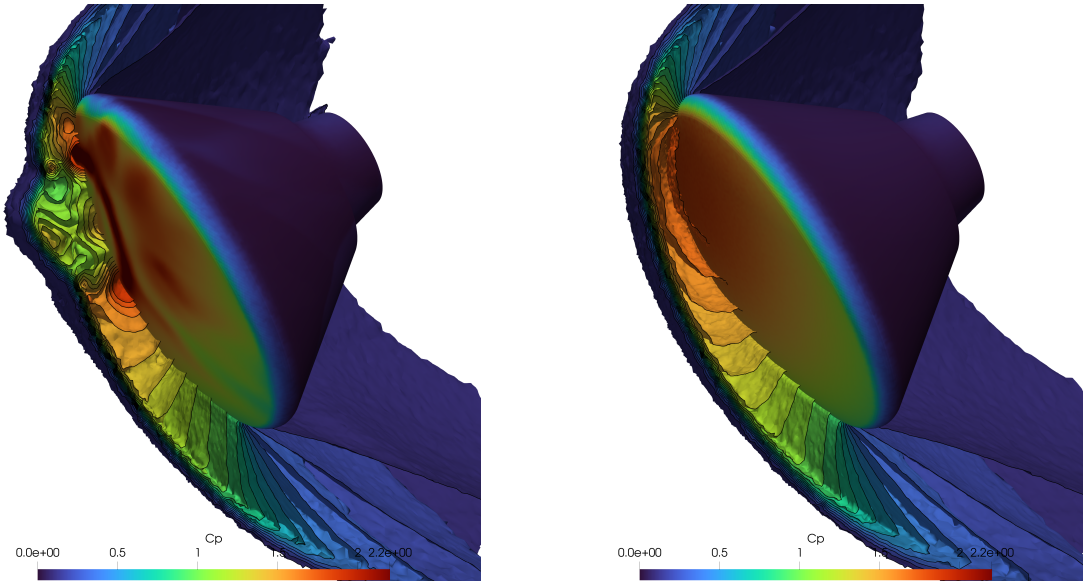
(a) Description of the geometry (lengths are given in millimeter).

(b) View of the computational grid.

Figure 24: ARD: Geometry and Grid.

8.6. Atmospheric Reentry Demonstrator test case

This test case is taken from [5] and we warmly thank J. Annaloro and S. Galera from CNES (French National Space Agency) for having communicated us the underlying geometrical data. The ARD (flight in 1998) is characterized by an axisymmetric Appolo-like shape. It consists of the assembly of a spherical capsule, a tore and a conical surface. The main geometrical characteristics are displayed in Figure 24a. The freestream conditions used for the computation are given in table 3. The numerical simulations are performed employing the first-order in space and time version of the two-point and the multi-point method. The computational grid, composed of 4.8 million tetrahedra, has been straightforwardly generated utilizing the Gmsh mesh generator [23], refer to Figure 24b. We have displayed in Figure 25 a three-dimensional representation of the flow around the ARD colored in terms of pressure iso-surfaces. More precisely, Figure 25a corresponds to the numerical result obtained using the two-point method, whereas Figure 25b corresponds to the numerical results obtained using the multi-point method. We observe that the two-point method has developed a carbuncle instability which renders the result useless. Finally, we produce a



(a) Two-point.

(b) Multi-point.

Figure 25: ARD: Pressure iso-surfaces.

more quantitative result computing the pressure coefficient, *i.e.*, $C_p = \dfrac{P_w - P_\infty}{\frac{1}{2}\rho_\infty \mathbf{v}_\infty^2}$. In Figure 26 we compare the C_p computed using the multi-point method to the values of C_p obtained by means of the MISTRAL code, which is a code solving the Navier-Stokes equations and developed by the R.Tech company (`https://fr.linkedin.com/company/r.tech`), and also to the values of C_p resulting from flight measurements. All these data are taken from [5]. We observe a quite good agreement with the experimental data.

8.7. PRE-X test case

This test case is also taken from [5]. The main objective of the PRE-X vehicle project was to obtain flight data to enhance the knowledge of the thermal and aerodynamic loads in the perspective of the future European reusable vehicle. This project has been stopped in 2007 and replaced by ESA IXV Program. The geometry of the PRE-X has been displayed in Figure 27a, it is characterized by a length of 4 m, a reference surface of 6 m^2 and a nose radius of 0.925 m. The presence of flaps implies a genuinely three-dimensional geometry whose meshing might be particularly cumbersome when utilizing multi-block structured grids. In the present case, the grid has been straightforwardly generated employing the Gmsh mesh generator. It is an unstructured grid composed of 6 million of tetrahedra, refer to Figure 27b. The freestream conditions used for the computation are given in table 3. The numerical simulation at Mach 25 has been run using the first-order in space and time version of the multi-point method. The trace of the planes $y = 0$ and $y = 0.3$ on the surface of the vehicle has been plotted in Figure 28a, whereas Figure 28b shows the pressure iso-surfaces in the flow surrounding the vehicle and the pressure coefficient at the surface. We compute the pressure coefficient, *i.e.*, $C_p = \dfrac{P_w - P_\infty}{\frac{1}{2}\rho_\infty \mathbf{v}_\infty^2}$ along the trace of the planes $y = 0$ and $y = 0.3$ on the surface of the vehicle. In Figure 29 we compare the C_p computed using the multi-point method to the values of C_p obtained by means of the Navier-Stokes MISTRAL code. As for the ARD test case, all these data are taken from [5]. Once more, we take note of a quite good agreement between the numerical solutions except in the vicinity of the flaps wherein complex physical phenomena might occur such as flow recirculation

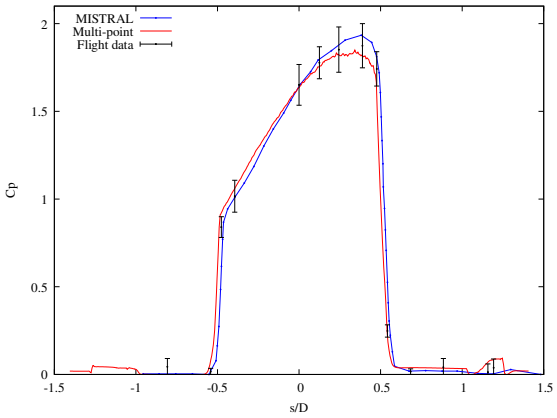
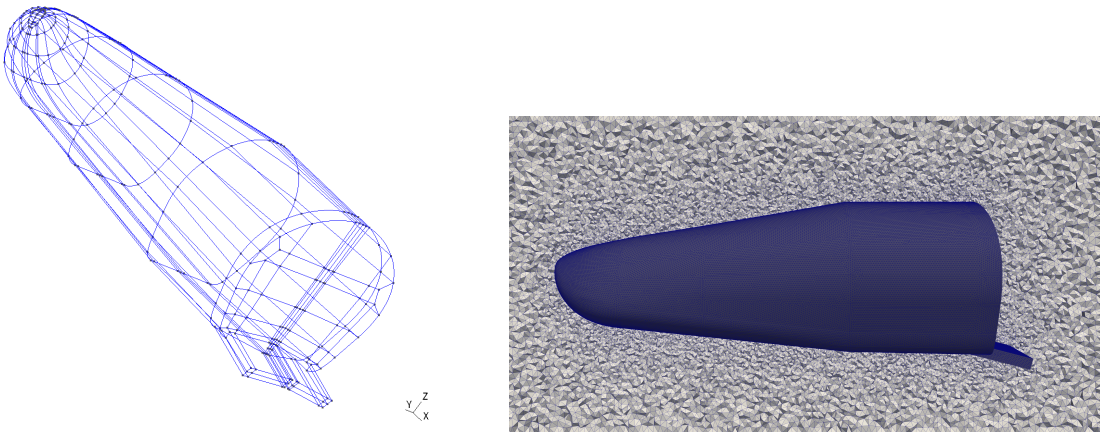


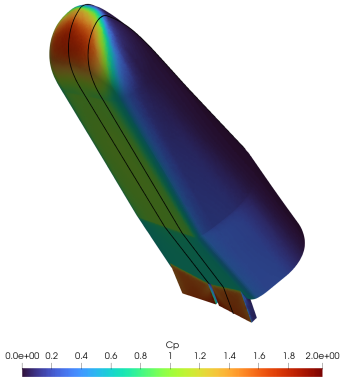
Figure 26: ARD: Comparison of pressure coefficients obtained by numerical simulations and from flight data along the trace of the symmetry plane $y = 0$ on the surface.



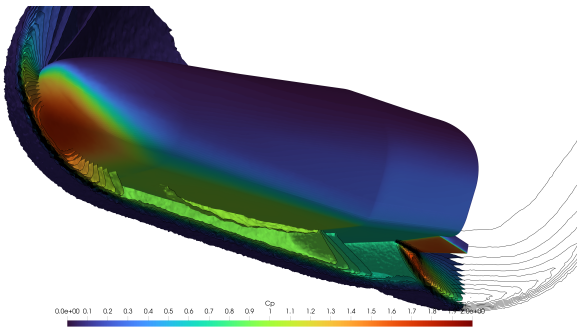
(a) Geometry.

(b) Grid and trace of the grid on the surface of the vehicle.

Figure 27: PREX: Geometry and grid.

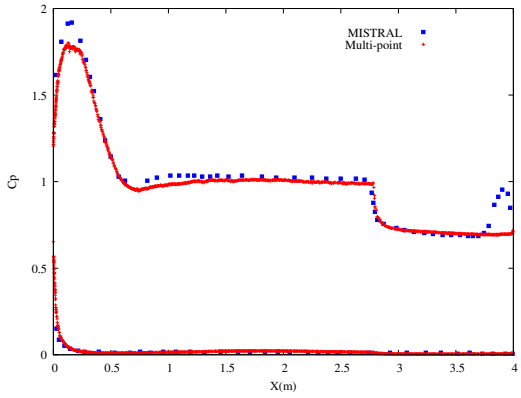


(a) Traces of the planes $y = 0$ and $y = 0.3$ m on the surface.

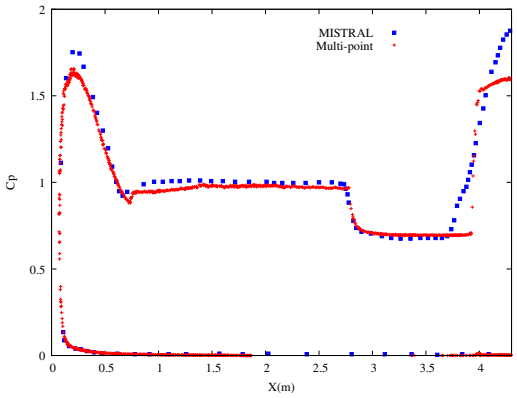


(b) Pressure iso-surfaces computed with the multi-point method.

Figure 28: PREX: Representations of the pressure field.



(a) Pressure coefficient along the trace of the plane $y = 0$ on the surface.



(b) Pressure coefficient along the trace of the plane $y = 0.3$ on the surface.

Figure 29: PREX: Comparison of pressure coefficients obtained by numerical simulations.

and viscous interactions, which cannot be captured by the inviscid model underlying the multi-point Finite Volume method.

9. Conclusion and Perspectives

We have derived an original subface-based Finite Volume method for solving compressible Euler equations on general multidimensional unstructured grids. The underlying subface flux approximation stems from an approximate Riemann solver which does not fulfill the classical HLL consistency condition. This approximate Riemann solver has been designed through the use of a particular decomposition of the flux into an advective plus a pressure/Lagrangian part. It is characterized by a nodal parameter and mass flux parameters which are nothing but the underlying Lagrangian wave speeds. The mass flux parameters are tuned to ensure the positivity and the thermodynamic consistency of the intermediate states. On the other hand, the non satisfaction of the HLL condition implies that the resulting Finite Volume method is not conservative in the classical face-based sense since the subface flux approximation is not unique anymore. The

restoration of conservation is obtained through the introduction of a local node-based conservation condition, which determines the nodal parameter as the nodal velocity and achieves the subface flux definition. Finally, we have obtained a multipoint Finite Volume method characterized by a node-based stencil. It is a positivity preserving and entropy stable numerical method provided an explicit condition of CFL type is satisfied by the time step. This numerical method is naturally able to cope with unstructured grids and we have assessed it against various demanding and representative tests of hypersonic flows. We have obtained promising results, which show its insensitivity to the numerical pathologies (shock instabilities) that usually plague the classical two-point Finite Volume method. Even if the modified two-point Finite Volume scheme proposed in this work better handles these numerical pathologies, still it suffers from such a lack of symmetry preservation and a more pronounced drawback of mesh imprinting. However, the results obtained are not perfect and the present numerical method still needs more investigations and developments to better understand its behavior and also to improve and to extend its capabilities.

As a summary, the proposed first-order accurate multi-point scheme in this work is the natural 3D extension of the multi-point scheme designed in [22]. Althought the extension seems very much alike compared to the 2D version, it nonetheless required some clarification, adapted notation and a rigorous derivation. The extension of the mulit-point scheme to a second-order of accuracy in this article proved to be promissing in terms of accuracy and efficiency thanks to an inexpensive gradient computation. At last, reusing the wave speeds for the approximate Riemann solver a new modified two-point scheme has been proposed. The quality of the obtained numerical results is improved without erasing some of its defects.

In the future, we plan to pursue the theoretical study of the multipoint flux approximation to have a better control of the numerical dissipation embedded in this formulation. Following the properties list proposed by Kitumura [37, 38], we intend to focus on: the shock stability, the conservation of total enthalpy and the ability to preserve slide lines. The first property is directly linked to the control of numerical instabilities attached to the simulation of strong shock waves (carbuncle, odd-even decoupling). The second one is a physical requirement since for steady flow the total enthalphy is conserved not only along stream line but also across shock wave by virtue of the Rankine-Hugoniot equations. The third one is related to the structure of the approximate Riemann solver knowing that we have to find an acceptable trade-off between robustness and stability. Namely, we want to ensure shock stability while resolving sharply the boundary layer, which is not a trivial task. We also aim at investigating the low Mach version of this formulation acknowledging that it could be useful to increase locally the accuracy of the method in the subsonic zones of the flow, the interested reader might refer to [26]. The control of the numerical dissipation shall be also improved by the construction of the entropy conservative [54, 55] and kinetic energy preserving [33] extensions of the multipoint flux. This shall be done employing the simple and elegant technique proposed by Abgrall [3, 2].

For now, the multipoint Finite Volume method has been essentially employed with an explicit time discretization. We need to investigate further its extension to implicit time discretization taking inspiration from [7, 18].

Finally, we need to address the Finite Volume discretizations of the viscous and heat conducting parts to achieve the discretization of the Navier-Stokes equations on general unstructured grids. This requires the use of Finite Volume discretization relying on a consistent numerical approximation of the diffusion fluxes. Since [36], it is well known that standard Finite Volume algorithms relying on a two-point face-based approximation of the diffusive flux behave poorly on skewed grids. Indeed, the classical face-based diffusion scheme run on a distorted quadrangular grid produces a numerical solution for which the temperature iso-surfaces are aligned with the grid! This is the consequence of consistency loss for the numerical method. That is why, the diffusive operators of the Navier-Stokes equations have to be discretized properly employing a flux approximation which remains consistent even on distorded grids. There exists a huge literature about this topic and the interested reader might consult [19] and the references therein. We have already investigated this question developing a multidimensional multipoint flux approximation [32] that is second-order accurate on tetrahedral grids. We have also used this technique to develop a two-dimensional unstructured Navier-Stokes Finite Volume discretization that produced promising results, refer

to [31]. Now, it remains to extend these works to general three-dimensional unstructured grids.

Acknowledgments

This work has been undertaken under the auspice of LRC ANABASE, which is a joined research laboratory between Institut de Mathématiques de Bordeaux and CEA-CESTA devoted to the development of innovative numerical methods for the simulation of complex fluid flows.

References

[1] R. Abgrall. Residual distribution schemes: Current status and future trends. *Comput. Fluids*, 35:641–669, 2006.

[2] R. Abgrall. A general framework to construct schemes satisfying additional conservation relations. Application to entropy conservative and entropy dissipative schemes. *J. Comput. Phys.*, 372:640–666, 2018.

[3] R. Abgrall. Some remarks about conservation for residual distribution schemes. *Computational Methods in Applied Mathematics*, 18(3):327–351, 2018.

[4] J.D. Anderson. *Hypersonic and High-Temperature Gas Dynamics-Second Edition*. AIAA education series, 2006.

[5] J. Annaloro, S. Galera, P. Kärräng, Y. Prevereaud, J.-L. Verant, M. Spel, and P. Van Hauwaert. Space debris atmospheric entry prediction with spacecraft-oriented tools. In *7th European Conference on Space Debris*, volume 7, https://conference.sdo.esoc.esa.int/proceedings/sdc7/paper/1062, 2017. ESA Space Debris Office.

[6] P. Batten, N. Clarke, C. Lambert, and D.M. Causon. On the choice of wavespeeds for the HLLC Riemann solver. *SIAM J. Sci. Comput.*, 18(6):1553–1570, 1997.

[7] P. Batten, M.A. Leschziner, and U.C. Goldberg. Average-State Jacobians and Implicit Methods for Compressible Viscous and Turbulent Flows. *J. Comput. Phys.*, 137:38–78, 1997.

[8] A. Beccantini, P. Galon, N. Lelong, and F. Baj. Tangential artificial viscosity to alleviate the carbuncle phenomenon, with applications to single-component and multi-material flows. *J. Comput. Phys.*, 516, 2024.

[9] F. Bouchut. *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws*. Birkhauser, first edition, 2004.

[10] G.V. Candler, M.D. Barnhardt, T.W. Drayna, I. Nompelis, D.M. Peterson, and P. Subbareddy. Unstructured Grid Approaches for Accurate Aeroheating Simulations. In *18th AIAA Computational Fluid Dynamics Conference, 25-28 June 2007, Miami, FL*, number AIAA 2007-3959, https://doi.org/10.2514/6.2007-3959, 2007.

[11] G.V. Candler, D. Mavriplis, and L. Trevi no. Current Status and Future Prospects for the Numerical Simulation of Hypersonic Flows. In *47th AIAA Aerospace Sciences Meeting 5-8 January 2009 Orlando, F*, https://doi.org/10.2514/6.2009-153, 2009.

[12] G.V. Candler, P.K. Subbareddy, and I. Nompelis. *Hypersonic Nonequilibrium Flows: Fundamentals and Recent Advances*, volume 247 of *Progress in Astronautics and Aeronautics*, chapter CFD Methods for Hypersonic Flows and Aerothermodynamics. American Institute of Aeronautics and Astronautics, Inc., 2015.

[13] E.J. Caramana, D.E. Burton, M.J. Shashkov, and P.P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146:227–262, 1998.

[14] F. Chalot and T.J.R Hughes. A consistent equilibrium chemistry algorithm for hypersonic flows. *Comp. Meth. Appl. Mech. Engrg.*, 112:25–40, 1994.

[15] F. Chalot, T.J.R Hughes, and F. Shakib. Symmetrization of conservation laws with entropy for high temperature hypersonic computations. *Computing Systems in Engineering*, 1:465–521, 1990.

[16] A. Chan, G. Gallice, R. Loubère, and P.-H. Maire. Positivity preserving and entropy consistent approximate Riemann solvers dedicated to the high-order MOOD-based Finite Volume discretization of Lagrangian and Eulerian gas dynamics. *Comput. Fluids*, 2021.

[17] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon. *Parallel programming in OpenMP*. Morgan Kaufmann, 2001.

[18] B. Cossart, J.-Ph. Braeunig, and R. Loubère. Toward robust linear implicit schemes for steady state hypersonic flows. *J. Comput. Phys.*, 522, 2024.

[19] J. Droniou. Finite volume schemes for diffusion equations: introduction to a review of modern methods. *Math. Models Methods Appl. Sci.*, 24(8):1575–1619, 2014.

[20] E. Gabriel, G.E. Fagg, G. Bosilca, T.Angskun, J.J. Dongarra, J.M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R.H. Castain, D.J. Daniel, R.L. Graham, and T. S. Woodall. Open mpi: Goals, concept, and design of a next generation mpi implementation. In Springer, editor, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, volume 3421 of *Lecture Notes in Computer Science*, 2004.

[21] G. Gallice. Positive and Entropy Stable Godunov-Type Schemes for Gas Dynamics and MHD Equations in Lagrangian or Eulerian Coordinates. *Numer. Math.*, 94(4):673–713, 2003.

[22] G. Gallice, A. Chan, R. Loubère, and P.-H. Maire. Entropy stable and positivity preserving Godunov-type schemes for multidimensional hyperbolic systems on unstructured grid. *J. Comput. Phys.*, 2022.

[23] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[24] P.A. Gnoffo. Point-implicit Relaxation Algorithm for Viscous, Compressible Perfect-Gas Flows. Technical Report 2953, NASA Langley Research Center, 1990.

[25] E. Godlewski and P.-A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer Verlag, 1996.

[26] A. Del Grosso, W. Barsukow, R. Loubère, and P.-H. Maire. An asymptotic-preserving multidimensionality-aware finite volume scheme for euler equations. In *Twelth International Conference on Computational Fluid Dynamics (ICCFD12), Kobe, Japan, July 14-19, 2024*, https://www.iccfd.org/iccfd12/assets/pdf/papers/ICCFD12_Paper_6-A-01.pdf, 2024. submitted to Comput. Fluids.

[27] A. Harten and P.D. Lax. A Random Choice Finite Difference Scheme for Hyperbolic Conservation Laws. *SIAM Journal on Numerical Analysis*, 18(2):289–315, 1981.

[28] A. Harten, P.D. Lax, and B. van Leer. On upstream Differencing and Godunov-Type schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1):35–61, 1983.

[29] Ph. Hoch. An induced limitation in the reconstruction step for the euler equations of compressible gas dynamics in arbitrary dimension. *Comptes Rendus. Mathématiques*, 362:911–935, 2024.
[30] L. Hu and S. Feng. A robust and contact preserving flux splitting scheme for compressible flows. *Commun Nonlinear Sci Numer Simulat*, 93, 2021.
[31] P. Jacq. *Finite Volume methods on unstructured grids for solving anisotropic heat transfer and compressible Navier-Stokes equations*. PhD thesis, Ecole Doctorale de Mathématiques et d'Informatique de l'université de Bordeaux, 2014. Available at https://theses.hal.science/tel-01136442.
[32] P. Jacq, P.-H. Maire, and R. Abgrall. A Nominally Second-Order Cell-Centered Finite Volume Scheme for Simulating Three-Dimensional Anisotropic Diffusion Equations on Unstructured Grids. *Commun. Comput. Phys.*, 16:841–891, 2014.
[33] A. Jameson. Formulation of kinetic energy preserving conservative schemes and direct numerical simulation of one-dimensional viscous compressible flow in a shock tube using entropy and kinetic energy preserving schemes. *J. Sci. Comput.*, 34:188–208, 2008.
[34] 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.
[35] G. Karypis and K. Vipin. A Fast and High Quality Multilevel Scheme for Partioning Irregular Graphs. *SIAM J. Sci. Stat. Comp.*, 20(1):359–392, 1998.
[36] D.S. Kershaw. Differencing of the diffusion equation in Lagrangian hydrodynamic codes. *J. Comput. Phys.*, 39(2):375–395, 1981.
[37] K. Kitamura, P.L. Roe, and F. Ismail. Evaluation of Euler Fluxes for Hypersonic Flow Computations. *AIAA Journal*, 47(1), 2009.
[38] K. Kitamura, E. Shima, Y. Nakamura, and P.L. Roe. Evaluation of Euler Fluxes for Hypersonic Heating Computations. *AIAA Journal*, 48(4), 2010.
[39] A.G. Kulikowskii, N.V. Pogorelov, and A. Yu. Semenov. *Mathematical Aspects of Numerical Solution of Hyperbolic Systems*. Chapman & Hall/CRC, 2001.
[40] L.D Landau and E.M. Lifshitz. *Fluid Mechanics*. Pergamon Press, 1978.
[41] R. Loubère, P.-H. Maire, and B. Rebourcet. *Handbook of Numerical Methods for Hyperbolic Problems: Basic and Fundamental Issues, edited by R. Abgrall and C.-W. Shu*, chapter 13 Staggered and colocated Finite Volume scheme for Lagrangian hydrodynamics, pages 319–352. North Holland, 2016.
[42] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *J. Comput. Phys.*, 228:2391–2425, 2009.
[43] P.-H. Maire. *Contribution to the numerical modeling of Inertial Confinement Fusion*. Habilitation à diriger des recherches, Bordeaux University, 2011. Available at https://tel.archives-ouvertes.fr/file/index/docid/589758/filename/hdr_main.pdf.
[44] P.-H. Maire and J. Breil. A nominally second-order accurate finite volume cell-centered scheme for anisotropic diffusion on two-dimensional unstructured grids. *J. Comput. Phys.*, 231:2559–2299, 2012.
[45] R. Menikoff and B.J. Plohr. The Riemann problem for fluid flow of real materials. *Reviews of Modern Physics*, 61(1), 1989.
[46] J.J. Quirk. A contribution to the great Riemann solver debate. *Int. J. Numer. Methods Fluids*, 18:555–574, 1994.
[47] R. Rabenseifner, G. Hager, and G. Jost. *Hybrid MPI and OpenMP Parallel Programming*. OpenMP, https://openmp.org/wp-content/uploads/HybridPP_Slides.pdf, 2013.
[48] A.V. Rodionov. Artificial viscosity Godunov-type schemes to cure the carbuncle phenomenon. *J. Comput. Phys.*, 345:308–329, 2017.
[49] P.L. Roe. Approximate Riemann solvers, parameter vectors and difference scheme. *J. Comput. Phys.*, 43:357–372, 1981.
[50] L.I. Sedov. *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York, 1959.
[51] Z. Shen, W. Yan, and G. Yuan. A robust and contact resolving Riemann solver on unstructured mesh, Part I, Euler method. *J. Comput. Phys.*, 268:432–455, 2014.
[52] Z. Shen, W. Yan, and G. Yuan. A robust HLLC-type Riemann solver for strong shocks. *J. Comput. Phys.*, 309:185–205, 2016.
[53] C.-W. Shu. Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws. Technical Report ICASE 97-65, NASA, 1997.
[54] E. Tadmor. The numerical viscosity of entropy stable scheme for systems of conservation laws i. *Mathematics of Computation*, 49:91–103, 1987.
[55] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time dependent problems. *Acta Numerica*, 12:451–512, 2003.
[56] C. Tang. Rapid Hypersonic Simulation using US3D and Pointwise. In *Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, Hi, July 11-15, 2022*, https://ntrs.nasa.gov/api/citations/20220010059/downloads/ICCFD11_CTang_v2.pdf, 2024.
[57] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, second edition, 1999.
[58] J.A. Trangenstein. *Numerical solution of hyperbolic partial differential equations*. Cambridge University Press, 2009.
[59] N. Wang, M. Li, R. Ma, and L. Zhang. Accuracy analysis of gradient reconstruction on isotropic unstructured meshes and its effects on inviscid flow simulation. *Advances in Aerodynamics*, 1(1):18, 2019.
[60] G.C. Zha and E. Bilgen. Numerical solutions of Euler equations by using a new flux vector splitting scheme. *Int. J. Numer. Methods Fluids*, 17:115–144, 1993.