Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, Hawaii, USA, July 11-15, 2022

ICCFD11-2022-0303

Flux reconstruction solver for arbitrarily unstructured grids with r-refinement

R. Dhib*, F. Ben Ameur*,**, R. Vandenhoeck*, A. Lani*,** and S. Poedts*,***
Corresponding author: rayan.dhib@kuleuven.be

* Centre for Mathematical Plasma-Astrophysics, KU Leuven
** Von Karman Institute for Fluid Dynamics
*** Institute of Physics, University of Maria Curie-Skłodowska

Abstract: The present paper will address the development and implementation of the high-order Flux Reconstruction (FR) schemes for high-speed flows on both straight and curved edged simplex elements within the open-source COOLFluiD (Computational Object-Oriented Libraries for Fluid Dynamics) platform. While the FR method potentially provides a more accurate detection of complex flow features over relatively coarser mesh, when compared to their low-order peers, the approach suffers from the same pacing items as the other high-order methods such as the slow convergence to steady state and the lack of robust shock capturing capabilities. To overcome such deficiencies, Adaptive Mesh Refinement (AMR) represent a robust procedure for improving the quality of the physical results, especially shock capturing capabilities, due to a local increase of the grid resolution and mesh/shock alignment. Particularly, spring-based and physics-driven r-refinement (r-AMR) requires a compact stencil and is suitable for parallel computing. This fact goes hand in hand with the FR method since the latter can obtain arbitrary high orders of accuracy without requiring a wide stencil at higher orders. In this work, a concise overview of the FR method and spring-based AMR techniques will be given, followed by some promising results of subsonic flow simulations using FR in simplex elements and r-AMR-FR applied to benchmark high-order supersonic test cases.

Keywords: Flux Reconstruction, r-AMR, high-speed flows, Spring analogy, simplex meshes.

1 Introduction

High-order methods have drawn considerable attention among the CFD community, mainly, the Flux Reconstruction (FR) method. FR, originally developed by Huynh for 1D advection problem, is a framework which allows to develop new high-order schemes while being simple and computationally efficient, particularly on graphical processor units (GPU) [1, 2]. Despite the established advantages of the FR method [3], the latter suffers from the lack of robust shock capturing methods. In order to overcome such deficit, Adaptive Mesh Refinement (AMR) procedure is used to ease the shock capturing.
Previous to this work, a state-of-the-art fully implicit high-order FR code has been implemented for quadrilateral and hexahedral meshes and used to solve Euler/Navier-Stokes equations, where its structure is extremely modular and can be easily coupled to arbitrary sets of advection-diffusion-reaction Partial Differential Equations (PDEs) [4, 5, 6]. In addition, a sub-cell order-dependent spring analogy is developed and applied for quadrilateral steady-state FR test cases. The latter module is parallel and physics-independent, letting the user decide which monitor physical quantity to use for driving the adaptation according to the application [7, 8].
In this work, we extend the existing FR code to deal with triangular meshes, develop a r-refinement sub-cell order-dependent spring analogy within the aforementioned FR solver. The resulting solver is implemented within the COOLFluiD platform [9, 10]. The authors invite the readers to consult the COOLFuiD's website to

have more detail about the FR solver and r-AMR module (https://github.com/andrealani/COOLFluiD/wiki).

The paper is structured as follows:
- Sec.2 and Sec.3 reviews the state-of-the-art of FR method for triangular elements and Vincent-Castonguay-Jameson-Huynh Schemes applied to triangular elements;
- Sec.4 covers the main aspects of the implementation of FR for triangles within the COOFLuiD platform;
- Sec.5 presents spring based r-AMR and the sub-cell order-dependent spring analogy;
- Sec.6 shows very promising results for the FR-AMR solver, for low- and high-speed flows.

2 Extension of the FR approach to triangles

In this section, a review of the FR approach on triangles is presented based on [11, 12, 13, 14, 15]. The latter was used as reference for the implementation of the solver.

Consider solving the 2D scalar conservation law within an arbitrary domain Ω:

$$\frac{\partial u}{\partial t} + \nabla_{xy}\mathbf{f} = 0 \;\; ; \;\; \nabla_{xy}\mathbf{f} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y}, \tag{1}$$

where x and y are spatial coordinates, t is the temporal variable, $u = u(x, y, t)$ is a conserved scalar and $\mathbf{f} = (f, g)$ with $f = f(u)$ and $g = g(u)$ being the fluxes of u in the x and y directions respectively. Similarly to the 1D case, one considers partitioning the domain Ω into N non-overlapping, conforming linear triangular elements Ω_n.

Following the same analogy as the 1D case, the exact solution u within each element Ω_n is represented by an approximate solution $u_n^\delta = u_n^\delta(x, y, t)$, a polynomial of degree P within Ω_n and identically zero outside the element. Likewise, the exact flux $\mathbf{f}$ within each element is represented by a function $\mathbf{f}_n^\delta = (f_n^\delta, g_n^\delta) = \mathbf{f}_n^\delta(x, y, t)$, a polynomial of degree $P+1$ within Ω_n and identically zero outside the element. Thus, the total approximate solution $u^\delta = u^\delta(x, y, t)$ and a total approximate flux $\mathbf{f}^\delta$ are defined as in the 1D case using Eq.2:

$$u(x, y, t) \approx u^\delta(x, y, t) = \sum_{n=1}^{N} u_n^\delta(x, y, t) \text{ and } f(x, y, t) \approx f^\delta(x, y, t) = \sum_{n=1}^{N} f_n^\delta(x, y, t). \tag{2}$$

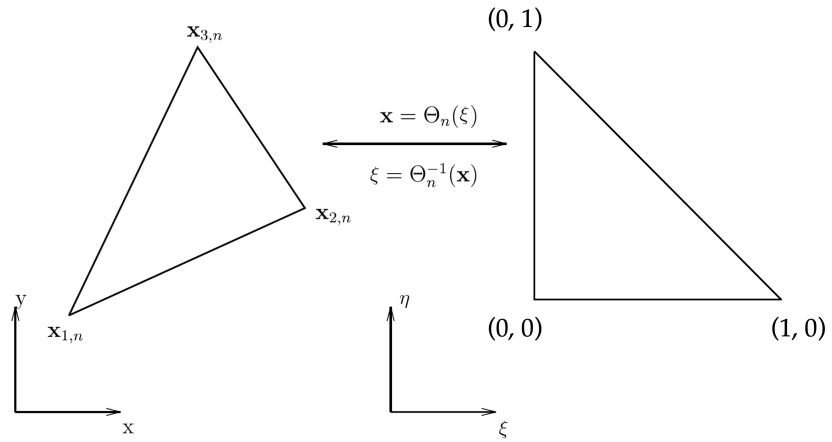


Figure 1: Mapping between the physical space (x, y) and the computational space (ξ, η)

To facilitate the implementation of the FR approach in triangles, each element Ω_n of the mesh is mapped

to a reference element Ω_S, a right angle triangle as depicted in Fig.1, using the mapping Θ_n such that:

$$\mathbf{x} = \mathbf{\Theta}_n(\xi) = (1 - \xi - \eta)\ \mathbf{x}_{1,n} + \xi\ \mathbf{x}_{2,n} + \eta\ \mathbf{x}_{3,n}, \tag{3}$$

where $\mathbf{x}_{1,n}$, $\mathbf{x}_{2,n}$ and $\mathbf{x}_{3,n}$ are the coordinates of the three vertices of the triangular element Ω_n in the physical space.

The governing Eq.1 in the physical domain can be transformed to the equivalent governing equation in the reference domain. The solution u_n^δ within each element Ω_n can thus be obtained by solving the transformed conservation equation within the reference element Ω_S:

$$\frac{\partial \hat{u}^\delta}{\partial t} + \nabla_{\xi\eta} \cdot \hat{\mathbf{f}}^\delta = 0, \tag{4}$$

where

$$\hat{u}^\delta = \hat{u}^\delta(\xi, t) = J_n u_n^\delta \left(\mathbf{\Theta}_n(\xi), t\right), \tag{5}$$

$$\hat{\mathbf{f}}^\delta = \hat{\mathbf{f}}^\delta(\xi, t) = \left(\hat{f}^\delta, \hat{g}^\delta\right) = \left(\frac{\partial y}{\partial \eta} f_n^\delta - \frac{\partial x}{\partial \eta} g_n^\delta, -\frac{\partial y}{\partial \xi} f_n^\delta + \frac{\partial x}{\partial \xi} g_n^\delta\right). \tag{6}$$

J_n represents the Jacobian and can be expressed as follows:

$$J_n = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \xi}. \tag{7}$$

The terms J_n, $\frac{\partial x}{\partial \xi}$, $\frac{\partial x}{\partial \eta}$, $\frac{\partial y}{\partial \xi}$ and $\frac{\partial y}{\partial \eta}$ used in the previous equations depend on the shape of the element Ω_n and can thus be evaluated using Eq.3.
Let Γ_n and Γ_S refer to the boundary of the physical element Ω_n and the reference element Ω_s, respectively. One can thus define $P_p(\Omega_S)$ the space of polynomials of degree $\leq P$ on Ω_S, where the dimension of $P_p(\Omega_S)$ is $\frac{1}{2}(P+1)(P+2)$. Another required definition for the extension of the FR approach to triangles is the polynomial space $R_p(\Gamma_S)$ on the edges of the reference element defined as:

$$R_P\left(\mathbf{\Gamma}_S\right) = \left\{\phi \,|\phi \in L^2\left(\mathbf{\Gamma}_S\right), \phi\big|_{\Gamma_f} \in P_P\left(\mathbf{\Gamma}_f\right), \forall \mathbf{\Gamma}_f\right\}, \tag{8}$$

with Γ_f being edge f of the reference element Ω_S. This implies that functions of $R_p(\Gamma_S)$ are polynomials of degree $\leq P$ on each side of the standard element, and are not necessarily continuous at the vertices. As in the 1D approach, the 2D FR approach requires sevens steps.

Stage 1

The first stage consists of representing the approximate solution $\hat{u}^\delta$ within the reference element Ω_S by a multi-dimensional polynomial of degree P, defined by its values at a set of $N_p = \frac{1}{2}(P+1)(P+2)$ solution points. These solution points are represented in Fig.2 as well as the flux points which are located at the Gauss quadrature points.

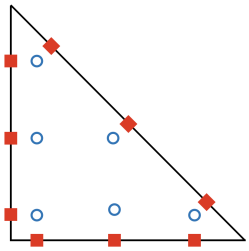


Figure 2: Distribution of solution points (circles) and flux points (squares) in the reference element for $P = 2$

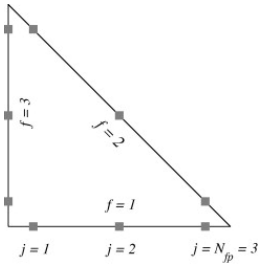


Figure 3: Numbering convention for the faces and flux points on the reference triangular element ($P = 2$) [15]

The approximate solution $\hat{u}^{\delta}$ in the reference element lies in the space $P_p(\Gamma_S)$ and can then be expressed as:

$$\hat{u}^{\delta}(\boldsymbol{\xi}, t) = \sum_{i=1}^{Np} \hat{u}_i^{\delta} l_i(\boldsymbol{\xi}), \tag{9}$$

where $\hat{u}_i^{\delta} = J_n \cdot \hat{u}^{\delta}\left(\Theta_n^{-1}\left(\boldsymbol{\xi}_i\right), t\right)$ is the value of $\hat{u}^{\delta}$ at the ith solution point and $l_i(\boldsymbol{\xi})$ is the multi-dimensional Lagrange polynomial associated with the considered solution point in the reference triangle Ω_S.

Stage 2

The second stage of the 2D FR approach involves determining the interface fluxes at the $N_{fp} = P + 1$ flux points along the standard element edges. Beforehand, it is necessary to present the convention used to number faces and flux points as illustrated in Fig.3. Let the indices f, j refer to the flux point j on the face f, where $1 \leq f \leq 3$ and $1 \leq j \leq N_{fp}$. Similarly to the 1D approach, the interface fluxes are computed by first evaluating the multiple defined values of $\hat{u}^{\delta}$ at each flux point using Eq.9. At each flux point, $\hat{u}_-^{\delta}$ is defined as the value of $\hat{u}^{\delta}$ computed using the local information of the element and $\hat{u}_+^{\delta}$ as the value of $\hat{u}^{\delta}$ computed using the information of the neighbouring element that shares the considered flux point.

After computing the approximate solution values on both sides ($\hat{u}_-^{\delta}$ and $\hat{u}_+^{\delta}$), one can then compute the common interface value $\hat{u}_{f,j}^{\delta I}$ associated to each flux point. The approach used to calculate such numerical interface fluxes can be one of the well-known approaches including the Central Flux (CF) [16], Local Discontinuous Galerkin (LDG) [17], Compact Discontinuous Galerkin (CDG) [18], and Internal Penalty (IP) [19] approaches. Note that the Central Flux approach is a form of the commonly used Bassi-Rebay approach (BR2) discussed in the 1D FR approach.

Stage 3

In this stage, the aim is to construct a transformed solution correction $\hat{u}^{\delta C} = \hat{u}^{\delta C}(\boldsymbol{\xi}, t)$ to approximate the transformed discontinuous solution $\hat{u}^{\delta D}$ such that the sum equals the transformed numerical interface value at the flux points ($\hat{u}^{\delta D}$ being defined by Eq.9). To do so, a vector correction function $\mathbf{h}_{f,j}(\boldsymbol{\xi})$, associated with each flux point f, j, is defined. Each vector correction function $\mathbf{h}_{f,j}(\boldsymbol{\xi})$ is restricted to lie in the Raviart-Thomas space [20] of order P, denoted by $RT_p(\Omega_S)$. The latter property implies that:

$$\begin{array}{c} \hat{\nabla} \cdot \mathbf{h}_{f,j} \in P_P\left(\boldsymbol{\Omega}_S\right) \\ \mathbf{h}_{f,j} \cdot \hat{\mathbf{n}}|_{\Gamma_S} \in R_P\left(\boldsymbol{\Gamma}_S\right) \end{array}, \tag{10}$$

meaning that the divergence of each correction function $\hat{\nabla} \cdot \mathbf{h}_{f,j}$ is a polynomial of degree $\leq P$ and that the normal trace $\mathbf{h}_{f,j} \cdot \hat{\mathbf{n}}$ on Γ_S is also a polynomial of degree $\leq P$ along each edge. For the sake of simplicity, the correction field $\phi_{f,j}(\boldsymbol{\xi})$ defined as the divergence of the correction function $\mathbf{h}_{f,j}(\boldsymbol{\xi})$ is introduced such as:

$$\phi_{f,j}(\boldsymbol{\xi}) = \hat{\nabla} \cdot \mathbf{h}_{f,j}(\boldsymbol{\xi}) \tag{11}$$

The vector correction functions $\mathbf{h}_{f,j}$ verify the following property:

$$\mathbf{h}_{f,j}\left(\boldsymbol{\xi}_{f_2,j_2}\right)\cdot\hat{\mathbf{n}}_{f_2,j_2}=\begin{cases}1 & \text{if } f=f_2 \text{ and } j=j_2\\ 0 & \text{if } f\neq f_2 \text{ or } j\neq j_2\end{cases} \tag{12}$$

Once the vector correction functions are chosen, an expression for the transformed solution correction $\hat{u}^{\delta C}$ can be constructed on each face such that it verifies the following equation:

$$\begin{aligned}\hat{u}^{\delta C}\big|_{\Gamma_f}&=\sum_{j=1}^{N_{fp}}\left[\hat{u}_{f,j}^{\delta I}-\hat{u}_{f,j}^{\delta D}\right]\left[\mathbf{h}_{f,j}\cdot\hat{\mathbf{n}}_{f,j}\right]\\&=\sum_{j=1}^{N_{fp}}\Pi_{f,j}\left[\mathbf{h}_{f,j}\cdot\hat{\mathbf{n}}_{f,j}\right],\end{aligned} \tag{13}$$

with $\Pi_{f,j}$ the difference between the transformed numerical flux (interface solution) and the transformed discontinuous solution on face f at flux point j.

Stage 4

The fourth stage of the FR approach in triangles deviates from the 1D case. In fact, this stage involves constructing the solution gradient $\hat{q}^{\delta D}$ expressed in the 2D case as:

$$\hat{\mathbf{q}}^{\delta D}=\hat{\nabla}\hat{u}^{\delta}=\hat{\nabla}\hat{u}^{\delta D}+\hat{\nabla}\hat{u}^{\delta C} \tag{14}$$

whereas, in the 1D formulation it was determined by applying the operator $\frac{\partial}{\partial\xi}$ to $\hat{u}^{\delta D}$ and $\hat{u}^{\delta C}$. The difference does not reside only in the aforementioned detail, but also in the definition of $\hat{u}^{\delta C}$ itself. As previously mentioned, $\hat{u}^{\delta C}$ depend on the vector correction function and there is not a unique analytical form for $\mathbf{h}_{f,j}$. Furthermore, $\hat{u}^{\delta C}$ is only defined on the boundary Γ_S of the element, while $\hat{q}^{\delta D}$ has to be defined within the standard element Ω_S. Thus, to obtain a proper form for $\hat{q}^{\delta D}$, an alternative approach for forming the correction gradient is used:

$$\hat{\nabla}\hat{u}^{\delta C}=\sum_{f=1}^{3}\sum_{j=1}^{N_{fp}}\Pi_{f,j}\psi_{f,j}\hat{\mathbf{n}}_{f,j}. \tag{15}$$

The new term $\psi_{f,j}$ is a correction field introduced to transform $\hat{u}^{\delta C}$ defined on Γ_S into $\hat{\nabla}\hat{u}^{\delta C}$ defined within the element Ω_S. The definition and details of forming $\psi_{f,j}$ are developed in [12]. Using Eq.15, on can express $\hat{q}^{\delta D}$ as:

$$\hat{\mathbf{q}}^{\delta D}=\hat{\nabla}\hat{u}^{\delta D}+\hat{\nabla}\hat{u}^{\delta C}=\sum_{i=1}^{N_p}\hat{u}_i^{\delta D}\hat{\nabla}l_i(\boldsymbol{\xi})+\sum_{f=1}^{3}\sum_{j=1}^{N_{fp}}\Pi_{f,j}\psi_{f,j}\hat{\mathbf{n}}_{f,j}. \tag{16}$$

For each of its components, $\hat{\mathbf{q}}^{\delta D}=(\hat{q}_\xi^{\delta D},\hat{q}_\eta^{\delta D})$ is represented by a degree P polynomial such as:

$$\hat{q}_\xi^{\delta D}=\sum_{i=1}^{N_p}\left(\hat{q}_\xi^{\delta D}\right)_i l_i(\boldsymbol{\xi})\qquad \hat{q}_\eta^{\delta D}=\sum_{i=1}^{N_p}\left(\hat{q}_\eta^{\delta D}\right)_i l_i(\boldsymbol{\xi}). \tag{17}$$

After obtaining $\hat{\mathbf{q}}^{\delta D}$, one can then compute the approximate discontinuous flux using $\hat{\mathbf{f}}^{\delta D}=\hat{\mathbf{f}}^{\delta D}(\hat{u}^{\delta D},\hat{\mathbf{q}}^{\delta D})$. Likewise, for both of its components, the approximate discontinuous flux $\hat{\mathbf{f}}^{\delta D}=(\hat{f}^{\delta D},\hat{g}^{\delta D})$ can be expressed with a degree P polynomial as

$$\hat{f}^{\delta D}=\sum_{i=1}^{N_p}\hat{f}_i^{\delta D}l_i,\quad \hat{g}^{\delta D}=\sum_{i=1}^{N_p}\hat{g}_i^{\delta D}l_i. \tag{18}$$

The coefficients $\hat{f}_i^{\delta D}$ and $\hat{g}_i^{\delta D}$ are the values of the transformed flux evaluated at the ith solution point using the approximate solution $\hat{u}_i$ and the auxiliary gradient $\hat{\mathbf{q}}_i$, i.e $\hat{f}_i^{\delta D} = \hat{f}(\hat{u}_i, \hat{\mathbf{q}}_i)$ and $\hat{g}_i^{\delta D} = \hat{g}(\hat{u}_i, \hat{\mathbf{q}}_i)$.

Stage 5

As in the 1D case, the fifth stage is similar to the second one except that the procedure is applied to compute the numerical interface fluxes $\hat{\mathbf{f}}_{f,j}^{\delta I}$ at the flux points along the edges of the standard element Ω_S. To that end, the multiple defined values of $\hat{\mathbf{q}}^{\delta D}$ are evaluated at each flux point using Eq.17 to obtain $\hat{\mathbf{q}}_-^{\delta D}$ and $\hat{\mathbf{q}}_+^{\delta D}$. In addition, the approximate solution is computed at the edges of the element to obtain $\hat{u}_-^{\delta D}$ and $\hat{u}_-^{\delta D}$ using Eq.9.

By combining the latter variables, on can compute the discontinuous fluxes $\hat{\mathbf{f}}_-^{\delta D}$ and $\hat{\mathbf{f}}_+^{\delta D}$ at the element boundary. Once both approximate fluxes are computed at each flux point the same approach as chosen in stage 2 is used to compute a common numerical normal flux $(\mathbf{f} \cdot \mathbf{n})_{f,j}^{\delta I}$.

Stage 6

The sixth stage of the FR approach consists in computing the transformed correction flux $\hat{\mathbf{f}}^{\delta C}$ to compensate for the difference between the discontinuous flux and the values reached at the edges of the element. As in the 1D case the goal is that the sum of the discontinuous flux $\hat{\mathbf{f}}^{\delta D}$ and the correction flux $\hat{\mathbf{f}}^{\delta C}$ equals the continuous flux $\hat{\mathbf{f}}^{\delta}$ within the element Ω_S and the interface flux $\hat{\mathbf{f}}^{\delta I}$ at the edges of Ω_S, i.e at the flux points. The correction flux $\hat{\mathbf{f}}^{\delta C}$ is expressed as:

$$\begin{aligned} \hat{\mathrm{f}}^{\delta C}(\boldsymbol{\xi}) &= \sum_{f=1}^{3} \sum_{j=1}^{N_{fp}} \left[(\hat{\mathbf{f}} \cdot \hat{\mathbf{n}})_{f,j}^{\delta I} - \left(\hat{\mathbf{f}}^{\delta D} \cdot \hat{\mathbf{n}} \right)_{f,j} \right] \mathbf{h}_{f,j}(\boldsymbol{\xi}) \\ &= \sum_{f=1}^{3} \sum_{j=1}^{N_{fp}} \Delta_{f,j} \mathbf{h}_{f,j}(\boldsymbol{\xi}) \end{aligned} , \tag{19}$$

where $\mathbf{h}_{f,j}(\boldsymbol{\xi})$ is the correction function depending on the considered flux point and $\Delta_{f,j}$ is defined as the difference between the normal transformed numerical flux and the normal transformed discontinuous flux at the flux point j on the face f. Therefore, by combining Equations 19 and 18, on can obtain the continuous flux $\hat{\mathbf{f}}^{\delta}$.

Stage 7

The seventh and final stage of the FR approach in triangles consists in computing the divergence of the continuous flux $\hat{\mathbf{f}}^{\delta}$. The obtained result will then be used to update the solution at the point $\boldsymbol{\xi}_i$ as follows:

$$\begin{aligned} \frac{d\hat{u}_i^{\delta}}{dt} &= - \left(\hat{\nabla} \cdot \hat{\mathbf{f}}^{\delta} \right) \Big|_{\boldsymbol{\xi}_i} \\ &= - \left(\hat{\nabla} \cdot \hat{\mathbf{f}}^{\delta D} \right) \Big|_{\boldsymbol{\xi}_i} - \left(\hat{\nabla} \cdot \hat{\mathbf{f}}^{\delta C} \right) \Big|_{\boldsymbol{\xi}_i} \\ &= - \sum_{k=1}^{N_p} \left(\hat{f}_k^{\delta D} \right) \frac{\partial l_k}{\partial \xi} \Big|_{\boldsymbol{\xi}_i} - \sum_{k=1}^{N_p} \left(\hat{g}_k^{\delta D} \right) \frac{\partial l_k}{\partial \eta} \Big|_{\boldsymbol{\xi}_i} - \sum_{f=1}^{3} \sum_{j=1}^{N_{fp}} \Delta_{f,j} \phi_{f,j} \left(\boldsymbol{\xi}_i \right) \end{aligned} \tag{20}$$

The nature of the FR approach when applied to triangles depends on four factors [12]: the location of the solution collocation points $\boldsymbol{\xi}_i$, the location of the flux points $\boldsymbol{\xi}_{f,j}$, the methodology for calculating the numerical fluxes $\hat{\mathbf{u}}_{f,j}^{\delta I}$ and $\hat{\mathbf{f}}_{f,j}^{\delta I}$ and finally the form of the correction fields $\phi_{f,j}$.

In the following sections, the location of the flux points $\boldsymbol{\xi}_{f,j}$ is defined using Gauss-Legendre quadrature [21]. Regarding the location of the solution collocation points $\boldsymbol{\xi}_i$, quadrature points are used as presented in [22].

3 Vincent-Castonguay-Jameson-Huynh Schemes

As stated in the previous sections, the stability of the FR approach is highly influenced by the choice of the correction function and their associated correction fields. Vincent-Castonguay-Jameson-Huynh (VCJH) schemes specify a stability-preserving form for the correction functions and fields as presented in [14] and detailed in [23]. In this section, an adapted version of an energy-stable VCJH approach for selecting the correction field $\phi_{f,j}$ will be summarized since it was used as a reference for its implementation. Indeed, the approach presented in [15] is applicable to right triangles defined between $[-1,1]$ while the reference element considered is defined between $[0,1]$.

Although a closed form expression of the correction functions $\mathbf{h}_{f,j}$ and their divergence $\phi_{f,j}$ has not been found (unlike the 1D case), the coefficients of the polynomials defining each correction field $\phi_{f,j}$ can be obtained from the solution of a system of equation [15]. To that end, some preliminary definitions are needed, such as the operator $D^{(v,w)}$ defined as

$$D^{(v,w)} = \frac{\partial^w}{\partial \xi^{(w-v+1)} \partial \eta^{(v-1)}}, \tag{21}$$

where v and w are integers verifying $1 \leq v \leq w+1$. The 2D orthonormal Dubiner basis [24] is also needed, which is given by

$$L_k(\boldsymbol{\xi}) = \frac{\sqrt{2}}{4} Q_v(a) Q_w^{(2v+1,0)}(b)(1-b)^v \tag{22}$$

with

$$k = w + (P+1)v + 1 - \frac{v}{2}(v-1), \qquad (v,w) \geq 0; \qquad v + w \leq P \tag{23}$$

$$a = \frac{2\xi}{1-\eta} - 1, \qquad b = 2\eta - 1 \tag{24}$$

and $Q_n^{(\alpha,\beta)}$ is the normalized Jacobi polynomial of order n. The Dubiner basis expressed in Eq.22 is orthonormal on the reference triangle Ω_S. Finally, the VCJH scheme coefficients (c_m's) are defined as

$$c_m = \kappa \binom{P}{m-1}, \tag{25}$$

where κ is a free parameter which will be specified later and the binomial coefficient of degree $m-1$ is given by

$$\binom{P}{m-1} = \frac{P!}{(m-1)!(P-(m-1))!}. \tag{26}$$

Being polynomials of degree P, the correction fields $\phi_{f,j}$ can be expressed as

$$\phi_{f,j} = \sum_{k=1}^{N_p} \sigma_k L_k(\boldsymbol{\xi}), \tag{27}$$

where σ_k are the expansion coefficients. In a VCJH scheme, these coefficients corresponding to each correction field $\phi_{f,j}$ can be obtained by solving the following system of N_p equations for the unknowns σ_k :

$$\sum_{k=1}^{N_p} \sigma_k \sum_{m=1}^{P+1} c_m \left(D^{(m,P)} L_i\right)\left(D^{(m,P)} L_k\right) = -\sigma_i + \int_{\Gamma_S} \left(\mathbf{h}_{f,j} \cdot \hat{\mathbf{n}}\right) L_i d\Gamma, \quad \text{for } 1 \leq i \leq N_p, \tag{28}$$

where κ is the free parameter that must lie within the range $0 \leq \kappa \leq \infty$. The integral term on the right hand side of Eq.28 can be evaluated using Eq.12 and knowing that $\mathbf{h}_{f,j} \cdot \hat{\mathbf{n}}$ is a polynomial of degree P on the edges of Ω_S.

The correction field obtained by solving the system of equations (i.e. Eq.28) and using Eq.27 results in an energy stable Flux Reconstruction scheme on triangles. This scheme is parameterized by a single scalar

parameter κ. Figure 4 represents an example of a correction field $\phi_{2,1}$ computed on the defined reference element Ω_S compared to reference data (the free paramter used is the same as in the reference [25]).

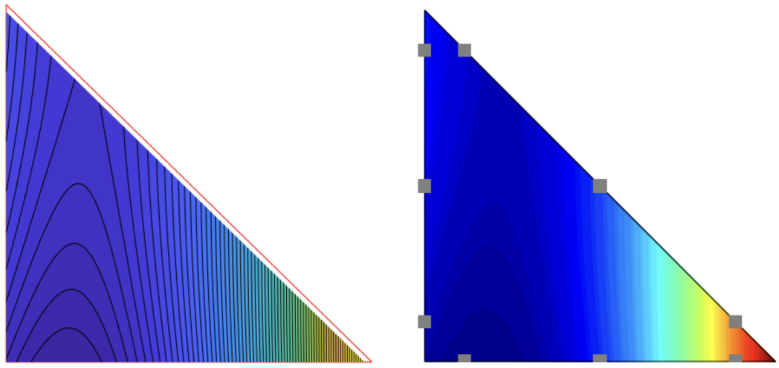


Figure 4: Plots of the VCJH correction field $\phi_{2,1}$ associated the flux point $j = 1$, $f = 2$ and $p = 2$ for a parameter $\kappa = 3.13 \cdot 10^{-2}$ (right: adapted version, left: reference).

4 Overview of the FR solver

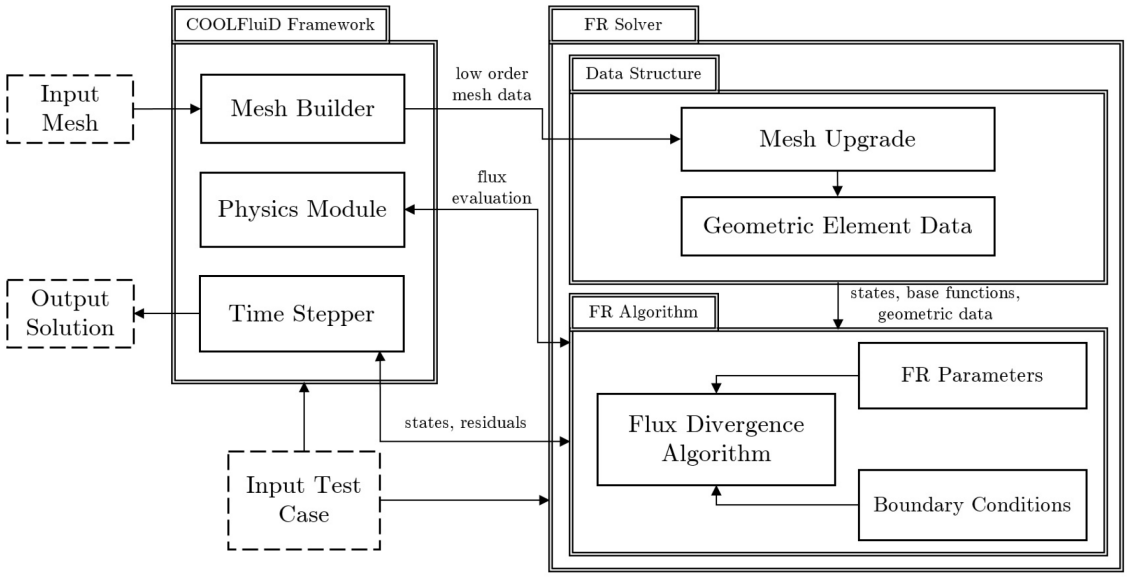


Figure 5: Illustration of the interactions between the FR solver and the COOLFluiD Framework [4]

The presented FR solver handles unstructured grids of 2D triangles and is implemented as a plug-in of COOLFluiD. COOLFluiD is a scientific high-performance computing platform written in C++, offering a component-based framework and handling complex multi-physics simulations. Since the implementation of the present solver is based on (and linked to) the existing FR solver that handles 2D quadrilaterals and 3D hexahedra, some aspects of the following description will include details on both solvers.
The FR solver is implemented to solve the compressible Euler and Navier-stokes equations in 2D (and 3D). A particular feature in COOLFluiD is that the algorithms that depend on the physics of a problem are decoupled from the numerical algorithms, meaning that the solver has been developed to solve advection and advection-diffusion problems in general. All the required physics-dependent algorithms for the Euler and Navier-Stokes equations were already implemented, and other physics-dependent functions can be added to extend the functionality of the solver.

As detailed in previous sections, the FR method is mainly characterized by the solution and flux point distributions, the choice of the correction function (and field) and the interface flux scheme. The implementation of the FR solver on triangles employs a Gauss-Legendre distribution for both flux and solution points. The latter point distributions are defined in an appropriate C++ class within the solver facilitating the addition of other distributions if needed by simply defining new classes. The VCJH scheme for the correction functions (discussed in Sec.3) has been implemented in the present solver for triangles (quadrilaterals and hexahedra were already done). The existing FR implementation included several interface flux schemes, namely: Centered flux [26], Lax-Friedrichs flux [27], Roe flux [28], AUSM$^+$ (Advection Upstream Splitting Method) [29] and AUSM$^+$-up flux [30]. The order of the FR method can be chosen by the user knowing that it has been implemented from $P = 0$ (1 order accuracy in space) up to $P = 9$ (i.e. 10 order accuracy in space). Another noteworthy feature of COOLFluiD is that the solver can either be executed serially or in parallel.
The implemented FR solver consists of two principal components: the FR data structure and the FR algorithm. The first component's main role is to create the data that defines the high-order unstructured mesh comprising geometric entities (e.i. elements and faces), their connectivity and their properties.The second component, also called core code, uses the created data to perform the computations and gives a numerical solution. These two components interact with the existing COOLFluiD framework as represented in Fig.5. The main block of the FR solver is the computation of the divergence of the reconstructed fluxes for each element and time step. The interaction with the COOLFluiD framework is also illustrated by Fig.5, and more precisely, the use of the physics module to evaluate the physical fluxes based on $\mathbf{u}$ and $\boldsymbol{\nabla}\mathbf{u}$.

Knowing that the Navier-Stokes and Euler equations can be written as a system of partial differential equations in the conservative form (cf. [4]), the flux can be split up into a convective and a diffusive part. Therefore, the algorithm consists of two major parts: the convective algorithm and the diffusive algorithm. As shown in Fig.6, both algorithms interact with other classes defining the parameters of the FR method set by the user, namely: the interface flux scheme, the correction function type and the geometric data (i.e. flux and solution point distribution). In addition, a shock capturing algorithm based on a modified Localized Laplacian Artificial Viscosity (LLAV) scheme combined with a positivity preservation method were previously implement as detailed in [4]. Their respective codes were extended to triangles in the present work in order to alleviate oscillations caused by the Gibbs phenomenon.

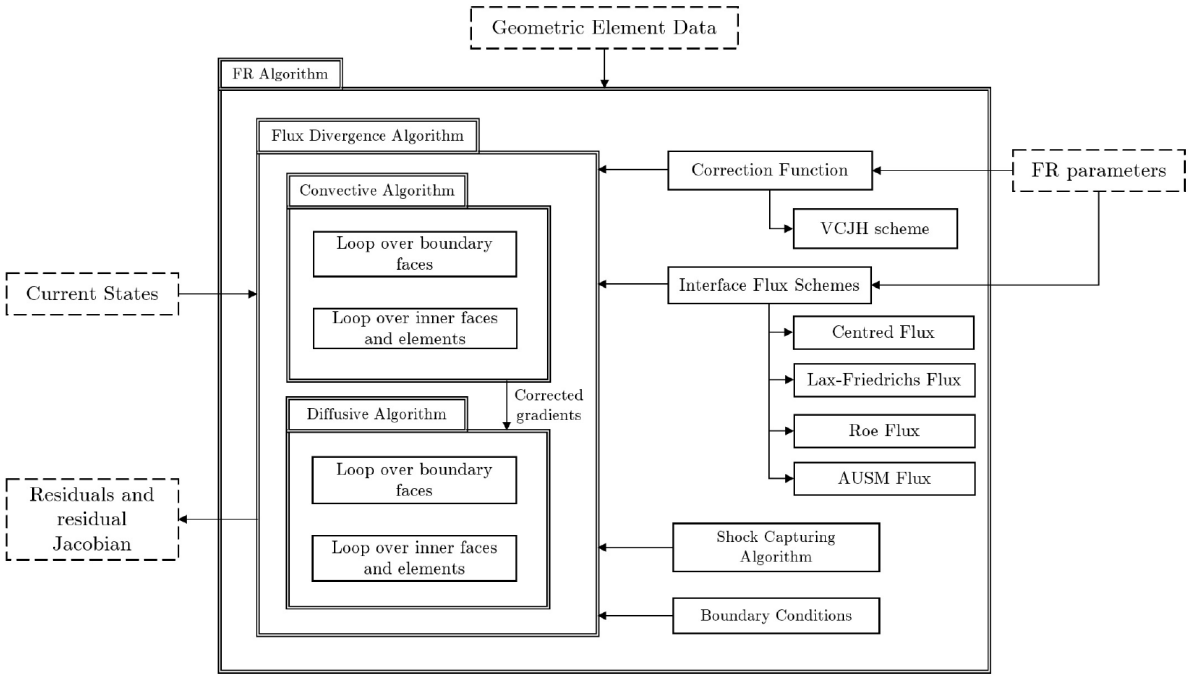


Figure 6: General structure of the FR algorithm taken from [4].

5 Spring-Based Adaptive Mesh Refinement

In [7, 8], the authors have proposed and developed an r-refinement adaptation algorithm based on a user-defined flow field variable, e.g. density or pressure, that relies upon the solution of pseudo-elastic systems associated to the given mesh. The same philosophy is applicable to the FR triangular framework. Consequently, a brief overview is given in the following.
Let $n \in \mathbb{N}$ be the number of the nodes in a mesh $\mathcal{M}$ and let $\mathbf{P}$ be the set of the nodes positions inside $\mathcal{M}$. Let $\mathbf{L}$ be the incidence matrix defined as:

$$L_{ij} = \begin{cases} 1, & \text{if nodes } i \text{ and } j \text{ are edge-connected} \\ 0, & \text{otherwise.} \end{cases}$$

To achieve the equidistribution condition of r-refinement, the line integral I, expressed in Eq.29, must be constant.

$$I = \int_0^1 W(\mathbf{r}(s)) \cdot r'(s) ds = \text{constant}, \tag{29}$$

where $\mathbf{r}(s) = \mathbf{P_i} + s(\mathbf{P_j} - \mathbf{P_i})$ is the parametrization of the edge connecting nodes i and j in function of the parameter $s \in [0, 1]$. Eq.29 is the solution of the Euler-Lagrange equation to the minimization of the energy which reads:

$$E_{ij} = L_{ij} \int_0^1 W(\mathbf{r}(s))(\mathbf{P_j} - \mathbf{P_i})^2 ds, \tag{30}$$

Considering the weight function as a constant, the energy equation can be re-written in a form analogous to the classical spring potential energy equation. In Eq.31, we can identify W_{ij} as the stiffness coefficient between nodes i and j and an equilibrium spring length set to zero.

$$E_{ij} = L_{ij} W_{ij} (\mathbf{P_j} - \mathbf{P_i})^2, \tag{31}$$

The simplest optimization problem depends on finding the equilibrium positions between two adjacent nodes in the mesh $\mathcal{M}$ based on a network of springs:

$$\frac{\partial E}{\partial \mathbf{P}} = 0 \qquad \& \qquad \frac{\partial^2 E}{\partial \mathbf{P}^2} > 0. \tag{32}$$

The analytic Jacobian is defined as:

$$\frac{\partial E_{ij}}{\partial \mathbf{P_i}} = -2 L_{ij} W_{ij} (\mathbf{P_j} - \mathbf{P_i}) = 0. \tag{33}$$

After simplifying the constant and collecting the contributions of each node, we obtain:

$$\sum_{j=1}^{n} L_{ij} W_{ij} (\mathbf{P_j} - \mathbf{P_i}) = 0. \tag{34}$$

The resulting linear system, i.e. the pseudo-elastic system, to be solved using the Generalized Minimal RESidual (GMRES) algorithm complemented by a parallel Additive Schwartz Preconditioner as provided by the PETSc toolkit [31, 32, 33, 34], is expressed in Eq.35.

$$A_{[n \times n]} \mathbf{P}_{[n \times 1]} = b_{[n \times 1]}, \tag{35}$$

where

$$A_{ij} = \begin{cases} -L_{ij} W_{ij}, & if\ i \neq j \\ \sum_{j=1}^{n} L_{ij} W_{ij}, & if\ i = j. \end{cases}$$

and $\mathbf{b}$ contains the non-homogeneous terms of the Dirichlet or the Neumann boundary conditions. As a

result, the nodal re-positioning obeys to the following relation:

$$\mathbf{P}^{k+l} = (1-\omega)\mathbf{P}^{k} + \omega\mathbf{P}_{new}, \tag{36}$$

where $\mathbf{P}^k$ is the old node position at iteration k, $\mathbf{P}^{k+l}$ is the new *relaxed* node position at iteration $k+l$, with $l > 0$ and $\mathbf{P}_{new}$ is computed from Eq.35. Within this work, we will mainly target the linear spring analogy. To achieve that goal, the physics-driven linear stiffness can be used based on the selected flow variables gradients. Eq.37 expresses the linear spring stiffness coefficient, k_{ij}^L, between two edge-connected nodes i and j with a respective nodal state U_i and U_j. The weight function introduced previously is described as:

$$W_{ij} = k_{ij}^L = |U_j - U_i|. \tag{37}$$

Choosing such stiffnesses ensure that the r-refinement algorithm is physics-driven because the spring constants will vary depending on the local physical properties of the flow. Nevertheless, supersonic flows display high variations of physical variables over narrow regions and, for the nodes located in the vicinity of shocks, the values of the spring constants computed with Eq.37 can reach high values. For that reason, in our implementation, these values are truncated and bounded thanks to a P^2 algorithm, which is based on the use of quantiles (the p-quantile of a distribution is defined as the value below which 100 % of the distribution lies) [35]. The P2 algorithm gets all the spring constants one after the other and dynamically computes truncated spring values based on user-defined minimum and maximum quantiles.

5.1 Boundary nodes

Two types of the boundary conditions are defined within our r-AMR algorithms:

1. Dirichlet (i.e. locked node) where the node position is kept constant:

$$P_i^m = P_i^0,$$

2. Neumann (i.e. moving node in boundary) where only the tangential displacement is allowed. Mathematically, this implies:

$$\frac{\partial(\mathbf{P_i} \cdot \mathbf{n_i})}{\partial \mathbf{x}} = 0,$$

where $\mathbf{n_i}$ is the boundary face normal vector.

5.2 Sub-cell order-dependent spring analogy

5.2.1 Concept

The sub-cell order-dependent spring analogy, developed by F. Ben Ameur et al. in [8], exploits the FR properties, mainly taking the advantage of the presence of the flux points. The key idea is to place fictitious springs, between the flux points and the end nodes of the edge as shown in Fig.7. The equivalent edge stiffness is computed as :

$$k_{\text{eq}} = k_{\text{eq,L}} + k_{\text{eq,R}}, \tag{38}$$

where, $k_{\text{eq,L/R}} = \sum_i^N k_{i,i+1,\text{L/R}}$, N equals the number of springs placed in series and $k_{i,i+1,\text{L}}$ is the stiffness of the spring placed between point i and $i+1$ on the left/right side of the edge.

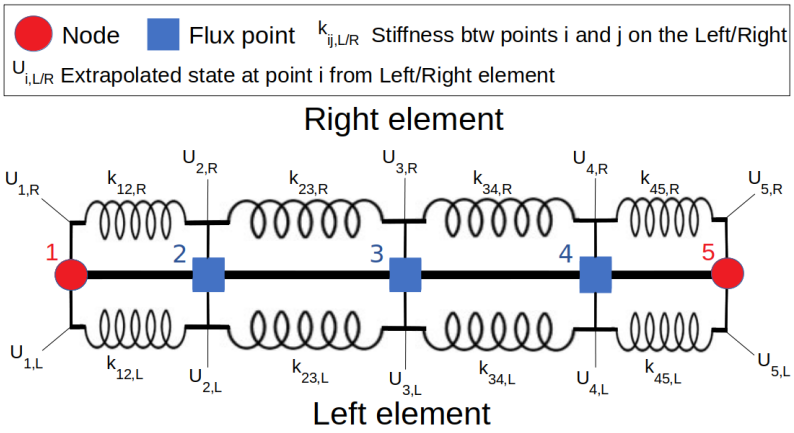


Figure 7: Sub-cell order-dependent spring analogy, edge linking node 1 and 2 (red dots) with 3 flux points (blue squares) for a P2 reconstruction

The main benefits of this method are its sub-cell resolution and its order dependency. Indeed, it tends to increase the flexibility of the mesh as the order of the polynomial extrapolation grows (because the number of spring contributions increases with the order). This is very important as we would like to use high-order polynomial reconstructions (to improve the accuracy per DOF) with coarse grids (to decrease the computational cost).

5.2.2 Connectivity information: Q2 meshes

When studying flows over curved bodies (such as cylinders), Q1 meshes produce unsatisfying solutions. This is due to the discrepancy between the physical boundary which is curved and its numerical description which is straight. In addition, as already reported in [36, 37], if the order of the numerical method increases and becomes highly different from the order of the geometrical approximation of the boundary (P4Q1 for instance), the quality of the physical results decreases significantly. Within this work, we will mainly focus our analysis on Q2 meshes. Consequently, connectivity information for the sub-cell order-dependent spring analogy should be established. Fig.8(a) shows a simplified mesh of 3 high-order triangular mesh and a P3 reconstruction. We introduce as well the following nomenclature to clarify our subsequent explanations: the 3 red nodes at each vertex of the triangle are called corner nodes. Each node located at the middle of each edge is called Middle node. Let us specify that this classification and numbering is only fictitious as of now. Indeed, these nodes are all COOLFluiD geometric entities and possess the same characteristics.

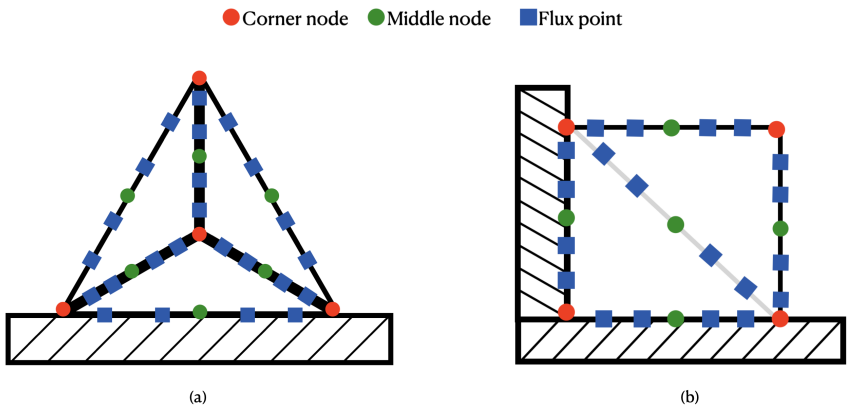


Figure 8: Location of the nodes and of the flux points for P3Q2 (a) unstructured elements attached to one boundary (b) structured elements attached to two boundaries.

In order to be able to place the subsprings between two consecutive entities, a sorting algorithm is applied in order to determine how these flux points are oriented with respect to the two edge-connected nodes. To summarize: over the edges the number of subsprings is:

- P+2 : for Q2 meshes and if the FR polynomial order is even,
- P+3 : for Q2 meshes and if the FR polynomial order is odd.

Fig.8(b) depicts a special case of edge connected nodes (i.e. gray edge) where they are both boundary nodes, yet, they are placed on different boundaries. Consequently, the gray edge linking both nodes should be dealt with as an inner edge and not a boundary edge.
Concerning the boundary treatments of the middle nodes, we distinguish 2 cases:

1. For Q2 meshes on straight lines, as shown in Fig.9(b), we deal with the middle node as it was a regular boundary node. Nevertheless, all boundary points are subject to a factor that multiplies its resulting spring stiffness, in order to rigid its movements.
2. For Q2 meshes on curved lines, as shown Fig.9(a), we propose an ad-hoc solution. As the sub-cell resolution is also present on a curved boundary, we approximate the second order line between two consecutive points as a straight line and we deal with as piecewise linear boundary subsprings.

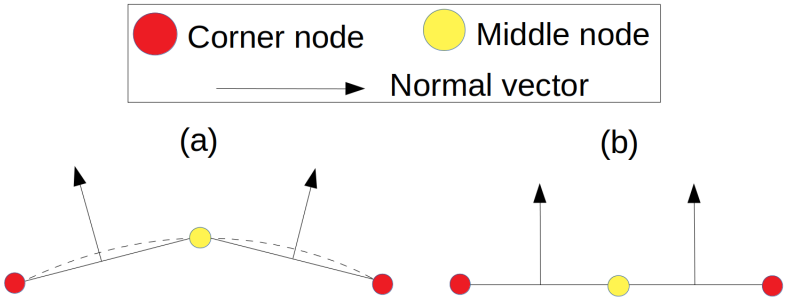


Figure 9: Simplified illustration of the Q2 boundary edge: Q2 straight line (b), Q2 curved line (a)

6 Numerical results

6.1 Inviscid flow in a channel with a bump

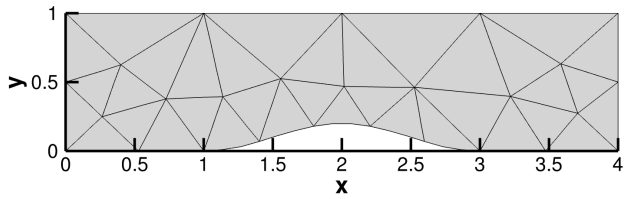
The first test case consists of an inviscid flow at $M = 0.5$ through a channel with a smooth sinusoid bump. The physical domain is bounded by the inlet at $x = 0$, the outlet at $x = 4$ and the bottom and top edges at $y = 0$ and $y = 1$ respectively. While the top edge is straight, the bottom one is characterized by a smooth bump with an axis of symmetry positioned at $x = 2$, as illustrated in Fig.10a. The aforementioned sinusoid bump is defined by the function:

$$f(x) = 0.1 + 0.1 \cdot \cos(\pi(x - 2)), \text{ with } x \in [1; 3]. \tag{39}$$

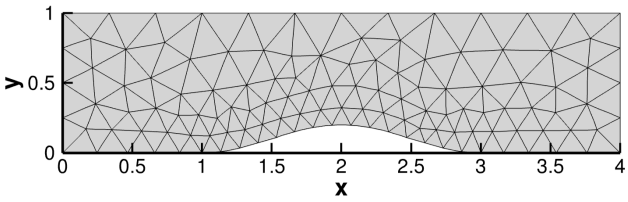
Regarding the boundary conditions, subsonic inlet and subsonic outlet conditions are imposed, as well as a slip-wall boundary condition on both top and bottom edges. The subsonic inlet condition implied fixing the total inlet pressure and temperature, while the subsonic outlet condition requires imposing the outlet pressure. The free-stream flow values for this test case are summarized in Tab.1.

M_∞	ρ_∞[-]	p_∞[-]	$v_{x,\infty}$[-]	$v_{y,\infty}$[-]
0.5	1	1	$0.5\sqrt{1.4}$	0

Table 1: Sinusoid bump: free-stream flow characteristics



(a) Coarse mesh: 35 elements (grid 1)



(b) Fine mesh: 203 elements (grid 2)

Figure 10: Discretization of the channel with triangular cells

As illustrated in Fig.10, the physical domain is discretized by means of two different grids formed by triangular elements. For the finest mesh, illustrated in Fig.10b, the number of elements on the boundaries was fixed to 4 on the inlet and outlet edges, 12 elements on the top and 25 elements on the bottom edge. Both unstructured grids were generated using *Gmsh* [38], an open-source mesh generator. Both straight edged elements with linear geometric mapping ($Q1$), as well as curvilinear elements with quadratic geometric mapping ($Q2$) were used. With $Q1$ straight-edged elements, the physical boundary may not coincide with the computational boundary leading to errors polluting the solution. Therefore most of the presented simulations are run with curvilinear triangular elements. In the present work, the elements are denoted by $PpQq$ following the convention of Bassi and Rebay [39], where p refers to the order of the polynomials used to approximate the solution and q refers to the order of the polynomials used for the geometric mapping.

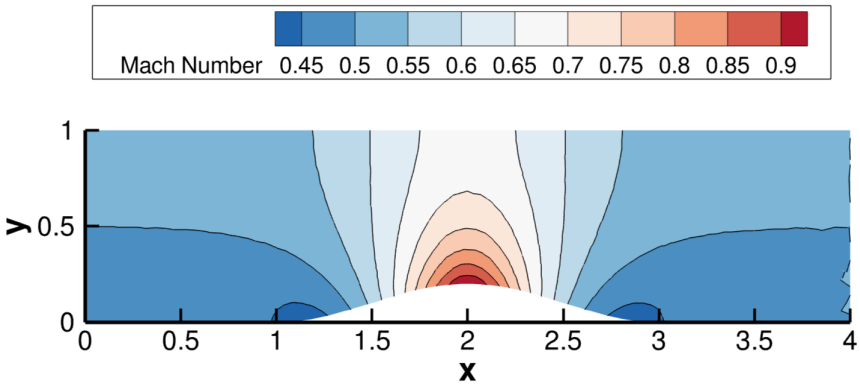


Figure 11: Mach number contours for the inviscid flow through a channel with a bump obtained with P3Q2 elements (grid 2)

The Mach contour plot obtained with P3Q2 elements (using grid in Fig.10b) is presented in Fig.11. The illustrated result was obtained using implicit time stepping method which greatly speeds up the convergence. Indeed, since there is no constraint on the CFL number, convergence was reached in 76 iterations for a total run time of 59.0447 seconds. Fig.12 shows the logarithm of the density residual and the CFL number evolution as a function of the simulation iterations. The CFL law used is the following:

$$\mathrm{CFL}(n) = \min(0.5 \cdot 2^n,\ 10^4)\ , \tag{40}$$

where n is the number of iterations. If an explicit time stepping method was to be used, the CFL number has to be limited to guarantee the stability of the scheme: $\text{CFL}_{max} = \frac{1}{P+1}$.

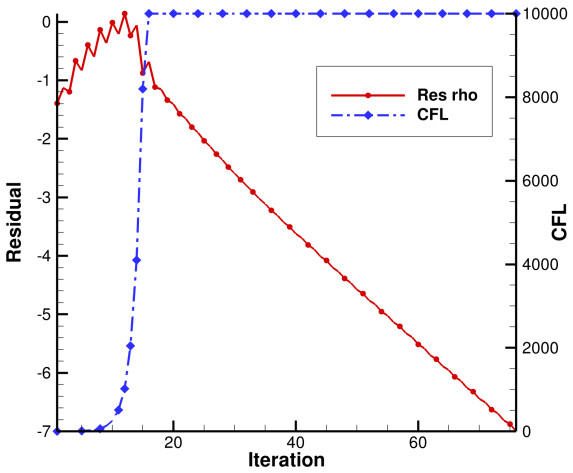


Figure 12: Logarithm of the residual of the density and CFL evolution for the inviscid flow through a channel with a bump obtained with P3Q2 elements (grid 2)

In order to verify the obtained results, a comparison with data produced with the already implemented and verified FR solver working with quadrilateral elements [4] was performed. Fig.13 illustrates the density measured at the bottom wall extracted from the results obtained with P3Q2 triangular elements opposed to P5Q2 quadrilateral elements. The total number of degrees of freedom in the triangles case is equal to 2030, while in the quadrilateral elements case the number of degrees of freedom is 3600. As illustrated, P3Q2 triangular elements produced satisfactory results as it follows well the reference data.

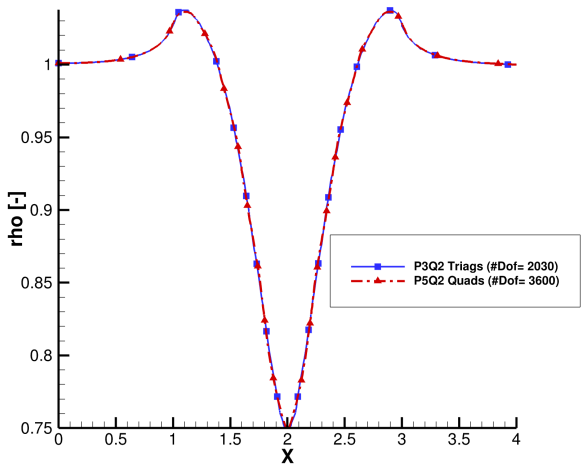


Figure 13: Wall density obtained with P3Q2 elements compared to reference results obtained with P5Q2 quadrilateral elements

For an inviscid flow, no entropy should be produced within the flow field and thus, should be constant on the entire domain. Therefore, the accuracy of the solution will be measured through the entropy error

ϵ_s, where ϵ_s in the i-th state is defined as:

$$\epsilon_{s,i} = \frac{s - s_i}{s}. \tag{41}$$

The value of the entropy s is determined using the free-stream flow characteristics. The L_2-norm of the entropy error ϵ_s is defined as follows:

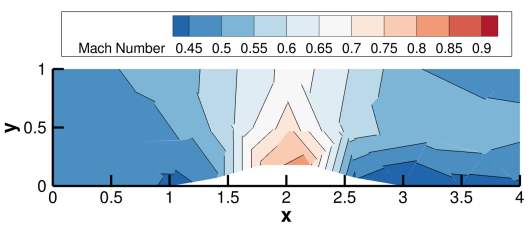
$$L_2(\epsilon_s) = \sqrt{\sum_{n=1}^{N} \int_{\Omega_n} \epsilon_s^2 d\Omega_n} \approx \sqrt{\sum_{n=1}^{N} \sum_{j=1}^{N_q} \hat{\epsilon_s}^2 J_n \omega_j}, \tag{42}$$

where N is the number of elements, Ω_n is the domain associated with the n-th element, J_n the Jacobian determinant of the n-th element, N_q the number of quadrature points and ω_j the j-quadrature weight. As shown in Eq.42, the integral over the element domain is computed using Gauss quadrature as in [22].

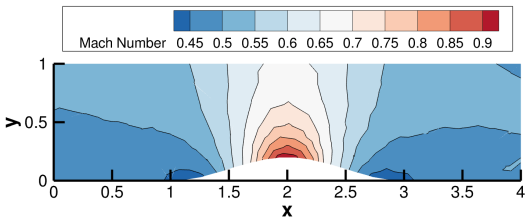
Order	**P1**			**P2**			**P3**		
Dof	105	303	795	210	606	1590	350	1010	2650
L_2 **error**	9.07E-03	3.35E-03	1.15E-03	1.31E-03	3.35E-04	1.08E-04	7.35E-04	1.31E-04	3.32E-05
Order	**P4**			**P5**					
Dof	525	1515	3975	735	2121	5565			
L_2 **error**	3.43E-04	4.43E-05	1.30E-05	1.84E-04	2.68E-05	7.16E-06			

Table 2: Results for the entropy L_2-error for the inviscid flow through a channel with a bump

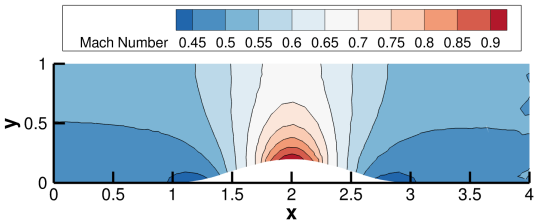
The L_2-norm of the entropy error has been computed for three successively refined grids starting with grid 1 (Fig.10a), with P1Q2, P2Q2, P3Q2, P4Q2 and P5Q2 FR schemes. For all the simulations the parameters were fixed, this includes the VCJH parameter c, for which the same working value was used. As illustrated in Tab.2, the accuracy is improved when either the polynomial order P approximating the solution within an element is increased or for the same element size and number (i.e. p-refinement), or when the mesh is refined for the same order (i.e. h-refinement). Tab.2 illustrates also the advantage of p-refinement as the L_2-error of P1Q2 elements on the finest grid are approximately reduced by a factor of 10 when using P5Q2 elements on the coarsest grid even though the total number of degrees of freedom is slightly higher for P1Q2, namely 795 as opposed to 735 for P5Q2.



(a) P1Q2



(b) P3Q2



(c) P5Q2

Figure 14: Mach contours for the inviscid flow through a channel with a smooth bump on the coarse mesh (grid 1)

6.2 Viscous flow around a cylinder Re=40

In this test case, the aim is to simulate the stable re-circulation bubble formed behind a cylinder occurring at a specific Reynolds number. In fact, The viscous flow around cylinders is greatly influenced by the Reynolds number:

$$\mathrm{Re} = \frac{\rho v_\infty D}{\mu} \tag{43}$$

with v_∞ the far-field speed, D the diameter of the cylinder and μ the dynamic viscosity. For different Re-numbers, different flow regimes can be defined:

- For a very low Re-number, e.g. Re < 1, the flow is symmetrical and no separation occurs
- For $1 <$ Re < 40 a stable re-circulation bubble is formed behind the cylinder holding two steady vortices at its trailing edge.
- For $40 <$ Re $< 10^3$ vortices are shed behind the cylinder and a von Karman vortex street is formed [40].

In order to simulate a steady-state solution and visualize stable vortices, the Re-number is chosen equal to 40. The standard value for air was taken for the Prandtl number (i.e. Pr= 0.72). Pr is defined as

$$\mathrm{Pr} = \frac{\mu c_p}{\kappa} \tag{44}$$

where κ is the Fourier heat conduction coefficient and c_p the specific heat capacity. The free-stream values for this second test case are compiled in Tab.3.

M_∞	ρ_∞[-]	p_∞[-]	$v_{x,\infty}$[-]	$v_{y,\infty}$[-]
0.15	1	1	$0.15\sqrt{1.4}$	0

Table 3: Viscous flow around a cylinder: free-stream flow characteristics

Regarding the geometry, the cylinder has a diameter of $D = 1$ while the outer boundary is characterized by diameter $d_{outer} = 30$. An adiabatic no-slip wall is applied to the cylinder's boundary. Moreover, a far-field boundary condition is associated with the outer boundaries of the physical domain. Thanks to the symmetry of the present case, only the upper half of the domain was considered and subdivided into 263 triangular elements as depicted in Fig.15. The CFL law used in this case is the same as the one used for the bump.

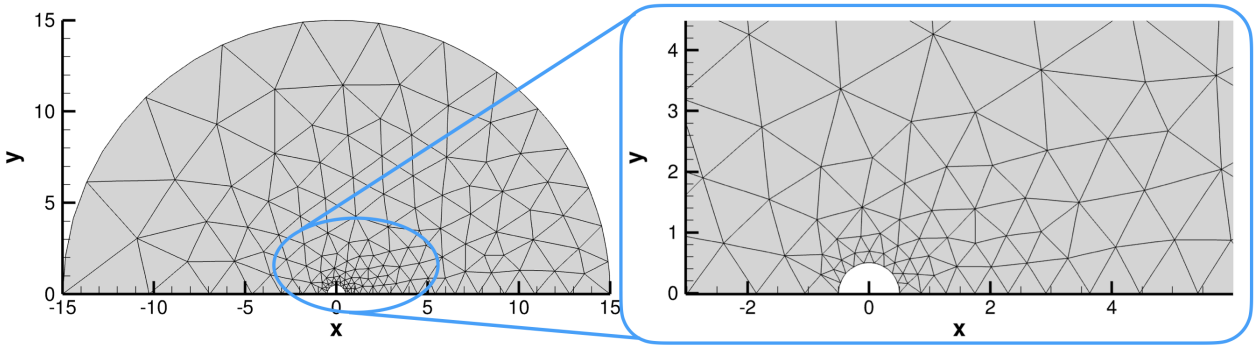


Figure 15: Discretization of the domain for the viscous flow around a cylinder case ($N_{elements} = 263$)

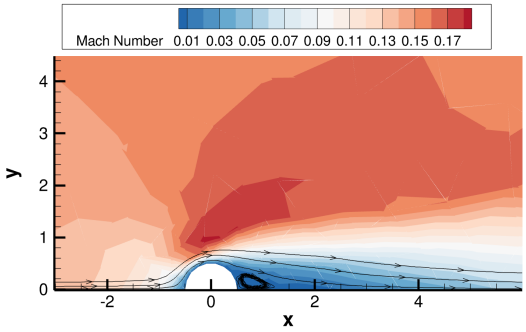


Figure 16: P1Q2 (#dof=789)

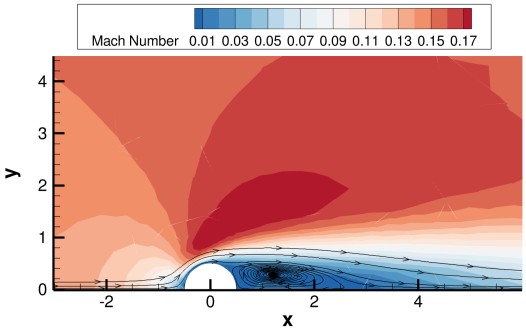


Figure 17: P2Q2 (#dof=1578)

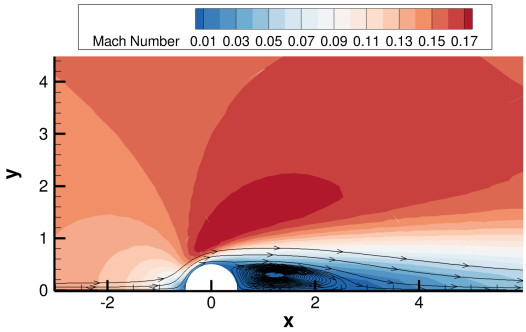


Figure 18: P3Q2 (#dof=2630)

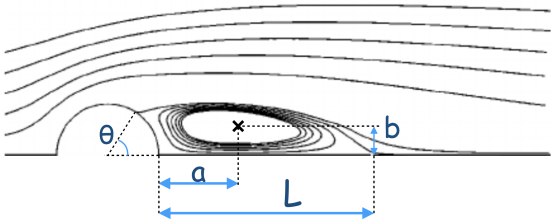


Figure 19: Flow characteristics for Re = 40 and definition of vortex centers position, length of the recirculation zone and separation angle

	a/D	b/D	L/D	θ
Coutanceau [41] & Bouard (experimental)	0.76	0.29	2.13	53°
Xu [42]	0.72	0.3	2.24	-
Lepilliez et al. [43]	0.707	0.297	2.220	-
Wang et al. [44]	0.75	0.3	2.29	52.1°
Xu et al. [42]	-	-	2.21	53.5°
Present results				
P3	0.7	0.29	2.23	53.31°
P2	0.704	0.27	2.16	50.32°

Table 4: Viscous flow over a cylinder for Re = 40: length L of recirculation zone, location (a, b) of vortex center and separation angle.

Table 4 shows the normalized center positions and lengths of the recirculation zone as well as the separation angles θ for several references and our present results (i.e. P2 and P3). Mainly, the error on the latter, with respect to the experimental value, measured by Coutanceau and Bouard and compared to the P3 solution, is of the order of 0.58%, whereas the errors on the center coordinates are (7.89%, 0%). In general, our obtained values are in good agreement with the other reference simulations.

6.3 Supersonic flow in a channel with a wedge

Multiple results of the sub-cell order-dependent spring analogy are obtained for the steady 2D wedge channel flow based on the FR-AMR sub-cell order-depent spring analogy for triangular grids. The test case conditions are presented in Tab.5 and Tab.6, whilst, the computational domain is shown in Fig.20.

Physical Model	M	ρ [-]	ρu [-]	ρv [-]	ρE [-]
Perfect gas	2	1	2.36643	0	5.3

Table 5: 2D wedge – Flow characteristics

Dimensions	Type	BC 1	BC 2	BC 3	BC 4
2D	Triangular	Inlet	Outlet	Symmetry	no-slip wall

Table 6: 2D wedge – Computational domain

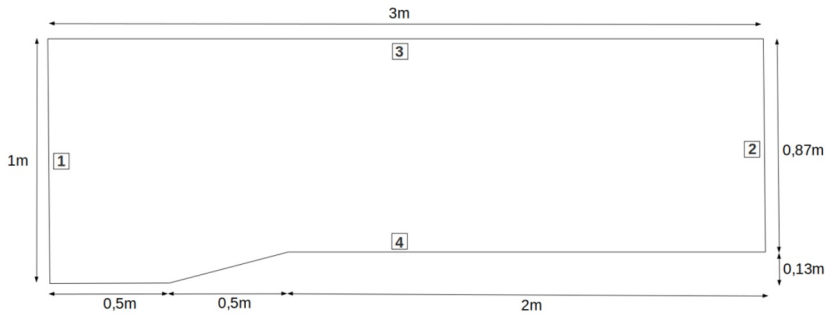
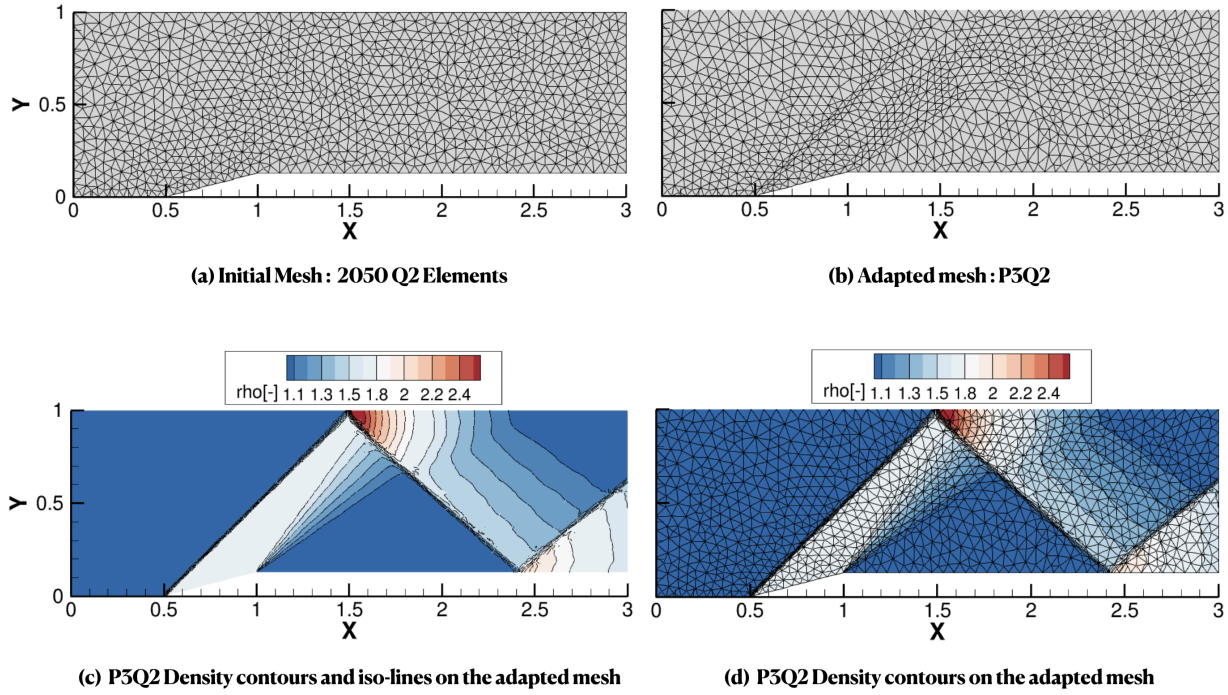


Figure 20: 2D wedge geometry

Three meshes are used to demonstrate the capabilities of the FR-AMR sub-cell order-dependent spring analogy. The results over the meshes, presented in Tab.7, are in agreement with the results in F. Ben Ameur et al. [8]. In fact, for all configurations, the mesh adapts itself properly to the phenomena encountered in the wedge channel. We can indeed see that the mesh is refined around the shocks, the expansion waves and their reflections. The results of the simulations carried out on the meshes (cf. Tab.7) are depicted in Fig.21, Fig.22 and Fig.23.

Q2 meshes used for	**# Elements**	**Coarsening # Elements**	**# DOF**	**Coarsening # DOF**
P3	2050	×1	20500	×1
P4	344	×6	5655	×3.62
P5	176	×11.6	3696	×5.55

Table 7: Mesh characteristics



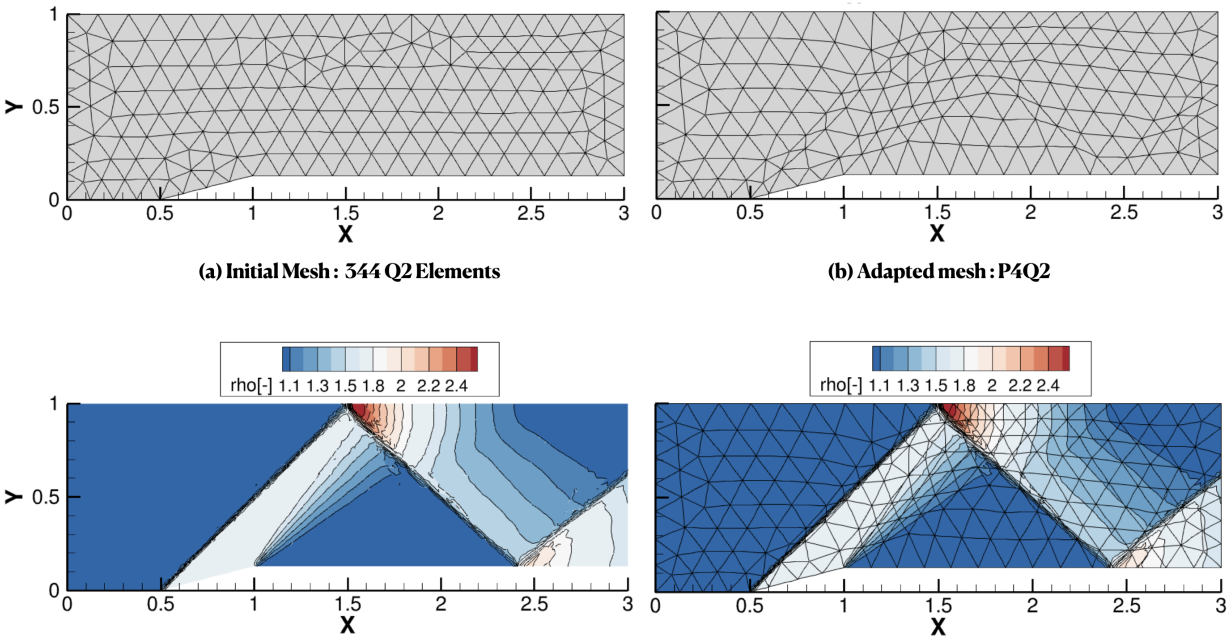
(a) Initial Mesh : 2050 Q2 Elements

(b) Adapted mesh : P3Q2

(c) P3Q2 Density contours and iso-lines on the adapted mesh

(d) P3Q2 Density contours on the adapted mesh

Figure 21: FR-AMR P3Q2 results



(a) Initial Mesh : 344 Q2 Elements

(b) Adapted mesh : P4Q2

(c) P4Q2 Density contours and iso-lines on the adapted mesh

(d) P4Q2 Density contours on the adapted mesh

Figure 22: FR-AMR P4Q2 results

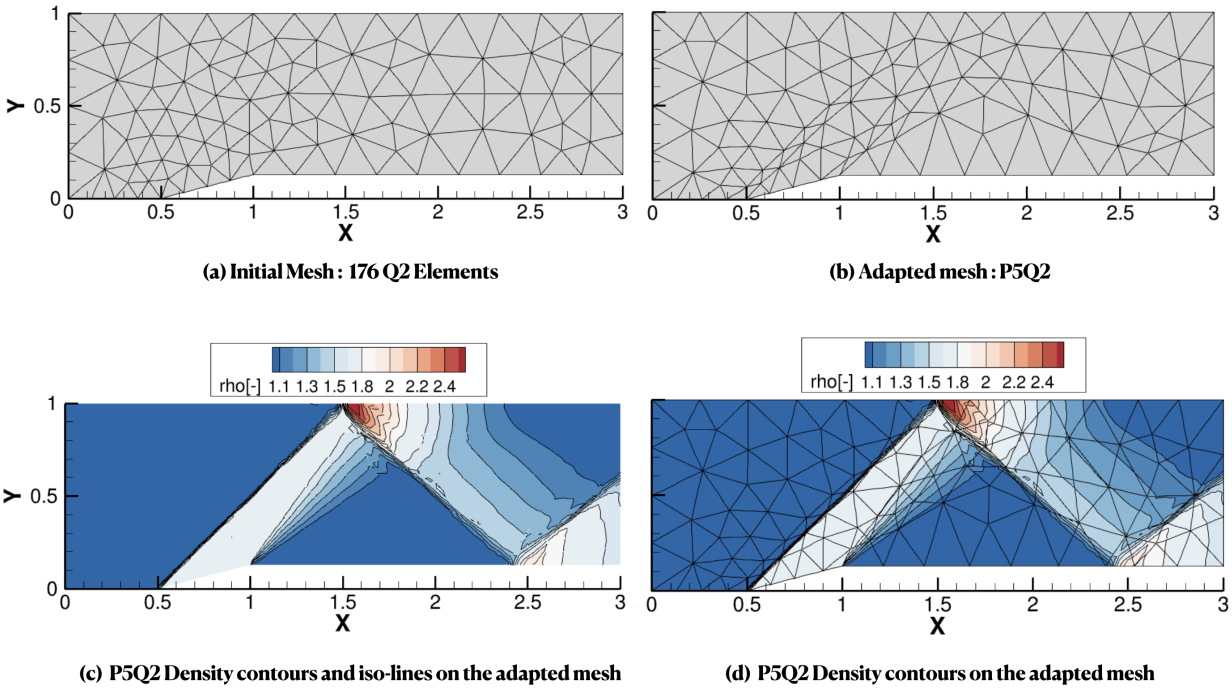


Figure 23: FR-AMR P5Q2 results

To further asses the capabilities of the unstructured FR-AMR solver, the shock position is compared to the reference, as studied in F. Ben Ameur et al. in [8].

The results obtained in Fig.24 show a considerable advantage brought by the AMR procedure: the shock (which is characterized by the density jump) is close to the reference solution, i.e. $x_{ref} = 1.0039$ and $\frac{\rho_2}{\rho_1} = 1.7047$. It comes with no surprises that P3 density field at the line section Y=0.5, is the closest to the analytical solution, as it contains a large number of DOFs. Nevertheless, the P4 and P5 solutions depicts excellent outcomes, considering that we are using $\times 6$ and $\times 11.6$ coarser meshes, respectively.

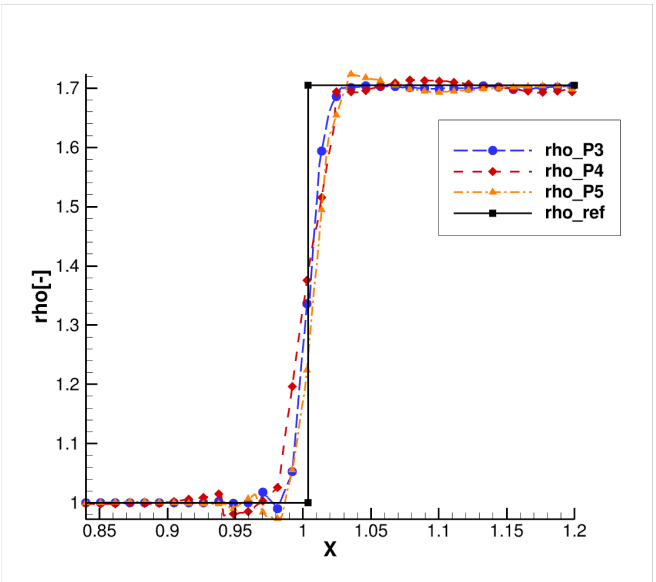


Figure 24: Comparison of the density fields obtained around the oblique shock for P3, P4 and P5 with respect to the analytical solution at Y=0.5m

6.4 Inviscid hypersonic flow around a cylinder

Promising preliminary results are obtained for the case of a Mach 8 inviscid flow around a cylinder with a radius of 1.5. The test case is well known in the literature as it is used to investigate the carbuncle phenomenon of the various upwind schemes [45, 46, 47, 48]. Fig.25 shows the pressure contours of the adapted mesh. We can clearly see that the bow shock is narrowly captured while using a relatively coarse mesh with 776 elements. This number is quite low for hypersonic simulations, even when using a third order polynomial (i.e. P3). The numerical instability associated with shock-induced anomalies are limited thanks to r-AMR.

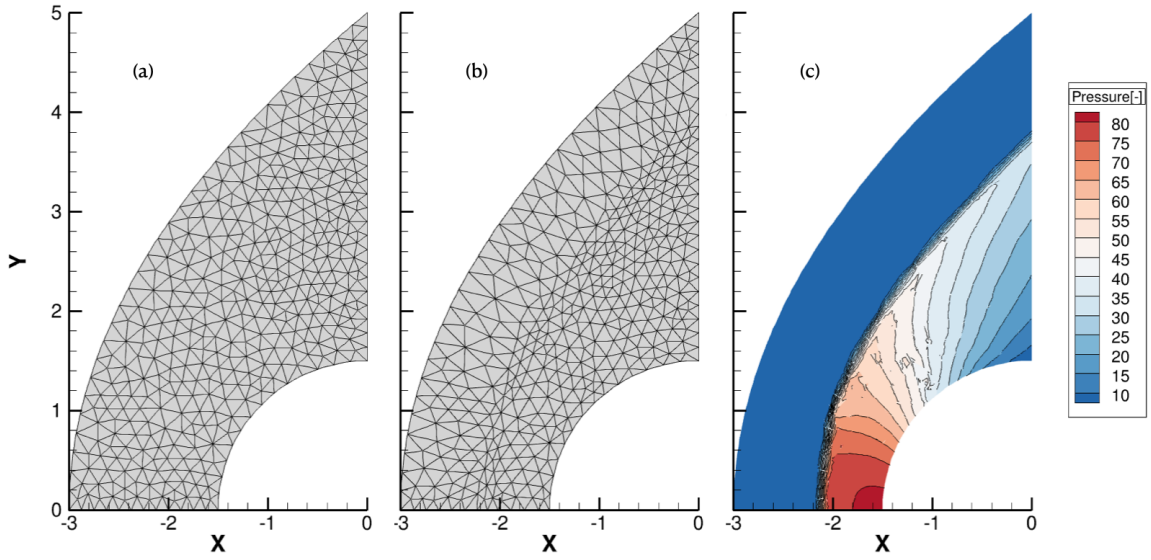


Figure 25: (a) Initial Mesh - (b) Adapted Mesh - (c) Pressure field and contours lines

7 Conclusion and Future Work

Within this work, the authors have successfully extended the COOLFluiD's existing FR solver to deal with triangular elements. The new solver can handles explicit/implicit and inviscid/viscous governed by the Euler or Stokes equations. Basically, this extension will mainly reduce the tedious mesh generation task, especially, for complex geometries as the solver can rely now on simplex elements. In addition, the new concept of sub-cell order-dependent spring analogy shows its promising results on this type of mesh elements. As our previous work concerning r-AMR, the FR-AMR is developed as a standalone module that can be coupled, with minor efforts, to any high-order CFD/plasmas solvers. Finally, this work contributed to the development of the first in its kind spring-based r-AMR within the high-order FR framework for triangular meshes.

8 Acknowledgments

The research of Firas Ben Ameur is supported by SB PhD fellowship 1SC3519N of the FWO.

References

[1] Peter E. Vincent, Freddie D. Witherden, Antony M. Farrington, George Ntemos, Brian C. Vermeire, Jin S. Park, and Arvind S. Iyer. Pyfr: Next-generation high-order computational fluid dynamics on many-core hardware (invited). *22nd AIAA Computational Fluid Dynamics Conference*, 2015.

[2] Freddie D. Witherden, Brian C. Vermeire, and Peter E. Vincent. Heterogeneous computing on mixed unstructured grids with pyfr. *Computers & Fluids*, 120:173–186, 2015.

[3] R. Abgrall and Mario Ricchiuto. High-order methods for cfd. *Encyclopedia of Computational Mechanics Second Edition*, page 1–54, 2017.

[4] R. Vandenhoeck and A. Lani. Implicit high-order flux reconstruction solver for high-speed compressible flows. 242:1–24.

[5] Ray Vandenhoeck, Andrea Lani, and Johan Steelant. Development of an implicit high-order flux reconstruction solver for the langtry-menter laminar-turbulent transition rans model. *Computer Physics Communications*, 278:108408, 2022.

[6] F. Ben Ameur, R Vandenhoeck, and Lani. A. *High-Order Flux Reconstruction Scheme for Thermo-Chemical Nonequilibrium High-Speed Flows*.

[7] Firas Ben Ameur and Andrea Lani. r-adaptive algorithms for high-speed flows and plasma simulations. *Computer Physics Communications*, 261:107700, 2021.

[8] Firas Ben Ameur, Joachim Balis, Ray Vandenhoeck, Andrea Lani, and Stefaan Poedts. r-adaptive algorithms for supersonic flows with high-order flux reconstruction methods. *Computer Physics Communications*, 276:108373, 2022.

[9] Dries Kimpe, Andrea Lani, Tiago Quintino, Stefaan Poedts, and Stefan Vandewalle. The coolfluid parallel architecture. In Beniamino Di Martino, Dieter Kranzlmüller, and Jack Dongarra, editors, *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pages 520–527, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[10] Andrea Lani, Nadege Villedie, Khalil Bensassi, Lilla Koloszar, Martin Vymazal, Sarp M Yalim, and Marco Panesi. Coolfluid: an open computational platform for multi-physics simulation and research. In *21st AIAA Computational Fluid Dynamics Conference*, page 2589, 2013.

[11] H. T. Huynh. *High-Order Methods Including Discontinuous Galerkin by Reconstructions on Triangular Meshes*. 01 2011.

[12] D. Williams, P. Castonguay, P. Vincent, and A. Jameson. An extension of energy stable flux reconstruction to unsteady, non-linear, viscous problems on mixed grids. In *20th AIAA Computational Fluid Dynamics Conference*.

[13] P. Castonguay, P. E. Vincent, and A. Jameson. Application of high-order energy stable flux reconstruction schemes to the euler equations. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics, 01 2011.

[14] P. Castonguay, P. E. Vincent, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes for triangular elements. 51(1):224–256.

[15] D.M. Williams, P. Castonguay, P.E. Vincent, and A. Jameson. Energy stable flux reconstruction schemes for advection–diffusion problems on triangles. *Journal of Computational Physics*, 250:53–76, 2013.

[16] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer-Verlag.

[17] B. Cockburn and CW. Shu. The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis*, 35(6):2440–2463, 12 1998.

[18] J. Peraire and P.-O. Persson. The compact discontinuous galerkin (cdg) method for elliptic problems. *SIAM Journal on Scientific Computing*, 30:1806–1824, 02 2007.

[19] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760, 08 1982.

[20] P. A. Raviart and J. M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In I. Galligani and E. Magenes, editors, *Mathematical Aspects of Finite Element Methods*, pages 292–315. Springer Berlin Heidelberg, 1977.

[21] P. Davis and P. Rabinowitz. Abscissas and weights for gaussian quadratures of high order. 56:35, 01 1956.

[22] D.M. Williams, L. Shunn, and A. Jameson. Symmetric quadrature rules for simplexes based on sphere close packed lattice arrangements. *Journal of Computational and Applied Mathematics*, 266:18–38, 2014.

[23] P. E. Vincent, P. Castonguay, and A. Jameson. A new class of high-order energy stable flux reconstruction schemes. 47(1):50–72, 04 2011.

[24] M. Dubiner. Spectral methods on triangles and other domains. 6(4):345–390, 12 1991.

[25] David M Williams. *Energy stable high-order methods for simulating unsteady, viscous, compressible flows on unstructured grids*. Stanford University, 2013.

[26] B. van Leer, J. Thomas, P. Roe, and R. Newsome. A comparison of numerical flux formulas for the euler and navier-stokes equations. 06 1987.

[27] V.V. Rusanov. The calculation of the interaction of non-stationary shock waves and obstacles. *USSR Computational Mathematics and Mathematical Physics*, 1(2):304–320, 1962.

[28] P.L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372, 10 1981.

[29] M.-S. Liou. A sequel to ausm: Ausm+. *Journal of Computational Physics*, 129(2):364–382, 12 1996.

[30] M.-S. Liou. A sequel to ausm, part ii: Ausm+-up for all speeds. *Journal of Computational Physics*, 214(1):137–170, 05 2006.

[31] Shrirang Abhyankar, Jed Brown, Emil M Constantinescu, Debojyoti Ghosh, Barry F Smith, and Hong Zhang. Petsc/ts: A modern scalable ode/dae solver library. *arXiv preprint arXiv:1806.01437*, 2018.

[32] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2018.

[33] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Dave A. May, Lois Curfman McInnes, Richard Tran Mills, Todd Munson, Karl Rupp, Patrick Sanan, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.9, Argonne National Laboratory, 2018.

[34] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[35] R. Jain and I. Chlamtac. *The P2 Algorithm for Dynamic Calculation of Quantities and Histograms Without Storing Observations*. Eastern Research Laboratory, Digital Equipment Corporation, 1985.

[36] Lilia Krivodonova and Marsha Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. *Journal of Computational Physics*, 211(2):492–512, 2006.

[37] Xiaojuan Luo, Mark Shephard, and Jean-François Remacle. Influence of geometric approximation on the accuracy of higher order methods. 01 2001.

[38] 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.

[39] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2d euler equations. *Journal of Computational Physics*, 138(2):251–285, 1997.

[40] M. S. Bloor. The transition to turbulence in the wake of a circular cylinder. *Journal of Fluid Mechanics*, 19(2):290–304, 1964.

[41] Madeleine Coutanceau and Roger Bouard. Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 1. steady flow. *Journal of Fluid Mechanics*, 79(2):231–256, 1977.

[42] Sheng Xu. The immersed interface method for simulating prescribed motion of rigid objects in an incompressible viscous flow. *Journal of Computational Physics*, 227(10):5045–5071, 2008.

[43] Mathieu Lepilliez, Elena Popescu, Frederic Gibou, and Sébastien Tanguy. On two-phase flow solvers in irregular domains with contact line. *Journal of Computational Physics*, 321, 06 2016.

[44] Li Wang, Gaetano M.D. Currao, Feng Han, Andrew J. Neely, John Young, and Fang-Bao Tian. An immersed boundary method for fluid–structure interaction with compressible multiphase flows. *Journal*

of Computational Physics, 346:131–151, 2017.

[45] Sutthisak Phongthanapanich. Healing of the carbuncle phenomenon for ausmdv scheme on triangular grids. *International Journal of Nonlinear Sciences and Numerical Simulation*, 17(1):15–28, 2016.

[46] S Phongthanapanich. An accurate and robust ausm-family scheme on two-dimensional triangular grids. *Shock Waves*, 29(5):755–768, 2019.

[47] Fan Zhang, Jun Liu, Biaosong Chen, and Wanxie Zhong. A robust low-dissipation ausm-family scheme for numerical shock stability on unstructured grids. *International Journal for Numerical Methods in Fluids*, 84(3):135–151, 2017.

[48] Fan Zhang, Jun Liu, and Biaosong Chen. Modified multi-dimensional limiting process with enhanced shock stability on unstructured grids. *Computers & Fluids*, 161:171–188, 2018.