Asociación Argentina
de Mecánica Computacional

Mecánica Computacional Vol XXXII, págs. 1213-1237 (artículo completo)
Carlos G. García Garino, Aníbal E. Mirasso, Mario A. Storti, Miguel E. Tornello (Eds.)
Mendoza, Argentina, 19-22 Noviembre 2013

# UNSTRUCTURED ANISOTROPIC MESH ADAPTATION FOR COMPRESSIBLE FLOWS USING REFINEMENT AND COARSENING PROCEDURES

**Renato V. Linn**[a] **and Armando M. Awruch**[b]

[a]*Graduate Student PPGEC/UFRGS renatolinn@gmail.com*

[b]*Graduate Program in Civil Engineering (PPGEC), Federal University of Rio Grande do Sul (UFRGS), Av. Osvaldo Aranha, 99, 3o andar, 90035-190, Porto Alegre, RS, Brazil, amawruch@ufrgs.br*

**Keywords:** Computational Fluid Dynamics (CFD), Anisotropic Mesh Adaptation, Finite Element Method.

**Abstract.** An anisotropic mesh adaptation procedure for unstructured meshes and its coupling with a compressible flow simulation code with large Mach range using the Finite Element Method (Bubnov-Galerkin) is presented in this work. The Characteristic Based Split method (CBS) is used for the compressible flow simulation, using linear triangular and linear tetrahedral finite elements for two and three dimensional cases, respectively. An anisotropic mesh adaptation technique which uses a tensor error estimation based on a Riemann metric, including mesh refinement and mesh coarsening procedures is implemented for accurate and computationally cheaper solutions evaluation. Anisotropic meshes are obtained without any mesh moving technique. Two-dimensional and three-dimensional analyses are performed in order to validate the proposed methodology and additional large and complex tests are also included.

## 1 INTRODUCTION

An adaptation strategy should be concerned with maximizing the resolution at a minimal computational cost. For numerical simulations, using the Finite Element Method (FEM), the main procedures for mesh adaptation are mesh refinement, mesh coarsening, node movement, edge and face swapping. Other alternative is the full or local remeshing, repeated many times using a classical or modified Delaunay triangulation. The computational cost is drastically reduced for an anisotropic adaptation case. Metric-based methodologies for anisotropic adaptation use the concept of Riemannian space to analyse the error as a tensorial quantity and they are the most popular in mesh adaptation formulations. The use of the Riemannian space allows the possibility to implement several improvements to the anisotropic adaptation, including important specific considerations for CFD simulations like a continuous bondary layer (Loiseille and Löhner, 2009), levelset functions for moving surfaces (Alexandra and Ducrot, 2009), anisotropic mesh generation (Loiseille et al., 2011) and time-dependent problems with moving boundaries (Alauzet and Oliver, 2011). Works of Castro-Díaz et al. (1997); Habashi et al. (2000) are also very general strategies with applications to CFD problems.

The mesh refinement can be performed in regular or irregular configurations. The irregular configuration is used in the present work in order to consider all anisotropic cases for refinement (Cougny and Shephard, 1999) with the edge split performed at the Riemannian midpoint. Mesh coarsening is done by edge contraction (Ollivier-Gooch, 2003). Edge collapse is usually considered as a discrete operation, collapsing the edge directly from one end point to the other or by considering some more discrete possibilities like the midpoint or barycentre (Walter et al., 2005). We propose a continuous edge collapse in the present work where the point of collapse is the optimal in an anisotropic sense. This is performed using a Sequential Quadratic Programming algorithm (SQP) to solve an optimization problem which takes into account the anisotropic quality of the neighbourhood of the collapsed edge. The edge swapping and face swapping procedures are also covered (Freitag and Ollivier-Gooch, 1997). The same optimized strategy is used to guide the collapse procedure, improving the anisotropy.

Other important subject covered by this work is the dimensional independence and the fact that only simple mesh information is needed. The identification of type of edge or surface and other complex geometrical information are not required at any stage of the proposed strategy.

The adaptation procedure is analysed for an analytical field and for compressible flow simulation. The simulation is performed using a Characteristic-Based Scheme (CBS) algorithm using linear triangular and linear tetrahedral finite elements for two and three dimensional cases, respectively.

## 2 MESH TOPOLOGY

To ensure generality of the adaptation procedures for any dimensional case, the mesh is described as a simplex in this section. A $k-simplex$ $(0 \leq k \leq d)$ is the convex enclosure of the $k+1$ independent nodes in $\mathbb{R}^d$ called vertices. Geometrically, a $0-simplex$ is a point, a $1-simplex$ is a line segment, a $2-simplex$ is a triangle, a $3-simplex$ is a tetrahedron and so on. The set of $k+1$ vertices $v$ of a $k-simplex$ $K$ is defined as $V_K$

$$V_K = \{v_0, ..., v_k\} \tag{1}$$

The set of $3k-3$ edge vectors $a$ of a $k-simplex$ $K$ is defined as $A_K$

$$A_K = \{a_0, ..., a_{3k-4}\} \tag{2}$$

The oriented measure $|T_K|$ of the size of a $k-simplex$ $K$ is defined by

$$|T_K| = \frac{\det\left(v_1 - v_0 \quad v_2 - v_0 \quad \ldots \quad v_{k-1} - v_0 \quad v_k - v_0\right)}{k!} \tag{3}$$

A $k-simplex$ $K$ is non-degenerated only if

$$|T_K| > 0 \tag{4}$$

The adaptation procedures modifies the topology $\mathcal{T}$ of the mesh and it must be ensured that topology is not violated anywhere during the whole process.

## 3 METRIC ESTIMATES

Both the direction and the magnitude of the anisotropy are important and necessary information required for the construction of an anisotropic mesh adaptation procedure. These information can be evaluated through the use of the so called metric-based methods, where a Riemannian metric space is evaluated based on the interpolation error of the solution field. In this Riemannian metric space, the error can be monitored as a tensorial quantity (a curved space), enabling direction and magnitude evaluation of the anisotropic field. A more detailed overview of such methodology can be found in Vallet (1992); Loiselle (2008); Frey and Alauzet (2005).

### 3.1 Metric Measures

A metric space $\left(\mathbb{R}^d, \mathcal{M}\right)$ is a finite vectorial space in which the dot product of two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined by:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} = \mathbf{u}^T \mathcal{M} \mathbf{v} \quad \text{for } (\mathbf{u}, \mathbf{v}) \in \mathbb{R}^d \times \mathbb{R}^d \tag{5}$$

where $\mathcal{M}$ is a $d \times d$ symmetric positive definite matrix called metric tensor, or just metric. A Riemannian metric space $\left(\mathcal{M}(\mathbf{x})\right)_{\mathbf{x}\in\Omega}$ is a spatial field that defines at any point $\mathbf{x} \in \Omega$ a metric tensor $\mathcal{M}(\mathbf{x})$ that is variable through the space domain $\Omega$. The case $\mathcal{M} = \mathcal{I}$, in which $\mathcal{I}$ is the $d$- dimensional identity matrix, defines the canonical Euclidian space as a particular case of Riemannian space where the metric is constant in space and isotropic. Thus, the computation of geometric quantities on a Riemannian space requires integration to take into account the variation of the metric field. The length of the edge $\mathbf{ab}$, $\ell_{\mathcal{M}}(\mathbf{ab})$ is continuously evaluated with a parametrization given by $\gamma(t) = \mathbf{a} + t\mathbf{ab}$ with $t \in [0, 1]$ as:

$$\ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 \|\gamma'(t)\|_{\mathcal{M}} dt = \int_0^1 \sqrt{\mathbf{ab}^T \mathcal{M}(\mathbf{a} + t\,\mathbf{ab})\,\mathbf{ab}}\, dt \tag{6}$$

and the volume of a simplex $K$ is evaluated by:

$$|T_K|_{\mathcal{M}} = \int_K \sqrt{\det \mathcal{M}(\mathbf{x})} d\mathbf{x} \tag{7}$$

The central idea of metric-driven mesh adaptation is to generate a mesh in which the edges have length as closer to unity as possible in the prescribed Riemannian metric space. The more closer to unity, better distributed is the error over the mesh. The quality of an element is controlled by a single quality function $Q_{\mathcal{M}}$ of a simplex $K$ that combines both sizing and orientation information:

$$Q_{\mathcal{M}}(K) = \alpha \frac{\sum \ell^2_{\mathcal{M}}(A_K)}{|T_K|_{\mathcal{M}}} \tag{8}$$

where $\alpha$ is a constant that gives the unity value to Eq. 8 for an element equilateral in the Riemannian space. In the previous equation, the numerator takes into account the sizing of the mesh. Decreasing the length of the edges evaluated in the Riemannian space also reduces $Q_{\mathcal{M}}$. The denominator is a measure of the orientation of the simplex in the Riemannian space, where a simplex oriented closer to the local field eigenvectors leads to a reduction of $Q_{\mathcal{M}}$. Thus, minimizing $Q_{\mathcal{M}}$ the quality of the element becomes maximized in an anisotropic sense.

### 3.2 Interpolation Error

The mesh $(\mathcal{N}, \mathcal{T})$ of the bounded domain $\Omega$ defines the $\mathbf{P}^1$ finite element space $\phi \in \mathcal{C}^0\left(\overline{\Omega}\right)$ such that $\phi\,|_{\mathcal{T}}$ is affine for all elements $\mathcal{T} \in (\mathcal{N}, \mathcal{T})$, being $\phi$ the base functions, $\overline{\Omega}$ the element domain and $\mathcal{N}$ the finite set of mesh nodes. For a given continuous function $u$, we denote by $\Pi_{(\mathcal{N},\mathcal{T})}u$ the $\mathbf{P}^1$-interpolant of $u$ on mesh $(\mathcal{N}, \mathcal{T})$. The objective is to minimize the error $u - \Pi_{(\mathcal{N},\mathcal{T})}u$ measured in a $\mathbf{L}^p$ norm. As the exact solution of $u$ is not available, a quadratic approximation is used instead:

$$\forall\, \mathbf{x} \in \Omega, \qquad u\left(\mathbf{x}\right) = \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \tag{9}$$

where $\mathbf{H}$ is a symmetric matrix representing the Hessian of $u$. For given mesh edge $\mathbf{ab}$, the following bound holds for the error $E$:

$$E(\mathbf{ab}) = \left|u - \Pi_{(\mathcal{N},\mathcal{T})}u\right|_{\mathbf{L}^1} \leq c_0\mathbf{ab}^T|\mathbf{H}|\mathbf{ab} \tag{10}$$

where $|\mathbf{H}|$ is the positive-defined Hessian matrix and $c_0$ is a constant independent of $u$. It can be observed that the error can be understood as a measure of the edge length on a Riemannian space, in which metric variation is proportional to $\mathbf{H}$. The estimative of the error on an edge $\mathbf{ab}$ is then defined as the upper bound:

$$E(\mathbf{ab}) \equiv \ell_{\mathcal{M}}\left(\mathbf{ab}\right) \tag{11}$$

As $\mathbf{H}$ is a symmetric matrix, it allows the following decomposition:

$$\mathbf{H} = \mathcal{R}\mathbf{\Lambda}\mathcal{R}^T \tag{12}$$

where $\mathcal{R}$ is an orthonormal matrix satisfying $\mathcal{R}^T\mathcal{R} = \mathcal{R}\mathcal{R}^T = \mathcal{I}_d$ such that $\mathcal{R}$ is composed by the $(\mathbf{v}_i)_{i=1,d}$ eigenvectors associated to $\mathbf{H}$ and $\mathbf{\Lambda}$ is a diagonal matrix composed by the eigenvalues associated to $\mathbf{H}$. To be positive-defined, the eigenvalues $(\lambda_i)_{i=1,d}$ must have strict positive non-null values. To ensure that $\mathbf{H}$ is a positive-defined matrix, the metric field variation is obtained as the Hessian matrix with normalized eigenvalues:

$$\mathcal{M} = |\mathbf{H}| = \mathcal{R}|\mathbf{\Lambda}|\mathcal{R}^T \tag{13}$$

The Hessian matrix is evaluated using a weak formulation based on the Green formula.

### 3.3 Metric Interpolation

The metric field is usually only known discretely at the mesh vertices. The use of metric interpolation enables a continuous metric field consideration over the domain. In Eq. 6, the edge length evaluation must take into account the spatial variation of the metric. The Log-Euclidean interpolation framework (Arsigny et al., 2006) is used here. It works in the logarithm

space in the same way as in the Euclidean space. Consequently, a sequence of $n$ metric tensors can be interpolated in any order while providing a unique metric. Given a sequence of points $(\mathbf{x}_i)_{i=1,k}$ and their respective metrics $\mathcal{M}_i$, then the interpolated metric in $\mathbf{x}$ verifying (Loiseille and Löhner, 2009):

$$\mathbf{x} = \sum_{i=1}^{k} \alpha_i \mathbf{x}_i, \quad \text{with} \quad \sum_{i=1}^{k} \alpha_i = 1 \tag{14}$$

is

$$\mathcal{M} = \exp\left(\sum_{i=1}^{k} \alpha_i \ln\left(\mathcal{M}_i\right)\right) \tag{15}$$

On the space of metric tensors, logarithm and exponential operators are acting on the eigenvalues of the metric directly:

$$\ln\left(\mathcal{M}_i\right) = \mathcal{R}_i^T \ln\left(\mathbf{\Lambda}_i\right) \mathcal{R}_i \quad \text{and} \quad \exp\left(\mathcal{M}_i\right) = \mathcal{R}_i^T \exp\left(\mathbf{\Lambda}_i\right) \mathcal{R}_i \tag{16}$$

An important feature of this procedure is that it preserves anisotropy through the interpolations. The integration of Eq. 6 with this Log-Euclidean spatial variation of the metric becomes, from a discrete point of view:

$$\ell_{\mathcal{M}}\left(\mathbf{ab}\right) \approx \sqrt{\mathbf{ab}^T \mathcal{M}_i \mathbf{ab}} \frac{r-1}{r \ln\left(r\right)} \tag{17}$$

where $r = \left(\sqrt{\mathbf{ab}^T M_i \mathbf{ab}} / \sqrt{\mathbf{ab}^T M_j \mathbf{ab}}\right)$ for an edge with end-points $i$ and $j$. Eq. 17 arises from considering a geometric approximation of the size variation along end-points of the edge: $\forall t \in [0,1]\, \lambda^{-1}\left(t\right) = \left(\lambda_i^{-1}\right)^{1-t} \left(\lambda_j^{-1}\right)^t$.

### 3.4 Multi-Scale Analysis

Not rarely, the solution field can have effects that differ from the others by several orders of magnitude. This is the case when strong shock waves are present simultaneously with weak local recirculations on the flow field. Adaptations strategies based on the control of the $\mathbf{L}^\infty$ interpolation norm will fail in this case because small features of the flow field will not be properly captured. Anisotropy will also be lost near strong shock waves (Loiseille and Alauzet, 2009). Many local modifications have been proposed to overcome this problem, such as the scaling of the Hessian matrix by a measure pondered by both gradient and local solution values, or by just one of those. Another more general alternative is the control of the error in the $\mathbf{L}^p$ norm. This procedure enables to capture all the scales of the numerical solution. The metric $\mathcal{M}_{\mathbf{L}^p}$ available in the $\mathbf{L}^p$ norm becomes:

$$\mathcal{M}_{\mathbf{L}^p} = \left(\det\left|\mathbf{H}\right|\right)^{-\frac{1}{2p+d}} \left|\mathbf{H}\right| \tag{18}$$

The last term $|\mathbf{H}|$ of the equation specifies the local mesh orientation and anisotropy while the scalar term $\left(\det\left|\mathbf{H}\right|\right)^{-\frac{1}{2p+d}}$ modifies the local mesh density to take into account the sensitivity of the $\mathbf{L}^p$ norm used in the error estimate.

### 3.5 Multi-Metric Intersection

When more than one metric field is specified at a point, the adaptation procedure must ensure covering the interpolation error of all these fields together. To do this, a simultaneous reduction technique is used. In this approach, a common basis $\mathcal{P}$ of two metric fields $\mathcal{M}_A$ and $\mathcal{M}_B$, such that they are congruent to a diagonal matrix in this basis, is sought. The matrix $\mathcal{B} = {\mathcal{M}_A}^{-1}\mathcal{M}_B$ is introduced such that $\mathcal{B}$ is diagonalizable with real-eigenvalues. The normalized eigenvectors of $\mathcal{B}$ are $\mathbf{e}_i$ with $i = 1, d$ and composes the common diagonalization basis $\mathcal{P}$. The principal components of $\mathcal{M}_A$ and $\mathcal{M}_B$ projected in such basis can be obtained with (Loiseille and Frey, 2003):

$$\mu_i = \mathbf{e}_i^T \mathcal{M}_A \mathbf{e}_i \quad \text{and} \quad \beta_i = \mathbf{e}_i^T \mathcal{M}_B \mathbf{e}_i \tag{19}$$

As $\mathcal{P}$ is also invertible, the metric intersection can be evaluated by:

$$\mathcal{M}_{A\cap B} = \mathcal{M}_A \cap M_B = \left(\mathcal{P}^{-1}\right)^T \text{diag}\left(\max\left(\mu_i, \beta_i\right)\right) \mathcal{P}^{-1} \tag{20}$$

For more than two metric fields, the procedure can be evaluated by the same way, with $\mathcal{M}_{A\cap B\cap C}$ using $\mathcal{M}_A = \mathcal{M}_{A\cap B}$ and $\mathcal{M}_B = \mathcal{M}_C$ and so on for more metric intersections. It is important to remark that this operation is not commutative and the resulting intersected metric for more than two metric depends on the order of the evaluation.

### 3.6 Level Set for Boundary Layer

A boundary layer improvement is performed using a level set function for viscous flow problems using the methodology proposed by Loiseille and Löhner (2009). The use of a level set function $\psi\left(\mathbf{x}_i\right)$ of the distance from the body and its gradient $\nabla\psi\left(\mathbf{x}_i\right)$ allows to determine distance and orthonormal directions at each point. To impose the boundary layer, a new metric $\mathcal{M}_{bl}\left(\mathbf{x}_i\right)$ is generated based on an initial metric $\mathcal{M}$. The mesh size in the normal direction $h_{\mathbf{n}_i}$ on such metric is evaluated for a point $P_i$ through an exponential growing law of the form:

$$h_{\mathbf{n}_i} = h_0 \exp\left(\chi\psi\left(\mathbf{x}_i\right)\right) \tag{21}$$

where $h_0$ is the initial boundary layer size and $\chi$ the growing factor. Using a local Frenet frame $\left(\mathbf{u}_i, \mathbf{v}_i, \nabla\psi\left(\mathbf{x}_i\right)\right)$ allows to prescribe the sizes on the orthogonal planes $h_{\mathbf{u}_i}$ and $h_{\mathbf{v}_i}$ by:

$$h_{\mathbf{u}_i} = \left(\mathbf{u}_i^T \mathcal{M}_w \mathbf{u}_i\right)^{-2}, \quad \text{and} \quad h_{\mathbf{v}_i} = \left(\mathbf{v}_i^T \mathcal{M}_w \mathbf{v}_i\right)^{-2} \tag{22}$$

where $\mathcal{M}_w$ is the metric $\mathcal{M}$ of the boundary node for which the body point $P_i$ reaches the minimum distance on the level set function. The final boundary layer metric is given by:

$$\mathcal{M}_{bl}\left(\mathbf{x}_i\right) = \left(\mathbf{u}_i, \mathbf{v}_i, \nabla\psi\left(\mathbf{x}_i\right)\right)^T \text{diag}\left(h_{\mathbf{u}_i}^{-2}, h_{\mathbf{v}_i}^{-2}, h_{\mathbf{n}_i}^{-2}\right) \left(\mathbf{u}_i, \mathbf{v}_i, \nabla\psi\left(\mathbf{x}_i\right)\right) \tag{23}$$

where $i = 1, d$ and the term $h_{\mathbf{v}_i}$ is suppressed for two-dimensional cases. This boundary layer metric is then intersected with the metric $\mathcal{M}$. This intersection allows to capture effects like shock-boundary layer interaction. For practical reasons, the level set function and the boundary layer metric are only evaluated up to a prescribed distance from the body.

## 4 MESH ADAPTATION

The mesh adaptation procedure uses the metric-based framework to evaluate the error associated to each edge of the mesh. When the error is above the specified error, the edge is split into

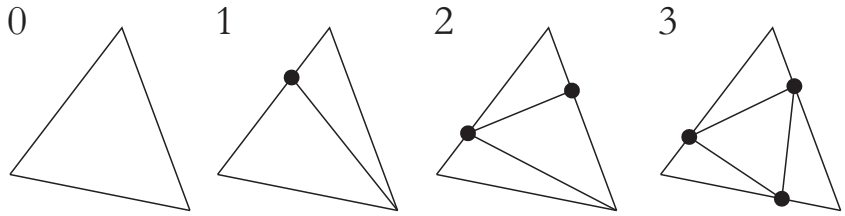



Figure 1: Edge-Based triangle subdivision scheme.

two new edges. When the error is less than the desired, the edge is removed by an edge collapse procedure. Edge and face swapping are also used to improve the mesh quality. All strategies are developed in order to adapt the mesh considering the anisotropy at each stage individually and aiming to use only naturally available mesh information at each procedure. Then, a very general and robust strategy is employed.

### 4.1 Mesh Refinement

When an edge $\mathbf{ab}$ is marked to be refined, a new node $\mathbf{c}$ is inserted in its center generating two new edges: $\mathbf{ac}$ and $\mathbf{cb}$. Usually, $\mathbf{c}$ is adopted as the Euclidean center of the edge, but using this way the anisotropic information is lost because the edge error is not equitably distributed by this point. We propose a Riemannian centered split such that the new inserted node $\mathbf{c}$ verifies the following relation:

$$\ell_{\mathcal{M}}(\mathbf{ac}) = \ell_{\mathcal{M}}(\mathbf{cb}) \tag{24}$$

The previous equation defines a non-linear relation because the integral relation given by Eq. 6 must take into account the spatial variation of the metric. Moreover, as the Log-Euclidean metric interpolation adopted is commutative, the following relationship is ensured for any point $\mathbf{c}$:

$$\ell_{\mathcal{M}}(\mathbf{ac}) + \ell_{\mathcal{M}}(\mathbf{cb}) = \ell_{\mathcal{M}}(\mathbf{ab}) \tag{25}$$

Determination of where $\mathbf{c}$ is located can be performed by a simple numerical procedure to determine the real root of Eq. 24. The nodal variables are then interpolated and projected back to the new point $\mathbf{c}$.

The refinement procedure must ensure that the topology $\mathcal{T}$ of the mesh will not be violated, generating only valid elements. In order to satisfy this condition, the partitioning of an element is performed according to the number and position of the refined edges. All anisotropic partitioning cases are covered. For a two-dimensional case, with $d = 2$, in a mesh formed by triangles, four types of subdivision are possible, covering eight anisotropic cases (see Fig. 1). This type of edge subdivision is unique and can always be performed (existence and uniqueness conditions are satisfied), ensuring that the topology will not be violated (Ruprecht and Müller, 1994).

For a three-dimensional case, with $d = 3$, in a mesh formed by tetrahedral elements, eleven types of subdivision are possible, covering sixty-four anisotropic cases (see Fig. 2): The subdivision method is fully described by Ruprecht and Müller (1994). Once again the existence and uniqueness conditions holds and the topology is not violated (Ruprecht and Müller, 1994). Because the procedure of subdivision of one element is fully independent of the division of the others, parallel processing can be easily performed to reduce computational time.

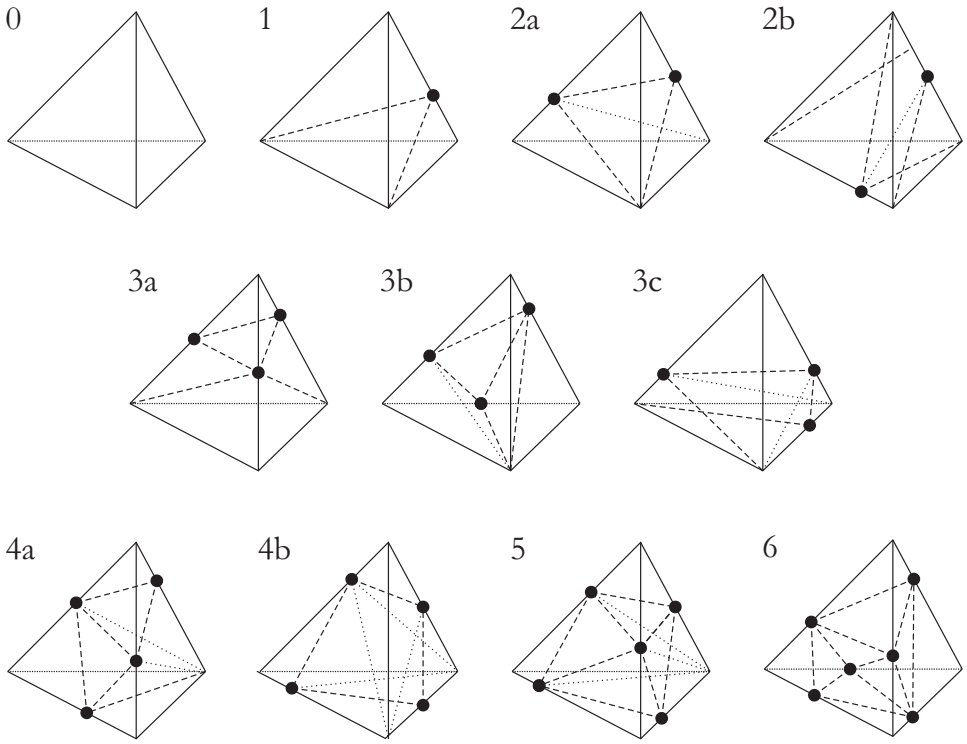



Figure 2: Edge-based tetrahedron subdivision scheme.

### 4.2 Mesh Coarsening

The coarsening procedure adopted in the present work is based in the edge collapse. The procedure has the advantage to be computationally fast and independent of the refinement history, i.e., it is not a simple annulment of a refinement step, returning to the previous situation. Usually, the mesh collapse is performed collapsing one edge vertex to the other directly, or both vertex to the Euclidean midpoint. The valid topology is successively ensured defining types of edges according to the dimensional simplex involved, requiring a different treatment for edges lying on boundary faces, identification of sharp corners, fold lines on surfaces and others cases. Thus, the anisotropic fashion is somewhat lost with such type of strategies, because the collapse performed with this methodology does not hold the anisotropic information at all and additional mesh information is required to hold a valid topology. We propose an anisotropic edge collapse strategy that optimizes the worst anisotropic quality of evolved elements surrounding the coarsened edge where the topology is preserved as an imposed constraint. The methodology is independent of the dimensional case, being directly applicable to both meshes formed by triangular and tetrahedral elements.

Let $a$ be an edge to be coarsened, with initial vertex $v_1$ and final vertex $v_2$. We define the set $G$ of all simplexes that share at least one edge with $a$ (see Fig. 3):

$$G = \{K : a \in A_K\} \tag{26}$$

Let $B$ be the set of all simplexes that shares the vertex $v_1$ but are not contained on the set $G$:

$$B = \{K : v_1 \in V_K \text{ and } a \notin A_K\} \tag{27}$$

Similarly, $C$ is defined as the set of all simplexes that shares the vertex $v_2$ but are not contained on the set $G$:

$$C = \{K : v_2 \in V_K \text{ and } a \notin A_K\} \tag{28}$$

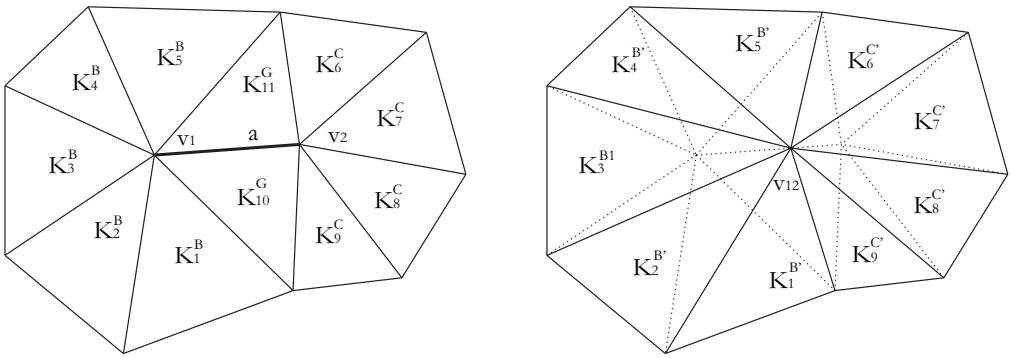



Figure 3: Edge collapse procedure.

The vertices $v_1$ and $v_2$ are joined along the direction of the edge $a$ to a new point $v_{12}$, collapsing all simplexes of the set $G$. All simplexes contained in the set $B$ are linearly transformed to a new set $B'$ by the following relationship:

$$B' = \{K \in B : (v_1 \in V_K) \mapsto v_{12}\} \tag{29}$$

Similarly, the simplexes of the set $C$ are transformed to a new set $C'$ by:

$$C' = \{K \in C : (v_2 \in V_K) \mapsto v_{12}\} \tag{30}$$

To hold the anisotropic information on the procedure, the choice of the collapsed point $v_{12}$ is performed by the minimization of an objective function of the form:

$$\min_{v_{12} \subset a} \left(\max \left(Q_{\mathcal{M}} \left(K \in B' \cup C'\right)\right)\right) \tag{31}$$

subjected to the following constraints:

$$|T_K| \ \forall \ K \in B' \cup C' > 0 \tag{32a}$$

$$\sum_{K \in G \cup B \cup C} |T_K| - \sum_{K \in B' \cup C'} |T_K| = 0 \tag{32b}$$

The objective function in Eq. 31 translates mathematically that the collapsed point $v_{12}$ must be a point which has the best quality of the worst anisotropic quality associated to each simplex affected by the collapse, thus holding size and orientation information due to the properties of $Q_{\mathcal{M}}$, already mentioned. The first and second constraints, given in Eqs. 32, are used to hold topology valid and to conserve domain bounds. If both vertices $v_1$ and $v_2$ do not belong to the boundary $\partial\mathcal{T}$, the second constraint 32b is not required to be checked. This domain conservation automatically ensures only topologically valid collapsing conditions as the case when an edge has only one vertex belonging to the boundary and can only be collapsed from the node that does not lye on such boundary or the case when both vertices belong to the boundary, allowing only collapses that do not creates holes on the domain.

Eq. 31 defines a non-linear unidimensional optimization problem subjected to non-linear constraints defined by Eqs. 32. A Sequential Quadratic Programming (SQP) algorithm is employed to solve the problem. The NLPQLP program (Schittkowski, 2010) is used, which is a SQP method using non-monotone line search tuned to run under distributed systems. As the problem is unidimensional, convergence is achieved with very few iterations.

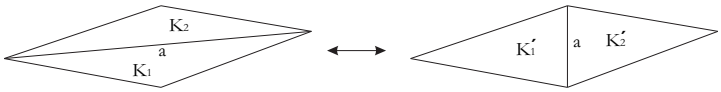



Figure 4: Edge swapping.

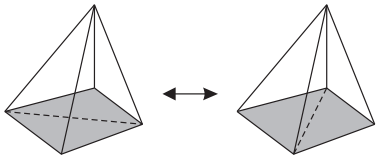


Figure 5: Boundary face swapping.

### 4.3 Edge and Face Swapping

Edge swapping is a procedure that alternates the connection of adjacent elements of the mesh keeping constant the number of nodes. It can be applied to a mesh formed by triangles or tetrahedral elements, with the difference that in the first case the alternation is unique and in the second there are a lot of possibilities to perform the mesh modification. This work is concerned with application of edge swapping only for triangular meshes. Such case can only be performed on an edge $a$ that does not belong to the boundary $\partial\mathcal{T}$. This edge shares two elements $K_1$ and $K_2$. The procedure consists of alternating the position of this edge ir order to obtain the elements $K_1'$ and $K_2'$ (see Fig. 4). To take into account the anisotropy, the following condition is checked:

$$\max\left(Q_{\mathcal{M}}\left(K'_1, K'_2\right)\right) < \max\left(Q_{\mathcal{M}}\left(K_1, K_2\right)\right) \tag{33}$$

If this condition is verified and the non-degeneration condition holds (Eq. 4), then the swapping is performed, increasing the quality of the mesh in an anisotropic sense.

The face swapping procedure is performed on tetrahedron meshes only. Two cases are possible: face swapping on interior face and on boundary face. In the first case (Fig. 5), two internal elements share a face and a new edge is inserted, resulting in three elements. The inverse procedure is also considered, going from three to two elements by removing the edge. For the second case (Fig. 6), two elements that share a face can swap through the boundary faces, resulting in two new elements. The condition checked for face swapping is analogous to the condition given by Eq. 33 for edge swapping.

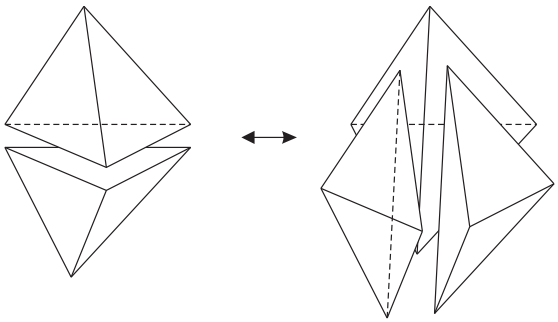


Figure 6: Interior face swapping.

### 4.4 Adaptation algorithm

Several approaches can be developed using the adaptation procedures. We use the methodology proposed by Habashi et al. (2000), following the following order:

1. Refine edges with error above a given error threshold $\eta_S$;
2. Edge/face swapping;
3. Coarse edges with error below a given error threshold $\eta_I$;
4. Edge/face swapping;

Those steps define an adaptation step. They are performed for a number of times fixed by the user. An edge $a$ is refined if:

$$E\left(a\right) \geq \eta_S \sum_{i=1,n} \frac{E\left(i\right)}{n} \tag{34}$$

where $n$ is the number of edges and it is coarsened if:

$$E\left(a\right) \leq \eta_I \sum_{i=1,n} \frac{E\left(i\right)}{n} \tag{35}$$

The upper and lower error threshold $\eta_S$ and $\eta_I$ are constants and they are adopted as being $1.4$ and $0.4$ respectively (Habashi et al., 2000).

## 5 COMPRESSIBLE FLOW SIMULATION

The simulation of compressible isentropic flows considering a thermally and calorically perfect gas, together with the assumptions that the continuum hypothesis, the Fourier Law of thermal conductibility and the Stokes' hypothesis are valid, leads to the following convection-diffusion system of partial differential equations on the conservative form for an Eulerian kinematic description (Schlichting, 1979):

$$\frac{\partial \mathbf{\Phi}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x_i} + \frac{\partial \mathbf{G}}{\partial x_i} = \mathbf{0} \tag{36}$$

where $t$ is the time variable, $x_i$ with $i = 1, d$ are the cartesian coordinates, $\mathbf{\Phi}$ contains the conservative variables:

$$\mathbf{\Phi} = \left\{ \begin{array}{c} \rho \\ \rho u_i \\ \rho E \end{array} \right\} \tag{37}$$

where $\rho$ is the mass density, $u_i$ stands for the component of velocity in the $i = 1, d$ direction and $E$ is the total energy:

$$E = e + \frac{1}{2} u_i u_i \tag{38}$$

with $e$ as the internal energy. The advective terms $\mathbf{F}$ are:

$$\mathbf{F} = \left\{ \begin{array}{c} \rho u_i \\ \rho u_j u_i + p\delta_{ji} \\ (\rho + p) E u_i \end{array} \right\} \tag{39}$$

where $p$ is the hydrostatic pressure and $\delta_{ji}$ is the Kronecker delta with $j = 1, d$. The last term of Eq. 36, $\mathbf{G}$, contains the diffusion terms:

$$\mathbf{G} = \begin{Bmatrix} 0 \\ -\tau_{ji} \\ -\tau_{ij} u_j - k \frac{\partial T}{\partial x_i} \end{Bmatrix} \tag{40}$$

where $T$ is the absolute temperature, $k$ is the diffusion coefficient and $\tau_{ij}$ are the viscous stresses components:

$$\tau_{ij} = \mu \left[ \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \frac{\partial u_k}{\partial x_k} \right] \tag{41}$$

with $\mu$ being the viscous coefficient. Body forces and source terms are not considered on Eq. 36. Neglecting the diffusion terms on Eq. 36 leads to the hyperbolic case known as the Euler equations. The closure of Eq. 36 is done with the addition of the state equation of a perfect gas:

$$p = (\gamma - 1)\, \rho e \tag{42}$$

where $\gamma = \frac{C_p}{C_v}$, with $C_p$ and $C_v$ being the specific heat coefficient at constant pressure and volume, respectively. The system of partial differential equations given by Eq. 36 must be solved considering initial and boundary conditions. For practical and numerical reasons, Eq. 36 is employed in a normalized form.

### 5.1 Numerical Solution

The numerical solution of the flow equations is performed with the Characteristic-Based Split (CBS) scheme introduced by Zienkiewicz and Codina (1995); Zienkiewicz et al. (1995). For the solution of Eq. 36, the CBS algorithm uses a fractional step with a split. The four steps can be briefly described as (Zienkiewicz et al., 2011):

1. solve momentum equation without pressure terms,
2. calculate pressure solving a Poisson equation,
3. correct velocity components,
4. calculate additional scalar variables, such as temperature, from appropriate governing equations.

The time increment is assumed to be $\Delta t = t^{n+1} - t^n$ and the flux mass $U_i^{n+1} = \rho u_i$, evaluated at time $n + 1$, is split into two terms:

$$U_i^{n+1} = U_i^n + \Delta U_i^* + \Delta U_i^{**} \tag{43}$$

A standard Galerkin finite element procedure is used for spatial discretization. We employ only linear triangular and tetrahedron elements in the present work. The spatial discretization of the variables are carried out as:

$$\begin{aligned} &U_i = \mathbf{N}_u \tilde{\mathbf{U}}_i, \quad \Delta U_i = \mathbf{N}_u \Delta \tilde{\mathbf{U}}_i, \quad \Delta U_i^* = \mathbf{N}_u \Delta \tilde{\mathbf{U}}_i^*, \quad \Delta U_i^{**} = \mathbf{N}_u \Delta \tilde{\mathbf{U}}_i^{**} \\ &u_i = \mathbf{N}_u \tilde{\mathbf{u}}_i, \qquad p = \mathbf{N}_p \tilde{\mathbf{p}}, \qquad \rho = \mathbf{N}_\rho \tilde{\rho} \end{aligned} \tag{44}$$

where a tilde superscript represents a nodal quantity and $\mathbf{N}_i$ are the shape functions such that:

$$\tilde{\mathbf{U}}_i = \left[U_i^1, U_i^2, \cdots, U_i^a, \cdots, U_i^m\right]^T \tag{45}$$

$$\mathbf{N}_j = \left[N_j^1, N_j^2, \cdots, N_j^a, \cdots, N_j^m\right] \tag{46}$$

with $a$ being a node (or variable), which varies from 1 to $m$, $i = 1, p$ and $j = u_i, p, \rho$.

The semi-discrete forms of the CBS equations are then weighted by $\mathbf{N}^T$ and integrated over the domain. The final equations for the four steps of the CBS scheme for the explicit formulation is summarized as:

- Step 1

$$\begin{aligned}\mathbf{M}\Delta\tilde{\mathbf{U}}_i^* = \Delta t\Big[(\mathbf{C}-\mathbf{S})\left(\tilde{\mathbf{u}}_j\tilde{\mathbf{U}}_i\right) + (\mathbf{K}_\tau - \mathbf{T}_\tau)\,\tau_{ij}\Big]^n \\ + \frac{\Delta t^2}{2}\tilde{\mathbf{u}}_k\Big[(\mathbf{K}_u - \mathbf{T}_u)\left(\tilde{\mathbf{u}}_j\tilde{\mathbf{U}}_i\right) + (\mathbf{K}_p - \mathbf{T}_p)\,\tilde{\mathbf{p}}\Big]^n\end{aligned} \tag{47}$$

- Step 2

$$\mathbf{M}\Delta\tilde{\rho} = \Delta t\Big[(\mathbf{D}-\mathbf{R}_u)\left(\tilde{\mathbf{U}}_i + \theta_1\Delta\tilde{\mathbf{U}}_i^*\right) - \Delta t\theta_1\mathbf{K}\tilde{\mathbf{p}}\Big]^n \tag{48}$$

- Step 3

$$\mathbf{M}\tilde{\mathbf{U}}_i = \mathbf{M}\Delta\tilde{\mathbf{U}}_i^* + \Delta t[(\mathbf{D}-\mathbf{R}_u)\,\tilde{\mathbf{p}}]^n \tag{49}$$

- Step 4

$$\begin{aligned}\mathbf{M}\Delta\tilde{\rho}\tilde{\mathbf{E}} = \Delta t\left[(\mathbf{C}-\mathbf{S})\,\tilde{\mathbf{u}}_j\left(\tilde{\rho}\tilde{\mathbf{E}} + \tilde{\mathbf{p}}\right) + (\mathbf{T}_\tau - \mathbf{K}_\tau)\left(k\frac{\partial T}{\partial x_i} + \tau_{ij}\tilde{\mathbf{u}}_j\right)\right]^n \\ - \frac{\Delta t^2}{2}\left[\tilde{\mathbf{u}}_k\,(\mathbf{K}_u - \mathbf{T}_u)\,\tilde{\mathbf{u}}_j\left(\tilde{\rho}\tilde{\mathbf{E}} + \tilde{\mathbf{p}}\right)\right]^n\end{aligned} \tag{50}$$

where vector and matrices are given by:

$$\mathbf{M} = \int_\Omega \mathbf{N}^T \mathbf{N} d\Omega \tag{51a}$$

$$\mathbf{C} = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} \mathbf{N} d\Omega \tag{51b}$$

$$\mathbf{D} = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_i} \mathbf{N} d\Omega \tag{51c}$$

$$\mathbf{K} = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_i} \frac{\partial \mathbf{N}}{\partial x_i} d\Omega \tag{51d}$$

$$\mathbf{K}_\tau = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_j} d\Omega \tag{51e}$$

$$\mathbf{K}_u = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_k} \frac{\partial \mathbf{N}}{\partial x_j} d\Omega \tag{51f}$$

$$\mathbf{K}_p = \int_\Omega \frac{\partial \mathbf{N}^T}{\partial x_k} \frac{\partial \mathbf{N}}{\partial x_i} d\Omega \tag{51g}$$

$$\mathbf{R}_u = \int_\Omega \mathbf{N}^T \mathbf{N} n_i d\Omega \tag{51h}$$

$$\mathbf{S} = \int_\Gamma \mathbf{N}^T \mathbf{N} n_j d\Gamma \tag{51i}$$

$$\mathbf{T}_\tau = \int_\Gamma \mathbf{N}^T n_j d\Gamma \tag{51j}$$

$$\mathbf{T}_u = \int_\Gamma \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial x_j} n_k d\Gamma \tag{51k}$$

$$\mathbf{T}_p = \int_\Gamma \mathbf{N}^T \frac{\partial \mathbf{N}}{\partial x_i} n_k d\Gamma \tag{51l}$$

with $\theta_1 = 0.5$.

### 5.2 Treatment of Compressible Flow Problems

To achieve a smooth solution for a large range of Mach numbers and to take into account temperature dependent viscosity, additional features are included on the CBS algorithm. Artificial dissipation based on the Hessian of pressure (Nithiarasu et al., 1998) and Sutherland's temperature dependent viscosity relation are employed.

### 5.3 Steady-State Solution

The four steps of the CBS algorithm are carried out in order to a steady-state solution using a lumped mass matrix until the following tolerance $R_T$ is achieved:

$$R_T = \sqrt{\frac{\sum_{a=1}^{nnodes} \left(\rho_a^{n+1} - \rho_a^n\right)^2}{\sum_{a=1}^{nnodes} \left(\rho_a^{n+1}\right)^2}} \tag{52}$$

where $nnodes$ is the number of nodes of the mesh and $\rho_a$ is the specific mass evaluated at node $a$. The adimensional tolerance is set to $R_T = 10^{-5}$ to reach a steady-state solution. Local time

stepping is employed. This enables different time steps at different nodes and the solution is accelerated to obtain the steady state solution (Nithiarasu et al., 2006).

## 6 APPLICATIONS

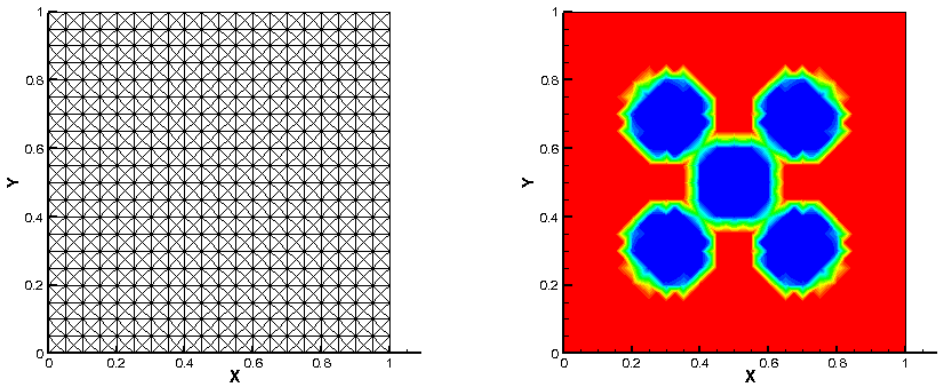
### 6.1 Analytical Field

To demonstrate the capabilities of the developed mesh adaptation methodology, an imposed analytical field is first studied over a square domain $\Omega = [0,1] \times [0,1]$. The imposed field $u$ is given by:

$$\begin{aligned} u = &\tanh\left(30\left((4x-2.00)^2+(4y-2.00)^2-0.25\right)\right) \\ &+\tanh\left(30\left((4x-2.75)^2+(4y-2.75)^2-0.25\right)\right) \\ &+\tanh\left(30\left((4x-2.75)^2+(4y-1.25)^2-0.25\right)\right) \\ &+\tanh\left(30\left((4x-1.25)^2+(4y-1.25)^2-0.25\right)\right) \\ &+\tanh\left(30\left((4x-1.25)^2+(4y-2.75)^2-0.25\right)\right) \end{aligned} \tag{53}$$

The initial structured mesh contains 841 nodes and 1600 triangular elements. The initial mesh and solution field are shown in Fig. 7. The adaptation is performed for two cases:

- Case 1: adaptation with the present approach for refinement, coarsening and swapping;
- Case 2: adaptation with the present approach for swapping, but default refinement and coarsening strategies.




(a) Mesh (b) Solution

Figure 7: Initial mesh and solution field of example 1.

where we define a default refinement strategy as an Euclidean midpoint split instead of the Riemannian one, and the default coarsening strategy as an edge collapse allowed to only one of the end points of the edge. To measure the error distribution over the mesh, the error variance $\sigma^2(E)$ is used. A comparison of error variance with respect to the number of nodes on the mesh is shown in Fig. 8 for the two cases analysed here. Each point is a mesh adaptation step using the same input parameters, differing only by the adaptation approach of each case. It can be observed that the present approach for refinement and coarsening strategies shows

lower oscillation on error variance through the adaptation steps. With a default refinement and coarsening strategies the adaptation operates as a binary 'black or white' strategy, where the refinement and coarsening procedures can lead to bad distributed operations. This is the case when the edge is a bit above the coarsening threshold and after the edge collapse the same edge pass to has a high edge error, requiring edge refinement and also the inverse operation (coarsening). The result of such type of strategy is therefore an oscillating process, which can be seen in Fig. 8, where the error variance grows up at some iterations instead of decreasing, even when the number of nodes increases at each step. Fig. 9 shows the adapted mesh for the

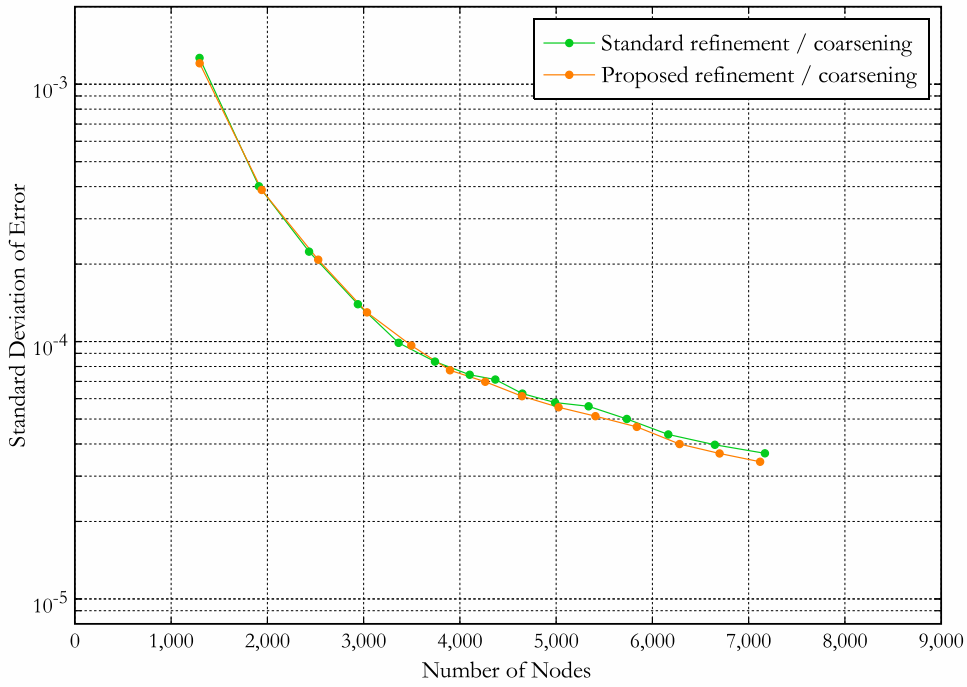
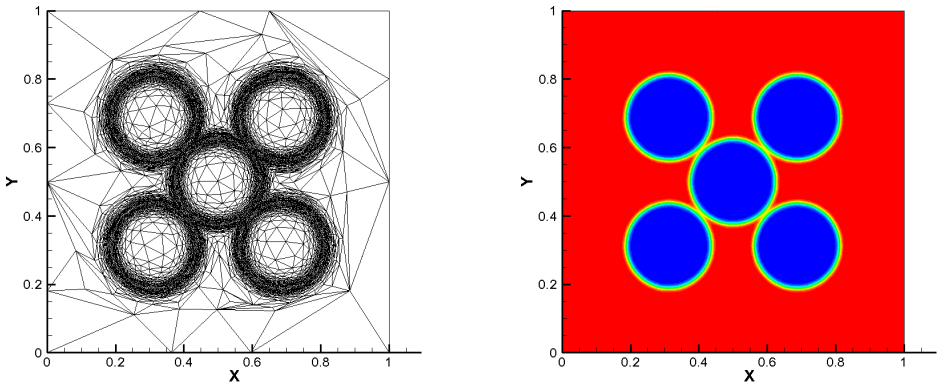



Figure 8: Error distribution through the adaptation steps of example 1.

present methodology (case 1) after twenty-five adaptation steps and Fig. 10 shows a detail of the same adapted mesh, where it can be seen the well sized and oriented elements with the anisotropic field and the isolines of the solution field. Table 1 presents a comparison of number of nodes, elements and error variance for some adaptation steps for the four adaptation cases. It can be seen that the present methodology shows good performance during the adaptation iterations when compared to the other cases, allowing a smoother adaptation convergence.

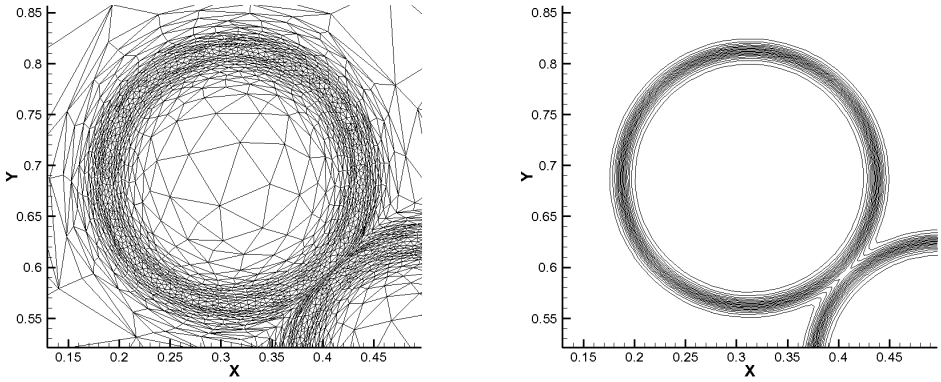
| Adaptation Case | Adaptation Step | Number of Nodes | $\sigma^2(E)$ |
|---|---|---|---|
| Case 1 | 1 | 1296 | $1.205 \times 10^{-3}$ |
| Case 1 | 10 | 5408 | $5.129 \times 10^{-5}$ |
| Case 1 | 14 | 7119 | $3.411 \times 10^{-5}$ |
| Case 2 | 1 | 1295 | $1.264 \times 10^{-3}$ |
| Case 2 | 10 | 4989 | $5.794 \times 10^{-5}$ |
| Case 2 | 14 | 6649 | $3.971 \times 10^{-5}$ |

Table 1: Mesh adaptation of example 1




(a) Mesh (b) Solution

Figure 9: Adapted mesh and solution field of example 1.



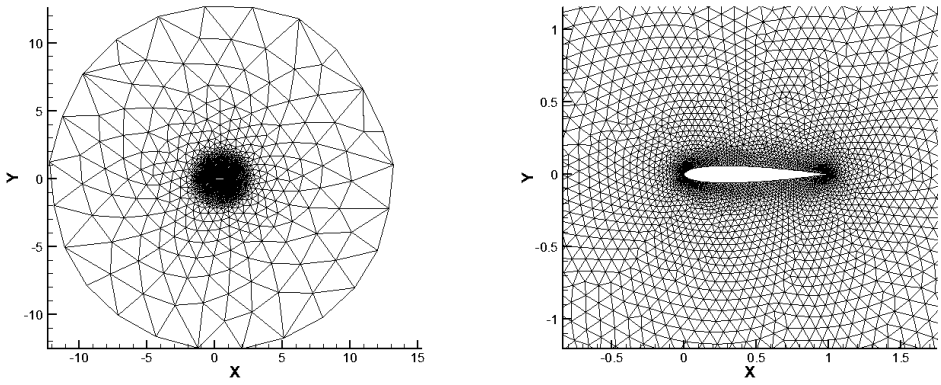

(a) Mesh (b) Isolines of the solution

Figure 10: Detail of the adapted mesh and solution isolines for the solution field of example 1.

### 6.2 Inviscid Supersonic Flow Around NACA0012 Airfoil

In this second example, the flow around a standard NACA0012 airfoil is analysed. The problem is treated as an inviscid flow, with an incoming supersonic free stream corresponding to Mach number $M_\infty = 2$ at zero angle of attack. The chord length of the airfoil $L$ is equal to one. The initial mesh contains $3753$ nodes and $7351$ triangular elements. The adaptation is performed with ten adaptation steps with each one spaced by $2000$ time steps times. The first adaptation is performed only after the initial steady-state convergence and after the last adaptation the problem is simulated until convergence is achieved once again. The initial mesh is shown in Fig. 11 and the final one in Fig. 12, having $36024$ nodes and $71643$ elements. The Mach field of the adapted solution is shown in Fig. 13.

The adapted mesh captures many effects of the flow that have different magnitudes. First, the presence of a frontal shock is the strongest effect in the flow, with the adapted mesh aligned with this anisotropy until the end of the computational domain. A secondary and weaker shock wave occurs and is correctly captured starting from the trailing edge of the airfoil. The mesh is also aligned with the shock direction until the end of the computational domain. The third anisotropic effect captured by the mesh adaptation is the straight wake prolongation from the

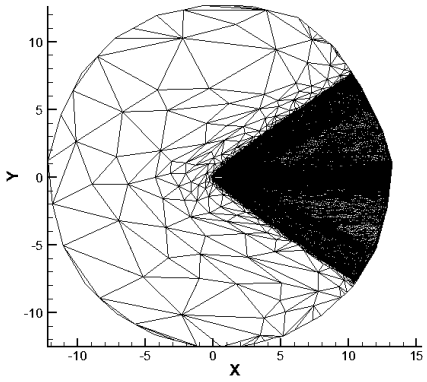



(a) Global view (b) Detail

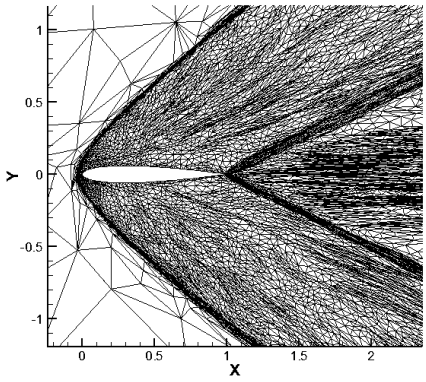
Figure 11: Initial mesh of example 2.

trailing edge to the end of the domain, which is a weaker effect compared to the shock waves. The incoming flow before the frontal shock zone is also a feature to be captured by the adaptation. This zone has almost none gradients and can be strongly coarsened to save computational resources.

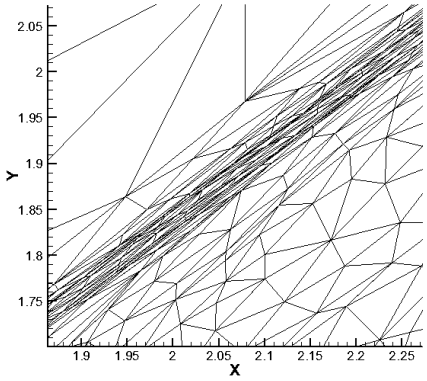
A comparison of the pressure coefficient $C_p$ obtained by Dolejsí and Felcman (2004) with that calculated here is presented in Fig. 14. Good agreement is obtained for the entire chord of the airfoil. The standard deviation of the error of the initial solution was $\sigma^2(E) = 2.426 \times 10^{-3}$ and the final value was $\sigma^2(E) = 1.713 \times 10^{-4}$ for the tenth adaptation step.



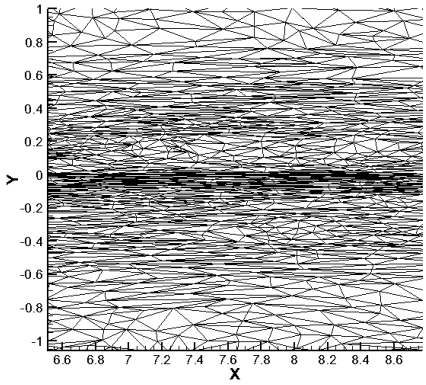

(a) Global view




(b) Airfoil detail

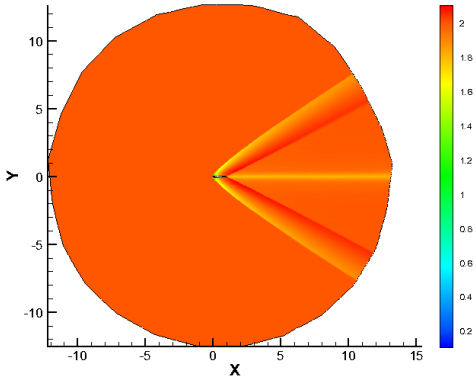



(c) Shock wave detail

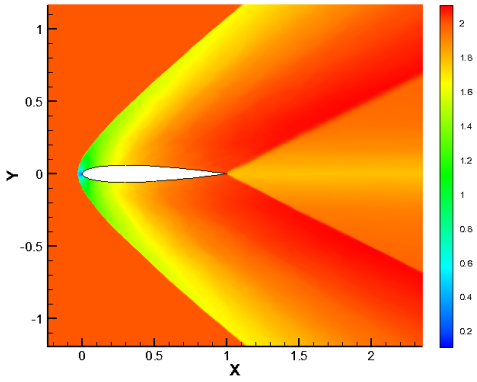



(d) Wake zone detail

Figure 12: Final adapted mesh of example 2.




(a) Global view




(b) Detail

Figure 13: Mach field for the final adapted mesh of example 2.

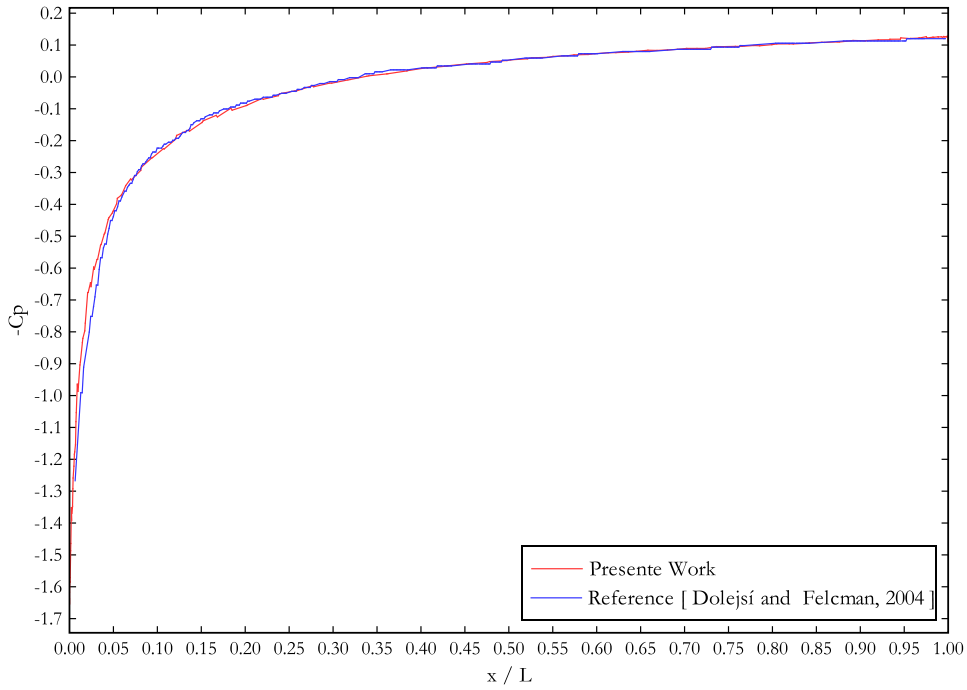
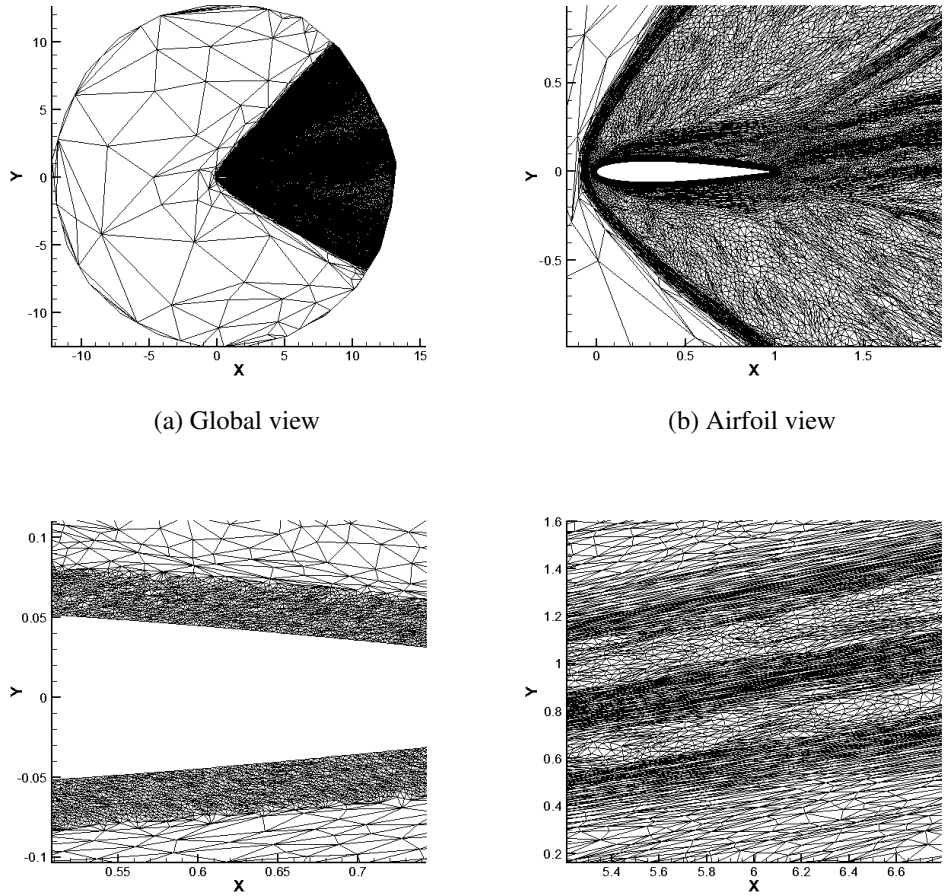



Figure 14: Pressure coefficient of example 2.

### 6.3 Viscous Supersonic Flow Around NACA0012 Airfoil

The same NACA0012 airfoil studied on the last example is here analysed again but for a viscous flow. The incoming supersonic free stream Mach number of $M_\infty = 2$ is used again, but at an angle of attack of $10^o$ with Reynold's number of $Re = 1000$. The same initial mesh is employed (Fig. 11) and the adaptation is performed through fifteen adaptation steps spaced by $2000$ times intervals. For a viscous flow simulation, the initial mesh employed is not well suitable due to the representation of the boundary layer with a very coarse mesh. The boundary layer metric is evaluated and intersected to the metric of the flow solution in the present example. This allows a continuous boundary layer preservation, where a boundary layer thickness of $0.03L$ with $h_0 = 0.003L$ was employed, being $L$ the chord length of the airfoil. The final mesh is shown in Fig. 15, having $80432$ nodes and $159783$ elements. The Mach field of the adapted solution is shown in Fig. 16. The hyperbolic flow characteristic of the second example




(a) Global view

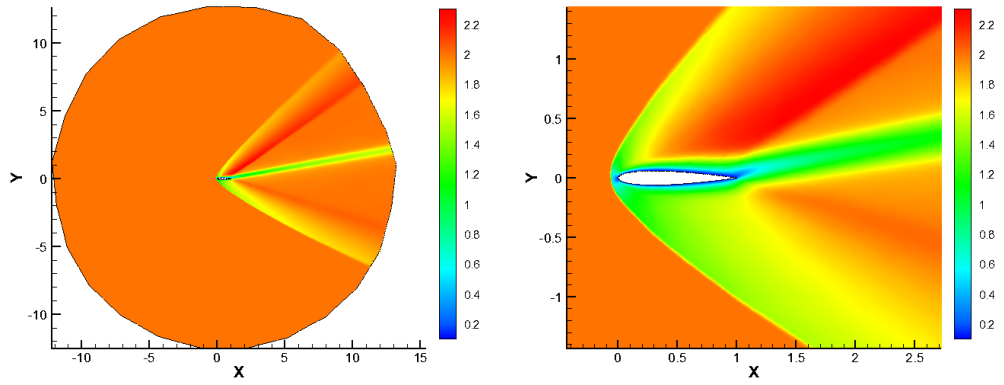
(b) Airfoil view

(c) Boundary layer detail

(d) Wake zone detail

Figure 15: Final adapted mesh of example 3.

is lost here due to the viscous effects. Although both the frontal and the back shock waves are still present, the boundary layer modifies the flow speed nearly the airfoil prolonging the effect through the wake zone. The adapted mesh contemplates the main effects of shock waves and the imposed boundary layer preservation. Without this imposition, the semi-structured boundary layer mesh region can be lost at some parts due to reduced gradients at some regions.




(a) Global view (b) Detail

Figure 16: Mach field of final adapted mesh of example 3.

This quasi-structured boundary layer can be more stretched when increasing the exponential growing law depending of the application. Some applications, however, are better developed for more isotropic boundary layer grids, like turbulent flows. This can be achieved reducing the growing law of the boundary layer metric.

Aerodynamic lift coefficient $C_L$ and drag coefficient $C_D$ are compared for the initial and adapted meshes with results obtained by Marshall et al. (2004), as it is shown in Tab. 2. The standard deviation of the error of the initial solution was $\sigma^2(E) = 9.115 \times 10^{-3}$ and the standard deviation of the error in the final mesh was $\sigma^2(E) = 2.544 \times 10^{-4}$ for the last adaptation step.

| Font | $C_L$ | $C_D$ |
|---|---|---|
| Present work (initial mesh) | 0.3558 | 0.2745 |
| Present work (adaptaded mesh) | 0.3310 | 0.2671 |
| Marshall et al. (2004) | 0.3284 | 0.2662 |

Table 2: Aerodynamic coefficient of example 3

### 6.4 Inviscid Supersonic Flow Around AGM-114 Hellfire Missile

The final example is a supersonic three-dimensional inviscid flow around an AGM-117 Hellfire. An incoming supersonic free stream with Mach number $M_\infty = 1.2$ at zero angle of attack is considered. The chord length of the airfoil $L$ is equal to one. The initial mesh contains $510\ 407$ nodes and $2\ 747\ 055$ elements. Full geometry is considered. The adaptation is performed through three adaptation steps spaced by $3000$ time intervals, being the first one performed only after initial convergence and after the last adaptation the problem is simulated until convergence is achieved. The initial mesh is shown in Fig. 17 and the final one in Fig. 18, having $19\ 860\ 325$ nodes and $115\ 496\ 405$ elements. The Mach field of the adapted solution is shown in Fig. 19. The standard deviation of the error of the initial solution was $\sigma^2(E) = 5.772 \times 10^{-3}$ and the final one was $\sigma^2(E) = 1.261 \times 10^{-3}$ for the last adaptation. This problem demonstrates a large-scale application of the methodology in a very complex real-form geometry. Although only three adaptations were performed, results are already quite satisfactory.

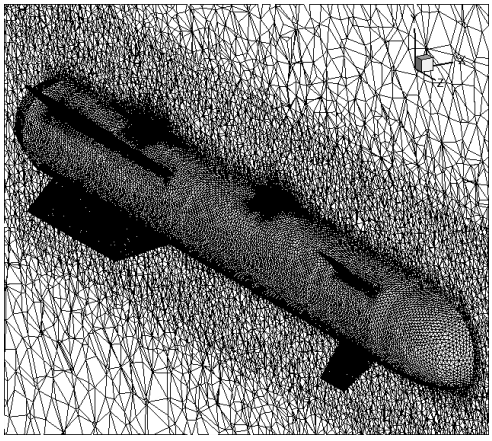
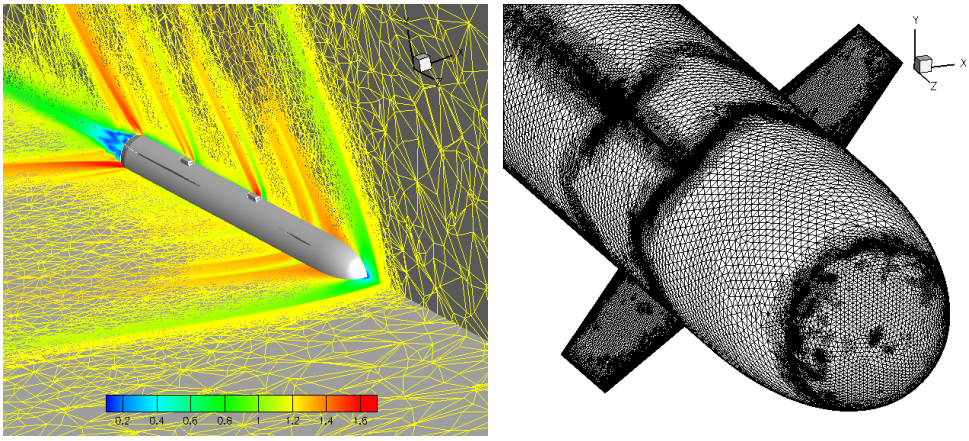
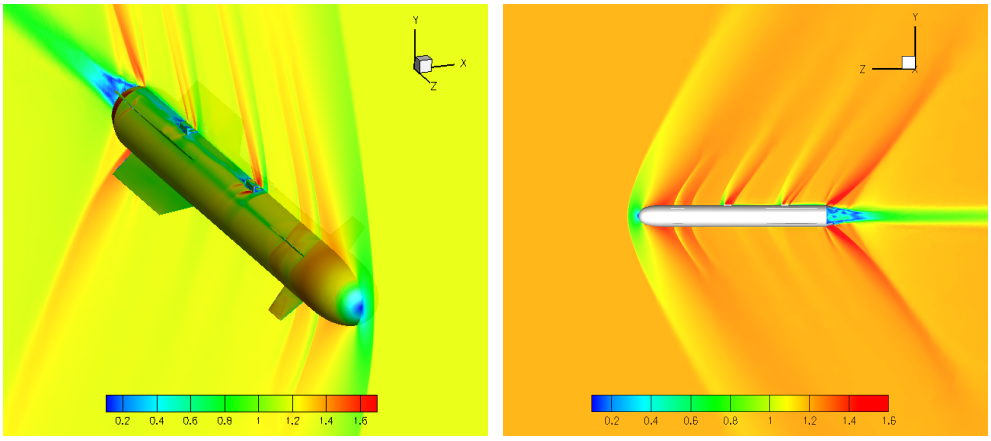


Figure 17: Initial mesh of example 4.




(a) Global view with Mach field (b) Detail of the adapted mesh

Figure 18: Final adapted mesh of example 4.




(a) Frontal view (b) Lateral view

Figure 19: Mach field of final adapted mesh of example 4.

## 7 CONCLUSION

The developed adaptation strategy was successfully employed to achieve good results for compressible flows simulations involving inviscid and viscous flows. The methodology shows

an interesting smooth convergence capability compared to other approaches presented here. The anisotropy quality included in the objective function of the optimization problem has achieved good anisotropy levels and the capability of prescribing constraints directly on the SQP algorithm enables a very generalized topological verification and guarantee. The dimensional generalization is also covered. As a drawback, the three-dimensional approach is practicable only for steady-state solutions due to the high computational efforts required.

## 8 ACKNOWLEDGMENTS

The authors wish to thank to CAPES and CNPq (Brazilian Research Committees) for their financial support and they thank also to CESUP (UFRGS Supercomputing Center) for its important contributions.

## REFERENCES

Alauzet F. and Oliver G. Extension of metric-based anisotropic mesh adaptation to time-dependent problems involving moving geometries. In *Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, 49th AIAA, pages 1–22. 2011.

Alexandra C. and Ducrot V. Levelsets and anisotropic mesh adaptation. *Discrete and Continuous Dynamical Systems*, Vol. 23(1–2):165–183, 2009.

Arsigny V., Fillard P., and Ayache N. Log-euclidean metrics for fast and simple calculus on diffusion tensors. *Magnetic Resonance in Medicine*, Vol. 56(2):441–421, 2006.

Castro-Díaz M.J., Hecht F., Mohammadi B., and Pironneau O. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, Vol. 25(4):475–49, 1997.

Cougny H.L. and Shephard M.S. Parallel refinement and coarsening of tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, Vol. 46(1):1101–1125, 1999.

Dolejsí V. and Felcman J. Anisotropic mesh adaptation for numerical solution of boundary value problems. *Numerical Methods for Partial Differential Equations*, Vol. 20(4):576–608, 2004.

Freitag L.A. and Ollivier-Gooch C. Tetrahedral mesh improvement using swapping and smoothing. *International Journal for Numerical Methods in Engineering*, Vol. 40(1):3979–4002, 1997.

Frey P.J. and Alauzet F. Anisotropic mesh adaptation for cfd computations. *Comput. Methods Appl. Mech. Engrg*, Vol. 194(48–49):5068–5082, 2005.

Habashi W.G., Dompierre J., Bourgault Y., Ait-Ali-Yahia D., Fortin M., and Vallet M.G. Anisotropic mesh adaptation: Towards user-independent, mesh-independent and solver independent cfd. part i: General principles. *International Journal for Numerical Methods in Fluids*, Vol. 32(6):725–744, 2000.

Loiseille A. and Alauzet F. Optimal 3d highly anisotropic mesh adaptation based on the continuous mesh framework. In *Proc. in 18th International Meshing Roundtable*, Springer-Verlag, pages 575–594. 2009.

Loiseille A., Alauzet F., Mohammadi B., and Pironneau O. Continuous mesh framework - part i: Well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, Vol. 49(1):38–60, 2011.

Loiseille A. and Frey P.J. Estimateur d'erreur géométrique et métrique anisotropes pour l'adaptation de maillage. partie i : aspects théoriques. Technical report, INRIA, RR-4759, France, 2003.

Loiseille A. and Löhner R. On 3d anisotropic local remeshing for surface, volume, and boundary layers. In *Proc. in 18th International Meshing Roundtable*, Springer-Verlag, pages 611–630. 2009.

Loiselle A. *Adaptation de Maillage Anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluids: Application à lá prédiction haute-fidélité du bang sonique*. Ph.D. thesis, Université Pierre et Marie Curie , Paris VI, 2008.

Marshall D.D., Ruffin S.M., Vallet M.G., and Guibault F. An embedded boundary cartesian grid scheme for viscous flows using a new viscous wall boundary condition treatment. In *Aerospace Sciences Meeting and Exhibit*, 42nd AIAA, pages 1–8. 2004.

Nithiarasu P., Codina R., and Zienkiewicz O.C. The characteristic-based split (cbs) scheme - a unified approach to fluid dynamics. *International Journal for Numerical Methods in Engineering*, Vol. 66(10):1514–1546, 2006.

Nithiarasu P., Zienkiewicz O.C., Satya-Sai B.V.K., Morgan K., and Vázquez M. Shock capturing viscosities for the general fluid mechanics algorithm. *International Journal for Numerical Methods in Fluids*, Vol. 28(9):1325–1353, 1998.

Ollivier-Gooch C. Coarsening unstructured meshes by edge contraction. *International Journal for Numerical Methods in Engineering*, Vol. 57(1):391–414, 2003.

Ruprecht D. and Müller H. A scheme for edge-based adaptive tetrahedron subdivision. Technical report, Universität Dortmund, Dortmund, Germany, 1994.

Schittkowski K. Nlpqlp: A fortran implementation of a sequential quadratic programming algorithm with distributed and non-monotone line search. Technical report, University of Bayreuth, Department of Computer Science, 2010.

Schlichting H. *Boundary Layer Theory 6th edition*. McGraw-Hill, 1979.

Vallet M.G. *Génération de maillages éléments finis anisotropes et adaptatifs*. Ph.D. thesis, Université Pierre et Marie Curie , Paris VI, 1992.

Walter M.A.T., Abdu A.A.Q., Silva L.F.F., and Azevedo J.L. Evaluation of adaptive mesh refinement and coarsening for the computation of compressible flow on unstructured meshes. *International Journal for Numerical Methods in Engineering*, Vol. 49(1):999–1014, 2005.

Zienkiewicz O.C. and Codina R. *Search for a general fluid mechanics algorithm*. Frontiers of Computational Fluid Dynamics, Wiley, 1995.

Zienkiewicz O.C., Morgan K., and Satya B.V.K.S. A general algorithm for compressible and incompressible flow - part ii: Tests on the explicit form. *International Journal for Numerical Methods in Fluids*, Vol. 20(1):887–913, 1995.

Zienkiewicz O.C., Taylor R.L., and Nithiarasu P. *The Finite Element Method For Fluid Dynamics*. Butterworth-Heinemann, 6th edition, 2011.