Anisotropic Goal-Oriented Mesh Adaptation for Unsteady Viscous Compressible Flows

A. Belme[b], F. Alauzet[a], A. Dervieux[b]

[a] INRIA, Projet Gamma, Domaine de Voluceau, Rocquencourt, BP 105,
78153 Le Chesnay Cedex, France.
[b] INRIA, Projet Tropics, 2004 route des lucioles - BP 93,
06902 Sophia Antipolis Cedex, France

Abstract

Key words: Viscous compressible flow, goal-oriented mesh adaptation, anisotropic mesh adaptation, adjoint, metric

1. Introduction

Engineering problems frequently require computational fluid dynamics (CFD) solutions with functional outputs of specified accuracy. The computational resources available for these solutions are often limited and errors in solutions and outputs are difficult to control. CFD solutions may be computed with an unnecessarily large number of mesh vertices (and associated high cost) to try to ensure that the outputs are computed within a required accuracy. One of the powerful methods for increasing the accuracy and reducing the computational cost is anisotropic mesh adaptation, the purpose of which is to control the accuracy of the numerical solution by changing the discretization of the computational domain according to mesh size and mesh directions constraints. This technique allows (i) to automatically capture the anisotropy of the physical phenomena, (ii) to substantially reduce the number of degrees of freedom, thus impacting favorably the CPU time, and (iii) to access to high order asymptotic convergence.

The objective of this paper is to propose a time-accurate anisotropic mesh adaptation method for functional outputs of Navier-Stokes calculations.

Pioneering works have shown a fertile development of Hessian-based or metric-based methods [17, 37, 35, 24, 38, 11, 15, 19, 22, 26] which rely on an ideal representation of the *interpolation error* and of the *mesh*. The "multiscale" version relies on the optimization of the $\mathbf{L^p}$ norm of the interpolation error [29]. It allows to approximate discontinuous solutions with higher-order convergence [33]. However, these methods are limited to the minimization of some interpolation errors for some solution fields, the "sensors", and do not take into account the PDE being solved. If for many applications, this simplifying standpoint is an advantage, there are also many applications where Hessian-based mesh adaptation is far from optimal regarding the way the degrees of freedom are distributed in the computational domain. Indeed, Hessian-based methods aim at controlling the interpolation error but this purpose is not often so close to the objective that consists in obtaining the best solution of the PDE. Further, in many engineering applications, a specific scalar output needs to be accurately evaluated, e.g. lift, drag, heat flux, and Hessian-based adaptation does not address this issue.

In contrast, *goal-oriented* mesh adaptation does focus on deriving the best mesh to observe a given output functional. Goal-oriented methods result from a series of papers dealing with *a posteriori estimates* (see e.g. [10, 39, 21, 38, 36, 40, 25]). However, extracting informations concerning mesh anisotropy from an *a posteriori* estimate is a difficult task. Starting from *a priori estimates*, Loseille *et al.* proposed in [32] a fully anisotropic goal-oriented mesh adaptation technique for steady problems. This latter method combines goal-oriented rationale and the application of Hessian-based analysis to truncation error.

Mesh adaptation for unsteady flows is also an active field of research and brings an attracting increase in simulation efficiency. Complexity of the algorithms is larger than for steady case: for most flows, the mesh should change during the time interval. Meshes can be moved as in [9], pattern-split [14, 27], locally refined [7], or globally rebuild as in [2, 23]. Hessian-based methods are essentially applied with a non-moving mesh system. In this paper, we do *not* account for *time discretization error* but concentrate on spatial error in unsteady simulations. A mesh adaptation fixed-point method was proposed in [2]. The Hessian criteria at the different time steps of a sub-interval are synthesized into a single criterion for these steps with the metric intersection [2, 23]. A mesh-PDE solver iteration is applied on time sub-intervals. Extension to $\mathbf{L}^p$ error estimator [33] requires: (i) space-time $\mathbf{L}^\infty - \mathbf{L}^p$ error analysis, (ii) a global fixed-point algorithm to converge the mesh adaptation. This extension has been proposed in [7].In [12], we combine the fully anisotropic goal-oriented mesh adaptation method of [32] and the global fixed-point advances of [7] for unsteady Euler flow.

The object of the present work is the extension of the [12] goal-oriented global fixed point to the Navier-Stokes system. To this end, ...

We start this paper with...

2. Continuous mesh model

2.1. Mesh parametrization

We propose to work in the continuous mesh framework, introduced in [30, 31]. The main idea of this framework is to model continuously discrete meshes by Riemannian metric spaces. It allows us to define proper differentiable optimization [1, 8], *i.e.,* to use a calculus of variations on continuous meshes which cannot apply on the class of discrete meshes. This framework lies in the class of metric-based methods.
A continuous mesh $\mathbf{M}$ of computational domain Ω is identified to a Riemannian metric field [13] $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric 3×3 matrix having $(\lambda_i(\mathbf{x}))_{i=1,3}$ as eigenvalues along the principal directions $\mathcal{R}(\mathbf{x}) = (\mathbf{v}_i(\mathbf{x}))_{i=1,3}$. Sizes along these directions are denoted $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$ and the three *anisotropy quotients* r_i are defined by: $r_i = h_i^3\,(h_1h_2h_3)^{-1}$. The diagonalisation of $\mathcal{M}(\mathbf{x})$ writes:

$$\mathcal{M}(\mathbf{x}) = d^{\frac{2}{3}}(\mathbf{x})\,\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_1^{-\frac{2}{3}}(\mathbf{x}) & & \\ & r_2^{-\frac{2}{3}}(\mathbf{x}) & \\ & & r_3^{-\frac{2}{3}}(\mathbf{x}) \end{pmatrix} {}^t\mathcal{R}(\mathbf{x}), \tag{1}$$

The *node density* d is equal to: $d = (h_1h_2h_3)^{-1} = (\lambda_1\lambda_2\lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$. By integrating the node density, we define the *complexity* C of a continuous mesh which is the continuous counterpart of the total number of vertices:

$$C(\mathbf{M}) = \int_\Omega d(\mathbf{x})\,\mathrm{d}\mathbf{x} = \int_\Omega \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x}.$$

Given a continuous mesh $\mathbf{M}$, we shall say, following [30, 31], that a discrete mesh $\mathcal{H}$ of the same domain Ω is a **unit mesh with respect to M**, if each tetrahedron $K \in \mathcal{H}$, defined by its list of edges $(\mathbf{e}_i)_{i=1...6}$, verifies:

$$\forall i \in [1,6], \quad \ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \quad \text{and} \quad Q_{\mathcal{M}}(K) \in [\alpha, 1] \ \text{ with } \ \alpha > 0\,,$$

in which the length of an edge $\ell_{\mathcal{M}}(\mathbf{e}_i)$ and the quality of an element $Q_{\mathcal{M}}(K)$ are defined as follows:

$$Q_{\mathcal{M}}(K) = \frac{36}{3^{\frac{1}{3}}}\frac{|K|_{\mathcal{M}}^{\frac{2}{3}}}{\sum_{i=1}^{6}\ell^2_{\mathcal{M}}(\mathbf{e}_i)} \in [0,1], \ \text{ with } |K|_{\mathcal{M}} = \int_K \sqrt{\det(\mathcal{M}(\mathbf{x}))}\,\mathrm{d}\mathbf{x},$$
$$\text{and } \ \ell_{\mathcal{M}}(\mathbf{e}_i) = \int_0^1 \sqrt{{}^t\mathbf{ab}\ \mathcal{M}(\mathbf{a}+t\,\mathbf{ab})\ \mathbf{ab}}\,\mathrm{d}t, \ \text{ with } \mathbf{e}_i = \mathbf{ab}.$$

We choose a tolerance α equal to 0.8.

We want to emphasize that the set of all the discrete meshes that are unit meshes with respect to a unique $\mathbf{M}$ contains an infinite number of meshes.

2.2. Continuous interpolation error

Given a smooth function u, to each unit mesh $\mathcal{H}$ with respect to $\mathbf{M}$ corresponds a local interpolation error $|u-\Pi u|$. In [30, 31], it is shown that all these interpolation errors are well represented by the so-called continuous interpolation error related to $\mathbf{M}$, which is expressed locally in terms of the Hessian H_u of u as follows:

$$\begin{aligned}(u-\pi_{\mathcal{M}}u)(\mathbf{x},t) &= \frac{1}{10}\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x},t)|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}))\\ &= \frac{1}{10}d(\mathbf{x})^{-\frac{2}{3}}\sum_{i=1}^{3} r_i(\mathbf{x})^{\frac{2}{3}}\,{}^t\mathbf{v_i}(\mathbf{x})\,|H_u(\mathbf{x},t)|\,\mathbf{v_i}(\mathbf{x}),\end{aligned} \tag{2}$$

where $|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues and where time-dependency notations ", t)" have been added for use in next sections.

2.3. *Mesh convergence*

Convergence analysis for unstructured meshes is generally made difficult by the poor assumptions that are made concerning the sequence of meshes considered. In the present paper, we a family of meshes M_h described as refinements of a given one in a unusual way. Indeed, this is formalised in terms of metrics. Any mesh M_h of this family is a unit mesh for a metric $\mathcal{M}_h$ which is proportional to a reference one $\mathcal{M}_1$:

$$\mathcal{M}_h = \frac{1}{h^2}\mathcal{M}_1 \ .$$

Further, we assume that h is large enough, in such a way that, as far as the normalized metric $\mathcal{M}_1$ is smooth enough, the variation of mesh size between two neihboring cells can be made as small as we wish. At element scale, the mesh is a uniform mesh. Lastly, for simplicity, we assume that we are able to build each unit mesh in such a way that it is made of isoscele triangles with symmetry axes aligned with the stretching direction, in order to avoid obtuses angles, see Fig. 1.

3. A priori finite-element analysis

A priori estimates have been derived very earlier, in $H^1(\Omega)$ ("projection property"), and in $L^2(\Omega)$ (Aubin-Nitsche analysis), but only by means of inequalities, and the leading term of the error is generally not exhibited (only bounds of it are proposed). In this section, we try to go a little further than the standard *a priori* analysis. We concentrate on the usual Poisson problem, set in a polyhedral n-dimensional domain Ω for the sake of simplicity:

$$-\Delta u \;=\; f \;\; on \;\; \Omega \;\; ; \;\; u = 0 \;\; on \;\; \partial\Omega. \tag{3}$$

Its variational form writes:

$$a(u,v) \;=\; \int_{\Omega} \nabla u . \nabla v \, \mathrm{d}\mathbf{x} \;=\; (\, f \, , \, v) \qquad \forall \, v \in V \tag{4}$$

where V holds again for the Sobolev space $V = H^1_0(\Omega) = \{u \in L^2(\Omega), \nabla u \in (L^2(\Omega))^n, u|_{\partial\Omega} = 0\}$. In order to derive an *a priori* estimate, we assume that the solution u has some *extra regularity*:

$$u \in \mathcal{V} = V \cap H^2(\Omega).$$

We observe that u is therefore the solution of:

$$a(u,v) \;=\; (\, f \, , \, v) \qquad \forall \, v \in V.$$

Let $\mathcal{T}_h$ be a mesh of Ω made of simplices, and let V_h be the subspace of V of continuous functions that are $\mathcal{P}_1$ on each element of the mesh. The discrete variational problem is thus defined by:

$$a(u_h, v_h) \;=\; (\, f \, , \, v_h) \qquad \forall \, v_h \in V_h \tag{5}$$

Let us introduce the *linear interpolation* Π_h from vertices values:

$$\Pi_h : \mathcal{V} \to V_h \;\; ; \;\; v \mapsto \Pi_h v \;\;\text{ such that }\;\; \Pi_h v|_T \text{ is affine } \forall \text{ element } T \;\;\text{ and }\;\; \Pi_h v|_i = v(\mathbf{x}_i) \forall i \text{ vertex }.$$

The approximation error in the sequel will be split into two components:

$$u_h \;-\; u \;=\; u_h \;-\; \Pi_h u \;+\; \Pi_h u \;-\; u \tag{6}$$

where we recognize in the first difference $\Pi_h u - u$ the *interpolation error*, while we shall refer to the first difference $E_h = u_h - \Pi_h u$ as the *implicit error*. It is useful to remark that the discrete statement is equivalently written:

$$a(u_h, \Pi_h \varphi) \;=\; (\, f \, , \, \Pi_h \varphi) \qquad \forall \, \varphi \in \mathcal{V}. \tag{7}$$

Combining with continuous and discrete systems we get :

$$a(\Pi_h u \;-\; u_h, \Pi_h \varphi) = a(\Pi_h u, \Pi_h \varphi) - a(u_h, \Pi_h \varphi) = a(\Pi_h u, \Pi_h \varphi) - (f, \Pi_h \varphi) = a(\Pi_h u, \Pi_h \varphi) - a(u, \Pi_h \varphi)$$

which gives:

$$a(E_h, \Pi_h \varphi) \;=\; a((u - \Pi_h u), \; \Pi_h \varphi) \qquad \forall \, \varphi \in \mathcal{V}. \tag{8}$$

3.1. 2D Truncation error analysis

We concentrate in this section on analysing the right-hand side of Estimation (8), that is:

$$< \nabla(u - \Pi_h u), \nabla \Pi_h \varphi > \tag{9}$$

We assume we are in the limit conditions of mesh convergence described in Sec. 2.

In such a extremely regular mesh, the main part of an approximation error can show compensations between two neighboring elements. For a smooth function u, the approximate gradient $\nabla \Pi_h u$ can be of first order accuracy on a given isoscele triangle $T_+ = ABC, CA = CB$, while some of second order convergence can be obtained on the union of this triangle with the triangle T_- symmetric with respect to the basis AB (cf. Figure 1).

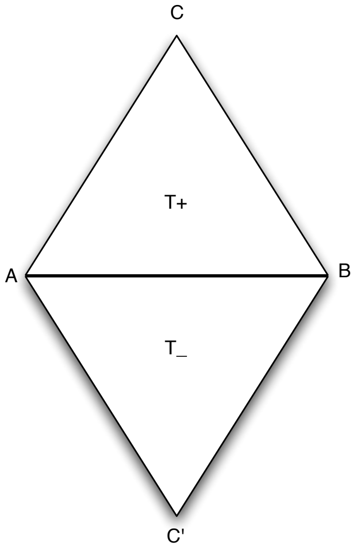


Figure 1: Superconvergent molec In case of stretching, the basis is smaller that the other sides.ule for a vertical derivative

Decomposing the previous Estimate (8) into an integral on each element leads to:

$$\begin{aligned} < \nabla(u - \Pi_h u), \nabla \Pi_h \varphi > &= \int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u) \frac{\partial}{\partial x} \Pi_h \varphi + \frac{\partial}{\partial y}(u - \Pi_h u) \frac{\partial}{\partial y} \Pi_h \varphi \right) \mathrm{d}\mathbf{x} \\ &= \sum_{T,\text{element}} \int_T \left(\frac{\partial}{\partial x}(u - \Pi_h u) \frac{\partial}{\partial x} \Pi_h \varphi + \frac{\partial}{\partial y}(u - \Pi_h u) \frac{\partial}{\partial y} \Pi_h \varphi \right) \mathrm{d}\mathbf{x} \end{aligned}$$

where the sum Σ is taken over any element T of the mesh. Restricting to the integral over T for $xIncaseofstretching, thebasisissmallerthat$ terms contribution and applying a Green formula we get:

$$\begin{gathered} \int_T \left(\frac{\partial}{\partial x}(u - \Pi_h u) \frac{\partial}{\partial x} \Pi_h \varphi \right) \mathrm{d}\mathbf{x} \;=\; \\ \int_{\partial T} (u - \Pi_h u)(\frac{\partial}{\partial x} \Pi_h \varphi) \cdot n_x^T \mathrm{d}\sigma \;-\; \int_T \left((u - \Pi_h u) \left(\frac{\partial^2}{\partial x^2} \Pi_h \varphi \right) \right) \mathrm{d}\mathbf{x} \end{gathered}$$

where by n_x^T we denoted the x component of the outward normal to the triangle $T : \mathbf{n}^T = (n_x^T, n_y^T)$. The same analysis holds for y terms. We observe that since $\Pi_h \varphi$ is $\mathcal{P}^1$ on T, the last integral in the right-hand side vanishes. Further, the derivative in the first integral is constant, hence:

$$\int_{\partial T} (u - \Pi_h u)(\frac{\partial}{\partial x} \Pi_h \varphi) \cdot n_x^T \mathrm{d}\sigma \;=\; \left(\frac{\partial}{\partial x} \Pi_h \varphi \right)|_T \int_{\partial T} (u - \Pi_h u) \cdot n_x^T \mathrm{d}\sigma.$$

Considering we are in the situation picturized in Figure 1, then the integral over ∂T applies to the two triangles T_+ and T_- with

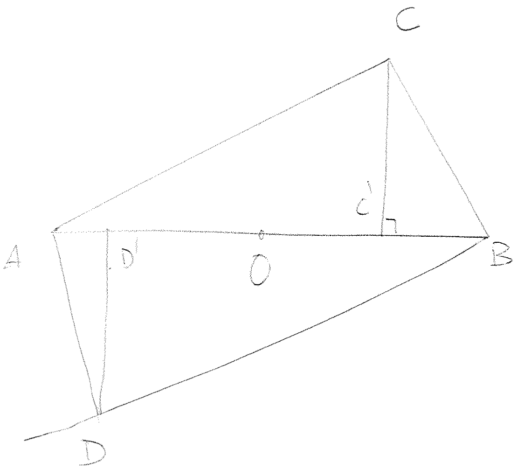


Figure 2: Molecule for a vertical derivative

edge e as a common edge. We observe that the sum of integrals along e provided by the two triangles gives:

$$\begin{aligned}&\left(\frac{\partial}{\partial x}\Pi_h\varphi\right)|_{T_+}\int_{\partial T_+\cap e}(u-\Pi_h u)\cdot n_x^{T_+}\mathrm{d}\sigma+\left(\frac{\partial}{\partial x}\Pi_h\varphi\right)|_{T_-}\int_{\partial T_-\cap e}(u-\Pi_h u)\cdot n_x^{T_-}\mathrm{d}\sigma\\&=(\frac{\partial}{\partial x}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial x}\Pi_h\varphi|_{T_-})\int_{\partial T_+\cap e}(u-\Pi_h u)\cdot n_x^{T_+}\mathrm{d}\sigma\ .\end{aligned}$$

Remembering that $\Pi_h\varphi$ is continuous along any edge e, we can write:

$$\begin{aligned}&(\frac{\partial}{\partial x}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial x}\Pi_h\varphi|_{T_-})\cdot n_x^{T_+}\int_{\partial T_+\cap e}(u-\Pi_h u)\mathrm{d}\sigma\ =\\&\qquad(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})\int_{\partial T_+\cap e}(u-\Pi_h u)\mathrm{d}\sigma\ .\end{aligned}\tag{10}$$

In the case of a regular mesh as in Figure 1, the two terms of the difference $(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})$ are derivatives evaluated at mid-altitudes:

Lemma 3.1. *For a regular enough mesh, the term* $\eta^{-1}\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)$ *is consistent with a second normal derivative weighted by the inverse of mean altitude of the two triangles:*

$$\eta^{-1}\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+}-\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)=...$$

□

Proof:

As explained in Sec.”Mesh Convergence”, we consider that the mesh is locally made of identical isoscele triangles, with an altitude aligned with symmetry axis of length η and with basis (orthogonal to axis of symmetry) of length ξ. In case of stretching, the basis is smaller that the other sides. A couple of neighboring triangles may have a common basis (Fig.1) or a common lateral side (Fig.2).

We analyse the second case, remarking that it somewhat involves the first one.

We first analyse how behave our geometry when the stretching, which we can represent by the quotient ξ/η, is growing, *i.e.* $\xi/\eta \to +\infty$. Let us denote α the small angle between the two equal sides, which therefore tend to zero. Let us introduce:

$$\theta = (x_B - x_C)/\xi = (x_D - x_A)/\xi$$

We observe that:

$$\begin{aligned} sin\alpha &= \eta/\xi \\ cos\alpha &= |AC'|/\xi = |BD'|/\xi = 1 - \theta \end{aligned}$$

which shows that when stretching ξ/η tend to infinity, we have:

$$\theta = \frac{\eta^2}{4\xi^2} + O(\frac{\eta^4}{\xi^4})$$

where $O(\frac{\eta^4}{\xi^4})/(\frac{\eta^4}{\xi^4})$ is bounded.

Let us come back to the derivatives under study. It is useful to assume that the origin of axes is the center O of AB, and that the horizontal axis is AB. We have:

$$x_C = \frac{1}{2}(1 - 2\theta)\xi \quad ; \quad x_D = \frac{1}{2}(2\theta - 1)\xi.$$

For high stretching and case of Fig.2, θ is small. Conversely, the case of Fig.1 gives $\theta = 1/2$. At nodes A, B, C, D, the interpolation $\Pi_h\varphi$ is identical to function φ:

$$\begin{aligned} \varphi_A &= \Pi_h\varphi(A) = \varphi(0) - \frac{1}{2}\xi\varphi_x + \frac{1}{8}\xi^2\varphi_{xx} + ... \\ \varphi_B &= \Pi_h\varphi(B) = \varphi(0) + \frac{1}{2}\xi\varphi_x + \frac{1}{8}\xi^2\varphi_{xx} + ... \\ \varphi_C &= \Pi_h\varphi(C) = \varphi(0) + \frac{1}{2}(1 - 2\theta)\xi\varphi_x + \eta\varphi_y + \frac{1}{2}\left(\frac{1}{4}(1 - 2\theta)^2\xi^2\varphi_{xx} + 2\frac{1}{2}(1 - 2\theta)\xi\eta\varphi_{xy} + \eta^2\varphi_{yy}\right) + ... \\ \varphi_D &= \Pi_h\varphi(D) = \varphi(0) - \frac{1}{2}(1 - 2\theta)\xi\varphi_x - \eta\varphi_y + \frac{1}{2}\left(\frac{1}{4}(1 - 2\theta)^2\xi^2\varphi_{xx} + 2\frac{1}{2}(1 - 2\theta)\xi\eta\varphi_{xy} + \eta^2\varphi_{yy}\right) + ... \end{aligned}$$

Denoting by C' and D' the feet of altitudes from (resp.) C and D, we observe that:

$$\begin{aligned} \varphi_{C'} &= \Pi_h\varphi(C') = \theta\varphi_A + (1 - \theta)\varphi_B = \varphi(0) - \frac{1}{2}(1 - 2\theta)\xi\varphi_x + \xi^2/8\varphi_{xx} + ... \\ \varphi_{D'} &= \Pi_h\varphi(D') = \theta\varphi_B + (1 - \theta)\varphi_A = \varphi(0) + \frac{1}{2}(1 - 2\theta)\xi\varphi_x + \xi^2/8\varphi_{xx} + ... \end{aligned}$$

The non-divided differences defining the normal derivatives write:

$$\begin{aligned} \varphi_C - \varphi_{C'} &= (1 - 2\theta)\xi\varphi_x + \eta\varphi_y + \left(\frac{1}{8}[(1 - 2\theta)^2 - 1]\xi^2\varphi_{xx} + \frac{1}{2}(1 - 2\theta)\xi\eta\varphi_{xy} + \frac{1}{2}\eta^2\varphi_{yy}\right) + ... \\ \varphi_{D'} - \varphi_D &= (1 - 2\theta)\xi\varphi_x + \eta\varphi_y - \left(\frac{1}{8}[(1 - 2\theta)^2 - 1]\xi^2\varphi_{xx} + \frac{1}{2}(1 - 2\theta)\xi\eta\varphi_{xy} + \frac{1}{2}\eta^2\varphi_{yy}\right) + ... \end{aligned}$$

Remembering that $\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right) = \frac{1}{\eta}\left(\varphi_C - \varphi_{C'} - \varphi_{D'} + \varphi_D\right)$, we get:

$$\eta^{-1}\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right) = \eta^{-2}\left(\frac{1}{4}[(1 - 2\theta)^2 - 1]\xi^2\varphi_{xx} + (1 - 2\theta)\xi\eta\varphi_{xy} + \eta^2\varphi_{yy}\right).$$

If we are in the case of Fig.2: then $\xi/\eta \to +\infty$, but:

$$\begin{aligned} [(1 - 2\theta)^2 - 1] &= 4\theta + 4\theta^2 = \frac{\eta^2}{\xi^2} + O(\frac{\eta^4}{\xi^4}) \\ (1 - 2\theta) &= 1 + O(\frac{\eta^2}{\xi^2}) \end{aligned}$$

$$|\eta^{-1}\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)| \leq \frac{\xi}{\eta}|\varphi_{xy}| + \frac{1}{4}|\varphi_{xx}| + |\varphi_{yy}| + O(\frac{\eta^2}{\xi^2}).$$

If we are in the case of Fig.1: then $\eta > \xi\sqrt{3}/2$ or $\xi^2/\eta^2 \leq 4/3$ and $\theta = 1/2$:

$$|\eta^{-1}\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)| \leq |\eta^{-2}\left(\frac{-1}{4}\xi^2\varphi_{xx} + \eta^2\varphi_{yy}\right)| \leq \eta^{-2}\left(\frac{1}{3}|\varphi_{xx}| + |\varphi_{yy}|\right).$$

Ancienne version

For the proof of this result we refer to Figure 1. Suppose we denote I the middle of segment $[AB]$ (also denoted edge e), then $\varphi(I) = (\varphi(A) + \varphi(B))/2$. Next, the difference $\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right)$ can be expressed by divided difference as:

$$\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-} = \frac{(y_I - y_{C'})(\varphi(C) - \varphi(I)) + (y_C - y_I)(\varphi(C') - \varphi(I))}{(y_C - y_I)(y_I - y_{C'})}.$$

By hypothesis the mesh is regular enough such that the two triangles (ABC) and (ABC') can be considered as isosceles, thus we assume that: $y_C - y_I = y_I - y_{C'} = h$. Replacing with h where possible and after further computation we get for the right-hand side of the previous relation:

$$\frac{(y_I - y_{C'})(\varphi(C) - \varphi(I)) + (y_C - y_I)(\varphi(C') - \varphi(I))}{(y_C - y_I)(y_I - y_{C'})} = h \cdot \frac{\varphi(C) - 2\varphi(I) + \varphi(C')}{h^2},$$

where we recognize an estimation of the second order derivative from finite difference theory.
To conclude, the following estimate holds:

$$\left(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-}\right) \approx h.\frac{\partial^2\varphi}{\partial y^2}(I)$$

In practice the jump for T_+ and T_- could differ, that is why we consider the mean jump $h = \frac{a_+ + a_-}{2}$, where by a_+ we denoted the altitude of T^+ triangle and respectively a_- the altitude of the T^- triangle.

□

Regarding y-terms contribution, the previous estimate holds too.

To synthetize, we have shown until here that:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi\right) d\mathbf{x} = \sum_{e,\text{edge}} [\nabla\Pi_h\varphi \cdot n]_e \int_e (u - \Pi_h u)_e \, de \tag{11}$$

where according to the previous Lemma 3.1 the jump $[\nabla\Pi_h\varphi \cdot n]_e$ is identified as a second order derivative:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi\right) d\mathbf{x} \approx \sum_{e,\text{edge}} h \cdot \nabla^2\varphi \int_e (u - \Pi_h u)_e \, de \tag{12}$$

Transformation of Estimate (12) into an integral on Ω. Once we have identify a second derivative, we can examine its weight. Indeed, it is multiplied by half the sum of the two altitudes of the triangles and integrated along the common edge (see proof of previous Lemma 3.1). Let us denote by G_+ (resp. G_-) the centroid of triangle T_+ (resp. T_-), a_+ (resp. a_-) the altitude of T_+ (resp T_-) orthogonal with respect to e. We shall refer in the sequel to the diamond-shaped surface D_e as the surface bounded

Figure 3: Diamont shape geometry : (AG_+BG_-)

by the segments joining either G_+ or G_- to an extremity of edge e (see Figure 3). The triangle formed by G_+ and e (or ABG_+), denoted here K_+, has an area of one third T_+'s area [1]:

$$|T_+| = \frac{1}{2} \cdot |e| \cdot |a_+| = 3 \cdot |K_+|.$$

Then, the sum of K^+ and K^- areas is equal to the area of the diamond-shaped surface D_e:

$$|D_e| = |K^+| + |K^-| = |e|\frac{|a_+| + |a_-|}{6} \ .$$

Let us approximate now the integral over edge e of the interpolation error $u - \Pi_h u$ from Estimate (12) with the integral over the diamond D_e for the same expression :

$$\frac{1}{|e|}\int_e (u - \Pi_h u) \ \mathrm{d}e \approx \frac{1}{|D_e|}\int_{D_e} (u - \Pi_h u) \ \mathrm{d}\mathbf{x}.$$

We observe that the union of diamond cells covers the whole computational domain Ω, *i.e.* $|\Omega| = \sum_{K\in\mathcal{H}} |K| = \sum_{D_e} |D_e|$. This allows to estimate the integral on Ω.

$$\begin{aligned}\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi\right)\mathrm{d}\mathbf{x} \ \approx \\ \sum_{D_e} 3\int_{D_e} (u - \Pi_h u)\nabla^2\varphi(\mathbf{x}) \ \mathrm{d}\mathbf{x} = 3\int_\Omega (u - \Pi_h u)\nabla^2\varphi(\mathbf{x}) \ \mathrm{d}\mathbf{x}\end{aligned} \qquad (13)$$

A first estimate. If we try to write an estimate which does not depend too much on function φ, we can over-estimate the error as follows:

Lemma 3.2. *We have the following bound:*

$$\begin{aligned}|\int_\Omega (\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi)d\mathbf{x}| \ \leq \\ 3\int_\Omega |\rho(H(\varphi)| \ |u - \Pi_h u| d\mathbf{x}\end{aligned} \qquad (14)$$

where $A \leq B$ holds for a majoration asymptotically valid, i.e. $A \leq B + O(A)$. Expression $|\rho(H(\varphi))|$ holds for the largest (in absolute value) eigenvalue of the Hessian $H(\varphi)$.□

3.2. 3D Truncation error analysis

The previous steps from two dimensional analysis applies here too. Thus, we can skip the calculations and go directly to the main result and say:

Lemma 3.3. *For a regular enough mesh, the term $(\frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_+} - \frac{\partial}{\partial n_{T_+}}\Pi_h\varphi|_{T_-})$ is consistent with a second normal derivative weighted by the inverse of mean altitude of the two tetrahedra T_+ and T_-.* □

To synthetize, as in two dimensional analysis, the following estimate holds:

$$\int_\Omega (\nabla(u - \Pi_h u)\nabla\Pi_h\varphi) \ \mathrm{d}\Omega \ \approx \sum_{f,\text{face}} [\nabla\Pi_h\varphi \cdot n]_f \int_f (u - \Pi_h u)_f \ \mathrm{d}\sigma \ . \qquad (15)$$

where the jump $\nabla\Pi_h\varphi \cdot n$ is estimated as a second order derivative, weighted by half the sum of corresponding altitudes of tetrahedras T_+ and T_-.

[1] $|\cdot|$ holds for volume, area or length of geometrical objects

Transformation of Estimate (15) into an integral on Ω.

Let us denote by G_+ (resp. G_-) the centroid of tetrahedron T_+ (resp. T_-), a_+ (resp. a_-) the altitude of T_+ (resp T_-) with respect to f. The volume of the tetrahedron K^+ formed by G_+ and f is equal to:

$$|K^+| = \frac{1}{9}|f| \cdot |a_+|.$$

As for two dimensional case, we construct the diamond-shaped volume D_f bounded by the triangular plans joining either G_+ or G_- to a side of face f. Then, the volume of D_f is :

$$|D_f| = |K^+| + |K^-| = |f|\frac{|a_+| + |a_-|}{9}.$$

We shall approximate the integral over face f with the integral over the diamond D_f. Then again, the union of diamond cells covers the whole computational domain Ω, *i.e.* $|\Omega| = \sum |D_f|$. This allows to estimate the integral on Ω as:

$$\int_\Omega \left(\frac{\partial}{\partial x}(u - \Pi_h u)\frac{\partial}{\partial x}\Pi_h\varphi + \frac{\partial}{\partial y}(u - \Pi_h u)\frac{\partial}{\partial y}\Pi_h\varphi \right) \mathbf{dx} \;\approx$$
$$\sum_{D_e} \frac{9}{2} \int_{D_e} (u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\, \mathbf{dx} = \frac{9}{2} \int_\Omega (u - \Pi_h u)\nabla^2\varphi(\mathbf{x})\, \mathbf{dx} \tag{16}$$

A first $3D$ **estimate.** If we go furthermore and search for an estimate of (16) which does not depend too much on the function φ, we can over-estimate the error according to the following lemma:

Lemma 3.4. *We have the following bound:*

$$|\int_\Omega (\nabla(u - \Pi_h u)\nabla\Pi_h\varphi) d\Omega| \;\lesssim\; \frac{9}{2} \int_\Omega |\rho(H(\varphi))|\; |u - \Pi_h u| d\Omega \tag{17}$$

where $A \lesssim B$ holds for a majoration asymptotically valid, i.e. $A \leq B + O(A)$. Expression $|\rho(H(\varphi))|$ holds for the largest (in absolute value) eigenvalue of the Hessian $H(\varphi)$.□

Remark: In the case where $u - \Pi_h u$ does not vanish on the domain boundary denoted here Γ, then we get an extra term equivalent to:

$$\int_\Gamma (u - \Pi_h u)\nabla\varphi \cdot \mathbf{n}\, \mathrm{d}\sigma.$$

4. Unsteady Navier-Stokes Models

4.1. Continuous state system

Continuous state system. The 3D unsteady compressible Navier-Stokes system for a perfect gas is set for $(\mathbf{x}, t)$ in the computational space-time domain $Q = \Omega \times [0, T]$, where T is the (positive) maximal time and $\Omega \subset \mathbb{R}^3$ is the spatial domain. It writes :

$$\frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}^E + \nabla \cdot \mathcal{F}^v = 0$$

In the above definition, W is the vector of conservative flow variables. The Euler fluxes are defined by: $\mathcal{F}^E(W) = (\mathcal{F}_1^E(W), \mathcal{F}_2^E(W), \mathcal{F}_3^E($
The column vector W and flux tensor $\mathcal{F}^E$ are given by

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} \quad ; \quad \mathcal{F}^E(W) = \begin{pmatrix} \rho\mathbf{u} \\ \rho\mathbf{u}u + p\mathbf{e}_x \\ \rho\mathbf{u}v + p\mathbf{e}_y \\ \rho\mathbf{u}w + p\mathbf{e}_z \\ \rho\mathbf{u}H \end{pmatrix} . \tag{18}$$

Here ρ, p, and E represent the fluid density, thermodynamic pressure, and total energy per unit mass. u, v, and w are the Cartesian components of the velocity vector $\mathbf{u}$ and H is the total enthalpy given by $H = E + \frac{p}{\rho}$.
Functions φ and W have 5 components, and therefore the product φW holds for $\sum_{k=1..5} \varphi_k W_k$. We have denoted by Γ the inviscid

subset of the boundary of the computational domain Ω, $\mathbf{n}$ is the outward normal to Γ, $W(0)(\mathbf{x}) = W(\mathbf{x}, t)|_{t=0}$ for any $\mathbf{x}$ in Ω, W_0 the initial condition and the boundary flux $\hat{\mathcal{F}}^E$ contains the different non-viscous boundary conditions, which involve inflow, outflow and slip boundary conditions.
We describe in short the viscous fluxes as:

$$\mathcal{F}^v \quad = \quad [0, \sigma, -(\mathbf{q} - \mathbf{u}.\sigma)]^T,$$

where $\mathbf{u} = (u_1, u_2, u_3)$ is the velocity vector and the viscous stress tensor σ is defined as:

$$\sigma \quad = \quad \mu(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}) - \frac{2}{3}\mu\nabla.\mathbf{uI},$$

with μ representing the constant viscosity.
The heat flux $\mathbf{q}$ is given by Fourier's law:

$$\mathbf{q} \quad = \quad -\lambda\nabla T$$

where λ is the heat conduction (assumed here to be constant), and T the temperature defined hereafter:

$$T = \frac{1}{c_v}\left(E - \frac{1}{2\rho}((\rho u_1)^2 + (\rho u_2)^2 + (\rho u_3)^2)\right),$$

with c_v also assumed to be constant.
The matrix expression of viscous fluxes $\mathcal{V}$ writes:

$$\left(\begin{array}{ccc} 0 & 0 & 0 \\ \sigma_{xx} & \sigma_{yx} & \sigma_{zx} \\ \sigma_{xy} & \sigma_{yy} & \sigma_{zy} \\ \sigma_{xz} & \sigma_{yz} & \sigma_{zz} \\ u.\sigma_{xx} + v.\sigma_{xy} + w.\sigma_{xz} + \lambda.\nabla T & u.\sigma_{yx} + v.\sigma_{yy} + w.\sigma_{yz} + \lambda.\nabla T & u.\sigma_{zx} + v.\sigma_{zy} + w.\sigma_{zz} + \lambda.\nabla T \end{array}\right),$$

And viscous stress σ has the following general matrix expression:

$$\sigma = \mu\left[\underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} & \frac{\partial u}{\partial z} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} & \frac{\partial v}{\partial z} \\ \frac{\partial w}{\partial x} & \frac{\partial w}{\partial y} & \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla\mathbf{u}} + \underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} & \frac{\partial v}{\partial x} & \frac{\partial w}{\partial x} \\ \frac{\partial u}{\partial y} & \frac{\partial v}{\partial y} & \frac{\partial w}{\partial y} \\ \frac{\partial u}{\partial z} & \frac{\partial v}{\partial z} & \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla\mathbf{u}^T}\right]$$

$$-\frac{2}{3}\mu\underbrace{\begin{pmatrix} \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} & 0 & 0 \\ 0 & \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} & 0 \\ 0 & 0 & \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \end{pmatrix}}_{\nabla.\mathbf{uI}}$$

An essential ingredient of our discretization and of our analysis is the elementwise linear interpolation operator. In order to use it easily, we define our working functional space as $V = \left[H^1(\Omega) \cap C(\bar{\Omega})\right]^5$, that is the set of measurable functions that are continuous with square integrable gradient. We formulate the Navier-Stokes model in a compact variational formulation in the functional space $\mathcal{V} = H^1\{[0, T]; V\}$ as follows:

$$\begin{aligned} &\text{Find } W \in \mathcal{V} \text{ such that } \forall\varphi \in V, \quad (\Psi(W), \varphi) = 0 \\ &\text{with } \Psi \quad = \quad \Psi^t + \Psi^E + \Psi^v, \\ \left(\Psi^t(W) + \Phi^E(W), \varphi\right) \quad &= \quad \int_\Omega \varphi(0)(W_0 - W(0))\,\mathrm{d}\Omega \; + \; \int_0^T\int_\Omega \varphi W_t\,\mathrm{d}\Omega\,\mathrm{d}t \\ &\quad + \int_0^T\int_\Omega \varphi\,\nabla\cdot\mathcal{F}^E(W)\,\mathrm{d}\Omega\,\mathrm{d}t \; - \; \int_0^T\int_\Gamma \varphi\hat{\mathcal{F}}^E(W).\mathbf{n}\,\mathrm{d}\Gamma\,\mathrm{d}t \\ (\Psi^v(W), \varphi) \quad &= \quad \int_0^T\int_\Omega \varphi\,\nabla\cdot\mathcal{F}^v(W)\,\mathrm{d}\Omega\,\mathrm{d}t\,. \end{aligned} \qquad (19)$$

Viscous fluxes provide seven new terms to which we apply the finite-element error-analysis of Section 3, that is based on estimations of interpolation errors:

$$(\Psi^{\mathcal{V}}, \psi) = \int_\Omega \psi\nabla\cdot\mathcal{F}^v\,\mathrm{d}\Omega = \sum_{k=1}^{7} E_k\,.$$

The three first terms come from moment equations and depend only on $\psi_{234} = (\psi_2, \psi_3, \psi_4)^T$:

$$\begin{aligned} E_1 &= \int_\Omega \psi_{234} \nabla \cdot \mu \nabla \mathbf{u} \, \mathrm{d}\Omega \\ E_2 &= \int_\Omega \psi_{234} \nabla \cdot \mu (\nabla \mathbf{u})^T \mathrm{d}\Omega \\ E_3 &= -\frac{2}{3} \int_\Omega \psi_{234} \nabla \cdot \mu \nabla . \mathbf{u} \mathbf{I} \, \mathrm{d}\Omega. \end{aligned}$$

The four last terms are derived from the energy equation:

$$\begin{aligned} E_4 &= \int_\Omega \psi_5 \nabla \cdot \lambda \nabla T \mathrm{d}\Omega \\ E_5 &= \int_\Omega \psi_5 \nabla \cdot (\mathbf{u} . \mu \nabla \mathbf{u}) \, \mathrm{d}\Omega \\ E_6 &= \int_\Omega \psi_5 \nabla \cdot \left(\mathbf{u} . \mu (\nabla \mathbf{u})^T \right) \mathrm{d}\Omega \\ E_7 &= -\frac{2}{3} \int_\Omega \psi_5 \nabla \cdot (\mathbf{u} . \mu \nabla . \mathbf{u} \mathbf{I}) \, \mathrm{d}\Omega \, . \end{aligned}$$

4.2. Variational discrete formulation

As a spatially semi-discrete model, we consider the Mixed-Element-Volume formulation [16]. As in [32], we reformulate it under the form of a finite element variational formulation, this time in the unsteady context. We assume that Ω is covered by a finite-element partition in simplicial elements denoted K. The mesh, denoted by $\mathcal{H}$ is the set of the elements. Let us introduce the following approximation space:

$$V_h = \left\{ \varphi_h \in V \, \middle| \, \varphi_{h|K} \text{ is affine } \forall K \in \mathcal{H} \right\},$$

Let Π_h be the usual $\mathcal{P}^1$ projector:

$$\Pi_h : \; V \rightarrow \; V_h \; \text{ such that } \; \Pi_h \varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i), \; \forall \; \mathbf{x}_i \text{ vertex of } \mathcal{H}.$$

We extend it to time-dependent functions:

$$\Pi_h : \; H^1\{[0,T]; V\} \rightarrow \; H^1\{[0,T]; V_h\} \; \text{ such that } \; (\Pi_h \varphi)(t) = \Pi_h(\varphi(t)), \; \forall \, t \in \; [0,T].$$

The weak discrete formulation writes:

$$\begin{aligned} &\text{Find } W_h \in H^1\{[0,T]; V_h\} \text{ such that } \forall \varphi_h \in H^1\{[0,T]; V_h\}, \quad (\Psi_h(W_h), \varphi_h) = 0, \\ \text{with: } (\Psi_h(W_h), \varphi_h) &= \int_\Omega \varphi_h(0)(\Pi_h W_h(0) - W_{0h}) \, \mathrm{d}\Omega + \int_0^T \int_\Omega \varphi_h \, \Pi_h W_{h,t} \, \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \int_0^T \int_\Omega \varphi_h \nabla \cdot \mathcal{F}_h(W_h) \, \mathrm{d}\Omega \, \mathrm{d}t - \int_0^T \int_\Gamma \varphi_h \hat{\mathcal{F}}_h(W_h) . \mathbf{n} \, \mathrm{d}\Gamma \, \mathrm{d}t + \int_0^T \int_\Omega \varphi_h \, D_h(W_h) \, \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \int_0^T \int_\Omega \varphi_h \nabla \cdot \mathcal{F}_h^v(W_h) \, \mathrm{d}\Omega \, \mathrm{d}t \, , \end{aligned} \tag{20}$$

In this system, the discretization of Euler terms are essentially applied by projecting the fluxes with the P^1 interpolator: $\mathcal{F}_h^E = \Pi_h \mathcal{F}^E$ and $\hat{\mathcal{F}}_h^E = \Pi_h \hat{\mathcal{F}}^E$.A numerical diffusion - term D_h- is added for numerical stability. In short, the D_h term involves the difference between the Galerkin central-differences approximation and a second-order Godunov approximation defined as in [16]. In the present study, we only need to know that for smooth fields, the D_h term is a third order term with respect to the mesh size parameter h.

Notation $\mathcal{F}_h^v(W_h)$ holds for a standard P^1 finite element discretization of the viscous terms, in which only the numerical quadrature of non-differentiated fields need be specified, which will be done for each term at the same time as we perform our error analysis.

Since we do not address time discretization errors, we keep the time derivation not-discretized. In the numerical examples, an explicit time stepping algorithm is used by means of a multi-stage, high-order strong-stability-preserving (SSP) Runge-Kutta scheme. More details can be found in [5].

4.3. *Mesh adaptation first problem statement*

Let g be a function of $L^2\{[0,T];V\}$. We assume that the purpose of the simulation problem is to evaluate the functional:

$$j = (g, W) \quad \text{where} \quad W \ \text{ is the solution of (19).}$$

The problem addressed in this paper is, given a number of nodes N, to *find the mesh which minimizes the following functional error:*

$$\delta j = (g, W - W_h) \quad \textit{where} \quad W \ \textit{ is the solution of (19)} \quad \textit{and} \quad W_h \ \textit{ is the solution of (20).}$$

In order to make this problem more precise we propose in this section an error analysis.

5. Linearised error system

Let ψ a smooth test function; we choose:

$$\varphi = \Pi_h\psi \quad \text{in} \quad (19) \qquad \text{and} \qquad \varphi_h = \Pi_h\psi \quad \text{in} \quad (20).$$

Then:

$$(\Psi_h(W_h)\,,\,\Pi_h\psi) - (\Psi_h(\Pi_h W)\,,\,\Pi_h\psi) = -\,(\Psi_h(\Pi_h W)\,,\,\Pi_h\psi) + (\Psi(W)\,,\,\Pi_h\psi)$$

Here we need to assume that W_h can be made close enough to $\Pi_h W$ in such a way that we can identify the main term of the left hand side as a Jacobian times the difference:

$$\left(\frac{\partial \Psi_h}{\partial W}(W_h)(W_h - \Pi_h W)\,,\,\Pi_h\psi\right) \approx -\,(\Psi_h(\Pi_h W)\,,\,\Pi_h\psi) + (\Psi(W)\,,\,\Pi_h\psi) \quad \text{as} \quad h \to 0.$$

We are now interested by the right-hand term, made of Euler terms, a stabilization term which we neglect (due to smoothness of functions W and ψ, and viscous terms.

6. Inviscid right-hand side

6.1. *Interpolation errors*

The right-hand inviscid side writes:

$$RHS \;=\; \int_\Omega \phi_h \nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi_h(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W)) \cdot \mathbf{n}\,\mathrm{d}\Gamma$$

we recall that $\phi_h = \Pi_h\phi$ and next we add and substract a ϕ term:

$$RHS \;=\; RHS_1 + RHS_2$$

with:

$$\begin{aligned} RHS_1 \;=\; & \int_\Omega (\Pi_h\phi - \phi)\nabla \cdot (\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega \\ & - \int_\Gamma (\Pi_h\phi - \phi)(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W)) \cdot \mathbf{n}\,\mathrm{d}\Gamma\,. \end{aligned}$$

Assuming smoothness of ϕ and $\mathcal{F}(W)$, we deduce that on Ω, interpolation errors are of order two and their gradients are of order one, same on boundary, and RHS_1 is thus of order three:

$$RHS_1 \;\leq\; \text{const}.h^3.$$

The second term writes:

$$RHS_2 \;=\; \int_\Omega \phi\nabla \cdot (\mathcal{F}(W) - \Pi_h\mathcal{F}(W))\mathrm{d}\Omega - \int_\Gamma \phi(\bar{\mathcal{F}}^{out}(W) - \Pi_h\bar{\mathcal{F}}^{out}(W)) \cdot \mathbf{n}\,\mathrm{d}\Gamma$$

and we transform it as follows:

$$\begin{aligned} RHS_2 &= -\int_\Omega (\nabla\phi)\cdot(\mathcal{F}(W)-\Pi_h\mathcal{F}(W))\mathrm{d}\Omega \\ &+ \int_\Gamma \phi(\mathcal{F}(W)-\Pi_h\mathcal{F}(W))\cdot\mathbf{n}\,\mathrm{d}\Gamma \\ &- \int_\Gamma \phi(\bar{\mathcal{F}}^{out}(W)-\Pi_h\bar{\mathcal{F}}^{out}(W))\cdot\mathbf{n}\,\mathrm{d}\Gamma. \end{aligned}$$

The above estimates shows again the central role of the **interpolation error on internal and boundary fluxes** for the global approximation error.

Remark: In RHS_2 we can apply the same asymptotic extension as in the elliptic case studied in previous section. The expression of RHS_2 is in fact very good news. Indeed, due to the smoothness assumptions for ϕ and W, L^2 estimates for interpolation error on volume and on boundary apply, so that this term appears as a second-order one:

$$RHS_2 \;\leq\; \text{const}.h^2.$$

Further, using the same techniques as in [**?**], this term can be extended as follows:

$$RHS_2 \;=\; h^2\;(G(W,m),\phi)\;+\;R$$

where the last parenthesis is to be understood as a distribution one. The term R is of higher order:

$$R \;=\; o(h^2).\;\square$$

6.2. *Temporary conclusion*

The above study shows that the implicit error $W_h - \Pi_h W$ is essentially a function of the interpolation error $W - \Pi_h W$. In the numerical applications we shall discard the boundary terms in order to simplify the mesh generation. In [28], it has been observed that this simplification does not reduce much the quality of the results.

7. Viscous right-hand side

7.1. *Notations*

As noted below, viscous fluxes provide seven E_k terms to which we apply the finite-element error-analysis of Section 3, that is based on estimations of interpolation errors:

$$(\Phi^{\mathcal{V}},\psi) = \int_\Omega \psi\nabla\cdot\mathcal{V}\,\mathrm{d}\Omega = \sum_{k=1}^{7} E_k\,.$$

The three first terms come from moment equations and depend only on $\psi_{234} = (\psi_2,\psi_3,\psi_4)^T$:

$$\begin{aligned} E_1 &= \int_\Omega \psi_{234}\nabla\cdot\mu\nabla\mathbf{u}\,\mathrm{d}\Omega \\ E_2 &= \int_\Omega \psi_{234}\nabla\cdot\mu(\nabla\mathbf{u})^T\mathrm{d}\Omega \\ E_3 &= -\frac{2}{3}\int_\Omega \psi_{234}\nabla\cdot\mu\nabla.\mathbf{u}\mathbf{I}\,\mathrm{d}\Omega. \end{aligned}$$

The four last terms are derived from the energy equation:

$$\begin{aligned} E_4 &= \int_\Omega \psi_5\nabla\cdot\lambda\nabla T\mathrm{d}\Omega \\ E_5 &= \int_\Omega \psi_5\nabla\cdot(\mathbf{u}.\mu\nabla\mathbf{u})\,\mathrm{d}\Omega \\ E_6 &= \int_\Omega \psi_5\nabla\cdot\left(\mathbf{u}.\mu(\nabla\mathbf{u})^T\right)\mathrm{d}\Omega \\ E_7 &= -\frac{2}{3}\int_\Omega \psi_5\nabla\cdot(\mathbf{u}.\mu\nabla.\mathbf{u}\mathbf{I})\,\mathrm{d}\Omega\,. \end{aligned}$$

In the sequel, for each of these seven terms, we derive an error estimator following the *a priori* finite-element analysis of previous sections where we have shown how the implicit error is bounded by interpolation error weighted by some weights. The main idea of the resolution is to retrieve Relation (9) for which Lemma 3.4 can be applied.

7.2. Study of truncation error terms

As for the Euler flows case we do not consider in the sequel the boundary error terms.

Study of E_1, E_2 and E_3

. Let us concentrate first on the three terms summation derived from the momentum equation:

$$\begin{array}{rclcl}
E_1 & = & \mu \int_\Omega \psi_{234} \nabla \cdot \nabla \mathbf{u} d\Omega & = & \mu \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j} \left(\frac{\partial}{\partial x_j} u_i \right) d\Omega, \\
E_2 & = & \mu \int_\Omega \psi_{234} \nabla \cdot (\nabla \mathbf{u})^T d\Omega & = & \mu \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j} \left(\frac{\partial}{\partial x_i} u_j \right) d\Omega, \\
E_3 & = & -\mu \frac{2}{3} \int_\Omega \psi_{234} \nabla \cdot (\nabla.\mathbf{uI}) d\Omega & = & -\mu \frac{2}{3} \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_j} u_j \right) d\Omega.
\end{array}$$

We remark that E_2 and E_3 expressions can be directly additioned with an exchange of i and j derivatives. Then the summation of these terms writes:

$$E_1 + E_2 + E_3 = \mu \left(\sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_j} \left(\frac{\partial}{\partial x_j} u_i \right) d\Omega \; + \; \frac{1}{3} \sum_{i=1}^3 \sum_{j=1}^3 \int_\Omega \psi_{i+1} \frac{\partial}{\partial x_i} \left(\frac{\partial}{\partial x_j} u_j \right) d\Omega \right).$$

It is sufficient to consider e_{123} :

$$e_{123} = \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 u_l}{\partial x_i \partial x_j} d\Omega.$$

The other terms are depicted identically.
We are interested in the error term:

$$\begin{aligned}
\delta e_{123} = e_{123} - e_{123}^h \; &= \; \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 u_l}{\partial x_i \partial x_j} d\Omega - \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2 \Pi_h u_l}{\partial x_i \partial x_j} d\Omega \\
&= \int_\Omega \Pi_h \psi_{i+1} \frac{\partial^2}{\partial x_i \partial x_j} (u_l - \Pi_h u_l) d\Omega.
\end{aligned}$$

After a first integration by part and neglecting the boundary terms, the error term writes:

$$\delta e_{123} = - \int_\Omega \frac{\partial}{\partial x_i} \left(\Pi_h \psi_{i+1} \right) \frac{\partial}{\partial x_j} (u_l - \Pi_h u_l) \, d\Omega.$$

Now, according to Lemma 3.4 from the elliptic error analysis these volumic contribution is overestimated as:

$$\delta e_{123} \leq \frac{9}{2} \int_\Omega \rho(H(\psi_{i+1})) |u_l - \Pi_h u_l|.$$

And finally, going back to our initial sommation the following *a priori* estimate holds for the first three terms of viscous flux contribution:

$$\delta E_1 + \delta E_2 + \delta E_3 \leq \frac{9}{2} \mu \sum_{i=1}^3 \left(3 \, \rho(H(\psi_{i+1})) + \frac{1}{3} \sum_{j=1}^3 \rho(H(\psi_{i+1})) \right) |u_i - \Pi_h u_i|.$$

We recall that $9/2$ constant comes from the three dimensional estimator, and since the first term of the summation is independent on j, it is thus multiplied by 3.

Study of E_4

. The first term from the energy equation is discussed now:

$$E_4 \; = \; \int_\Omega \psi_5 \nabla \cdot \lambda \nabla T \mathrm{d}\Omega.$$

We consider the following discretisation :

$$E_{4,h} \; = \; \int_\Omega \Pi_h \psi_5 \nabla \cdot \lambda \nabla \Pi_h T \mathrm{d}\Omega.$$

We focus on the error term:

$$\delta E_{4,h} \;=\; \int_\Omega \Pi_h\psi_5\nabla\cdot\lambda\nabla T\mathrm{d}\Omega - \int_\Omega \Pi_h\psi_5\nabla\cdot\lambda\nabla\Pi_h T\mathrm{d}\Omega.$$

After a first integration by parts we get:

$$\begin{aligned}\delta E_{4,h} &= \int_\Omega \Pi_h\psi_5\lambda\nabla\cdot(\nabla(T-\Pi_h T))\,\mathrm{d}\Omega\\ &= \int_\Gamma \lambda\,\Pi_h\psi_5\nabla(T-\Pi_h T)\cdot\mathbf{n}\,\mathrm{d}\Gamma - \int_\Omega \lambda\nabla(\Pi_h\psi_5)\,(\nabla(T-\Pi_h T))\,\mathrm{d}\Omega.\end{aligned}$$

The boundary terms contribution is neglected as already mentioned. Then, the volume integral is equivalent to Estimation (9) for which Lemma 3.4 can be applied.
We obtain thus the final estimate:

$$\delta E_{4,h} \leq \frac{9}{2}\int_\Omega |\lambda|\rho(H(\Psi_5))\,|\Pi_h T - T|\mathrm{d}\Omega. \tag{21}$$

For the next three remaining terms, E_5, E_6 and E_7, because of their non-linearity, a slightly different algorithm of resolution is employed, by some mathematical artifice.

Study of E_5
. We start from the following developpement:

$$E_5 = \mu\sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega \psi_5\frac{\partial}{\partial x_j}\left(u_i\frac{\partial u_i}{\partial x_j}\right)d\Omega.$$

We resume next to the integral formulation and analyse the error term:

$$\delta e_5 = e_5 - e_5^h = \int_\Omega \Pi_h\psi_5\frac{\partial}{\partial x_j}\left(u_i\frac{\partial u_i}{\partial x_j}\right)d\Omega - \int_\Omega \Pi_h\psi_5\frac{\partial}{\partial x_j}\left(\Pi_h u_i\frac{\partial\Pi_h u_i}{\partial x_j}\right)d\Omega.$$

Because of the non-linearity of this term, we cannot directly add the two integrals and perform an integration by parts for this summation. In order to obtain a relation equivalent to (9) we perform firstly an integration by part for each one of the two integrals and discard the boundary terms.

$$\delta e_5 = -\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i\frac{\partial u_i}{\partial x_j}\right)d\Omega + \int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(\Pi_h u_i\frac{\partial\Pi_h u_i}{\partial x_j}\right)d\Omega =$$
$$\underbrace{-\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i\frac{\partial(u_i-\Pi_h u_i)}{\partial x_j}\right)d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\,(\Pi_h u_i - u_i)\,\frac{\partial\Pi_h u_i}{\partial x_j}d\Omega}_{I_2}.$$

Next, regarding integral I_1, after an integration by parts we have:

$$I_1 = -\int_\Omega \left(\frac{\partial^2}{\partial x_j^2}(\Pi_h\psi_5)\right)u_i\,(u_i-\Pi_h u_i)d\Omega +$$
$$\underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\frac{\partial u_i}{\partial x_j}\,(u_i-\Pi_h u_i)d\Omega}_{T_{11}} - \underbrace{\int_\Gamma u_i\left(\frac{\partial}{\partial x_j}(\Pi_h\psi_5)\right)\cdot n_j\,(u_i-\Pi_h u_i)d\Gamma}_{T_{12}}.$$

We recognize in T_{12} the hypothesis of Lemma 3.4. Thus:

$$T_{12} \leq \frac{9}{2}\int_\Omega |u_i|\rho(H(\psi_5))\,|u_i - \Pi_h u_i|\,d\Omega$$

Regarding T_{11} terms, we observe they are (closed to) identical to I_2, but opposite signs, thus we discard them.
Finally, the total contribution of E_5 term writes then:

$$\delta E_5 \leq \frac{27}{2}\mu\sum_{i=1}^{3}\int_\Omega |u_i|\rho(H(\psi_5))\,|u_i-\Pi_h u_i|\,d\Omega. \tag{22}$$

Study of E_6

. In contrast with the previous term, the gradient of the velocity vector is transposed, thus the velocity components will be crossed.

Let us write:

$$E_6 = \mu \sum_{i=1}^{3}\sum_{i=1}^{3} \int_\Omega \psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_j}{\partial x_i}\right) d\Omega.$$

We analyse the error term:

$$\delta e_6 = \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_j}\left(u_i \frac{\partial u_j}{\partial x_i}\right) d\Omega - \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_j}\left(\Pi_h u_i \frac{\partial \Pi_h u_j}{\partial x_i}\right) d\Omega$$

After a first integration by parts applied on both integrals we obtain:

$$\begin{aligned}\delta e_6 &= -\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i\frac{\partial u_j}{\partial x_i}\right) d\Omega + \int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(\Pi_h u_i \frac{\partial \Pi_h u_j}{\partial x_i}\right) d\Omega \\ &= \underbrace{-\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\left(u_i \frac{\partial (u_j - \Pi_h u_j)}{\partial x_i}\right) d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_j}(\Pi_h\psi_5)\ (\Pi_h u_i - u_i)\ \frac{\partial \Pi_h u_j}{\partial x_i} d\Omega}_{I_2}.\end{aligned}$$

As for the previous term, a second integration by parts is applied with the assumption that the boundary terms can be neglected:

$$I_1 = \underbrace{\int_\Omega \frac{\partial \Pi_h\psi_5}{\partial x_j}\ \frac{\partial u_i}{\partial x_i}\left(u_j - \Pi_h u_j\right) d\Omega}_{T_{11}} - \underbrace{\int_\Gamma u_i\left(\frac{\partial \Pi_h\psi_5}{\partial x_j}\right)\cdot n_j\ (u_j - \Pi_h u_j)\ d\Gamma.}_{T_{12}}$$

We recognize in T_{12} the estimation of Lemma 3.4, that is:

$$T_{12} \le \frac{9}{2}\mu \int_\Omega |u_i|\ \rho(H(\psi_5))\ |u_j - \Pi_h u_j|\ d\Omega.$$

For the T_{11} and I_2 after further calculation we obtain the interpolation error on velocity vector weighted by a vector with, as components, cross-products of gradient of the velocity with gradient of ψ_5 . Suppose we denote this weight Vec, we have:

$$\delta E_6 \le \frac{9}{2}\mu \sum_{i=1}^{3}\sum_{j=1}^{3}\int_\Omega |u_i|\ \rho(H(\psi_5))|u_j - \Pi_h u_j|\ d\Omega + \mu \int_\Omega \mathrm{Vec}\ |\mathbf{u} - \Pi_h \mathbf{u}|\ d\Omega.$$

with (for 3D problems):

$$\mathrm{Vec} = \begin{pmatrix} (\nabla w \times \nabla\psi_5)_y - (\nabla v \times \nabla\psi_5)_z \\ (\nabla u \times \nabla\psi_5)_z - (\nabla w \times \nabla\psi_5)_x \\ (\nabla v \times \nabla\psi_5)_x - (\nabla u \times \nabla\psi_5)_y \end{pmatrix}$$

and respectively $\mathrm{Vec} = (-\nabla v \times \nabla\psi_5, \nabla u \times \nabla\psi_5)^T$ for two dimensional case.

To synthetize, the following estimation holds:

$$\delta E_6 \le \mu \sum_{i=1}^{3} \int_\Omega \left(\sum_{j=1}^{3}\ (\frac{9}{2}|u_j|\ \rho(H(\psi_5)))\ +\ \mathrm{Vec}[i] \right) |u_i - \Pi_h u_i| \tag{23}$$

Study of E_7

. The previous remark regarding the crossing of terms holds for the 7th term too, because of multiplication with indentity matrix. This terms writes:

$$E_7\ =\ -\frac{2}{3}\mu \sum_{k=1}^{3}\sum_{i=1}^{3}\sum_{j=1}^{3} \int_\Omega \psi_5 \frac{\partial}{\partial x_k}\left(u_i\left(\frac{\partial u_j}{\partial x_j}\right)\right) d\Omega$$

And the error term to be analysed can be restricted to:

$$\delta e_7 = \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_k}\left(u_i\left(\frac{\partial u_j}{\partial x_j}\right)\right) d\Omega\ -\ \int_\Omega \Pi_h\psi_5 \frac{\partial}{\partial x_k}\left(\Pi_h u_i\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right)\right) d\Omega$$

After a first integration by parts the previous error term writes:

$$\begin{aligned}\delta e_7 \;=\; & -\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(u_i\frac{\partial u_j}{\partial x_j}\right)d\Omega + \int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(\Pi_h u_i\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right)\right)d\Omega = \\ & -\underbrace{\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(u_i\frac{\partial(u_j-\Pi_h u_j)}{\partial x_j}\right)d\Omega}_{I_1} + \underbrace{\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\,(\Pi_h u_i - u_i)\left(\frac{\partial \Pi_h u_j}{\partial x_j}\right)d\Omega}_{I_2}.\end{aligned}$$

Furthermore, we apply a second integration by parts:

$$\delta e_7 \;=\; \underbrace{\int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\frac{\partial u_i}{\partial x_j}\left(u_j-\Pi_h u_j\right)d\Omega}_{T_{11}} - \underbrace{\int_\Gamma \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\; u_i\cdot n_i\;(u_j\Pi_h u_j)d\Gamma}_{T_{12}} + I_2.$$

We recognize in T_{12} the estimation from Lemma 3.4, thus the following estimation holds:

$$T_{12} \le \frac{9}{2}\int_\Omega |u_i|\,\rho(H(\psi_5))\,|u_j-\Pi_h u_j| d\Omega.$$

Regarding the integral I_2 the following estimation holds:

$$I_2 \approx \int_\Omega \frac{\partial}{\partial x_k}(\Pi_h\psi_5)\left(\frac{\partial u_j}{\partial x_j}\right)\;(\Pi_h u_i - u_i)\,d\Omega$$

After summation we retrieve for this term the weight vector Vec from E_5, that is, for two dimensional case:

$$T_{11} + I_2 \;=\; -\,\mathrm{Vec}\;(\mathbf{u}-\Pi_h\mathbf{u}) \;+\; (v_x\psi_{5,x} - v_y\psi_{5,y}\;;\;u_y\psi_{5,y} - u_x\psi_{5,x})^T\;(\mathbf{u}-\Pi_h\mathbf{u})$$

Thus, the total contribution of E_7 term writes:

$$\delta E_7 \le -\frac{2}{3}\mu\;\sum_{k=1}^{3}\sum_{i=1}^{3}\sum_{j=1}^{3}\left[\frac{9}{2}\int_\Omega |u_i|\,\rho(H(\psi_5))\,|u_j-\Pi_h u_j| d\Omega + T_{11} + I_2\right]. \tag{24}$$

8. Continuous Adjoint system and discretization

Continuous adjoint system. The continuous adjoint system related to the objective functional writes:

$$W^* \in \mathcal{V}\,,\;\forall\psi\in\mathcal{V}\;:\;\left(\frac{\partial\Psi}{\partial W}(W)\psi, W^*\right) - (g,\psi) \;=\; 0. \tag{25}$$

Replacing $\Psi(W)$ by its Formulation (19) and integrating by parts, we get:

$$\begin{aligned}\left(\frac{\partial\Psi}{\partial W}(W)\psi, W^*\right) \;=\;& \int_\Omega (\psi(0)W^*(0) - \psi(T)W^*(T))\;\mathrm{d}\Omega \;+\; \int_0^T\int_\Omega \psi\left(-W^*_t - \left(\frac{\partial\mathcal{F}}{\partial W}\right)^*\nabla W^*\right)\mathrm{d}\Omega\,\mathrm{d}t \\ &+\; \int_0^T\int_\Gamma \psi\left[\left(\frac{\partial\mathcal{F}}{\partial W}\right)^* W^*.\mathbf{n} \;-\; \left(\frac{\partial\hat{\mathcal{F}}}{\partial W}\right)^* W^*.\mathbf{n}\right]\mathrm{d}\Gamma\,\mathrm{d}t\,.\end{aligned} \tag{26}$$

Consequently, the continuous adjoint state W^* must be such that:

$$-\,W^*_t - \left(\frac{\partial\mathcal{F}}{\partial W}\right)^*\nabla W^* \;=\; g_\Omega \;\text{ in }\; \Omega \tag{27}$$

with the associated adjoint boundary conditions:

$$\left(\frac{\partial\mathcal{F}}{\partial W}\right)^* W^*.\mathbf{n} \;-\; \left(\frac{\partial\hat{\mathcal{F}}}{\partial W}\right)^* W^*.\mathbf{n} \;=\; g_\Gamma \;\text{ on }\; \Gamma$$

and the final adjoint state condition:

$$W^*(T) \;=\; g_T.$$

The adjoint Euler equations is a system of advection equations, where the temporal integration goes backwards, *i.e.,* in the opposite direction of usual time. Thus, when solving the unsteady adjoint system, one starts at the end of the flow run and progresses back until reaching the start time.

Discrete adjoint system. Although any consistent approximation of the continuous adjoint system could be built by discretizing System (27), we choose the option to build the discrete adjoint system from the discrete state system defined by Relation (20) in order to be closer to the true error from which the continuous model is derived.

Consider the following semi-discrete unsteady compressible Euler model (explicit RK1 time integration):

$$\Psi_h^n(W_h^n, W_h^{n-1}) = \frac{W_h^n - W_h^{n-1}}{\delta t^n} + \Phi_h(W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N. \tag{28}$$

The time-dependent functional is discretized as follows:

$$j_h(W_h) = \sum_{n=1}^{N} \delta t^n \, j_h^{n-1}(W_h^{n-1}).$$

For the sake of simplicity, we restrict to the case $g_T = 0$ for the functional output defined by Relation (**??**). The problem of minimizing the error committed on the target functional $j(W_h) = (g, W_h)$, subject to the Euler system (28), can be transformed into an unconstrained problem for the following Lagrangian functional [20]:

$$\mathcal{L}(W_h, W_h^*) = \sum_{n=1}^{N} \delta t^n \, j_h^{n-1}(W_h^{n-1}) - \sum_{n=1}^{N} \delta t^n (W_h^{*,n})^T \Psi_h^n(W_h^n, W_h^{n-1}),$$

where $W_h^{*,n}$ are the N vectors of the Lagrange multipliers (which are the time-dependent adjoint states). The conditions for an extremum are:

$$\frac{\partial \mathcal{L}}{\partial W_h^{*,n}} = 0 \quad \text{and} \quad \frac{\partial \mathcal{L}}{\partial W_h^n} = 0, \quad \text{for } n = 1, ..., N.$$

The first condition is clearly verified from Relation (28). Thus the Lagrangian multipliers $W_h^{*,n}$ must be chosen such that the second condition of extrema is verified. This provides the unsteady discrete adjoint system:

$$\left\{ \begin{array}{lcl} W_h^{*,N} & = & 0 \\ W_h^{*,n-1} & = & W_h^{*,n} + \delta t^n \dfrac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - \delta t^n (W_h^{*,n})^T \dfrac{\partial \Phi_h}{\partial W_h^{n-1}}(W_h^{n-1}), \end{array} \right. \tag{29}$$

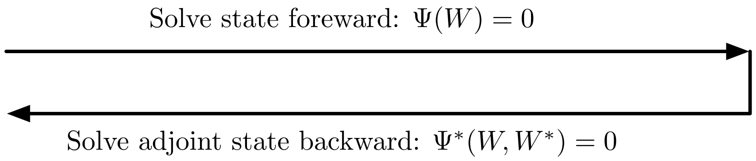
or equivalently, the semi-discrete unsteady adjoint model reads:

$$\Psi_h^{*,n}(W_h^{*,n}, W_h^{*,n-1}, W_h^{n-1}) = \frac{W_h^{*,n-1} - W_h^{*,n}}{-\delta t^n} + \Phi_h^*(W_h^{*,n}, W_h^{n-1}) = 0 \quad \text{for } n = 1, ..., N$$

with

$$\Phi_h^*(W_h^{*,n}, W_h^{n-1}) = \frac{\partial j_h^{n-1}}{\partial W_h^{n-1}}(W_h^{n-1}) - (W_h^{*,n})^T \frac{\partial \Phi_h}{\partial W_h^{n-1}}(W_h^{n-1}).$$

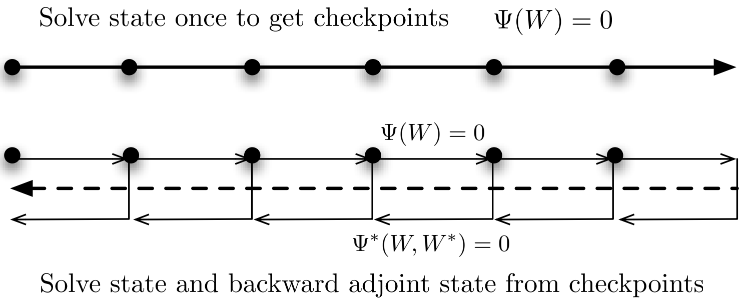
As the adjoint system runs in reverse time, the first expression in the adjoint System (29) is referred to as adjoint ”initialization”.


Solve state foreward: $\Psi(W) = 0$
Solve adjoint state backward: $\Psi^*(W, W^*) = 0$

Computing $W_h^{*,n-1}$ at time t^{n-1} requires the knowledge of state W_h^{n-1} and adjoint state $W_h^{*,n}$. Therefore, the knowledge of all states $\left\{W_h^{n-1}\right\}_{n=1,N}$ is needed to compute backward the adjoint state from time T to 0 which involves large memory storage effort. For instance, if we consider a 3D simulation with a mesh composed of one million vertices then we need to store at each iteration five millions solution data (we have 5 conservative variables). If we perform 1000 iterations, then the memory effort to store all states is 37.25 Gb for double-type data storage (or 18.62 for float-type data storage). Two strategies are employed to reduce importantly this drawback: checkpoints and interpolation.

The memory effort can be reduced by out-of-core storage of checkpoints as shown in the picture below. First the state-simulation is performed to store checkpoints. Second, when computing backward the adjoint, we first recompute all states from the checkpoint and store them in memory and then we compute the unsteady adjoint until the checkpoint physical time. This method implies a recomputing effort of the state W.

The other strategy consists in storing solution states in memory only each m solver iterations. When the unsteady adjoint is solved, solution states between two savings are linearly interpolated. This method leads to a loss of accuracy for the unsteady adjoint computation.


Solve state once to get checkpoints $\Psi(W) = 0$
$\Psi(W) = 0$
$\Psi^*(W, W^*) = 0$
Solve state and backward adjoint state from checkpoints

9. Optimal unsteady adjoint-based metric

9.1. *Error analysis (*N.B. applied to unsteady Euler model)

We replace in Estimation (**??**) operators Ψ and Ψ_h by their expressions given by Relations (19) and (20). In [32], it was observed that even for shocked flows, it is interesting to neglect the numerical viscosity term. We follow again this option. We also discard the error committed when imposing the initial condition. We finally get the following simplified error model:

$$\begin{aligned}\delta j \quad &\approx \quad \int_0^T \int_\Omega W^* \, (W - \Pi_h W)_t \, \mathrm{d}\Omega \, \mathrm{d}t \; + \; \int_0^T \int_\Omega W^* \, \nabla . (\mathcal{F}(W) - \Pi_h \mathcal{F}(W)) \, \mathrm{d}\Omega \, \mathrm{d}t \\ &- \quad \int_0^T \int_\Gamma W^* \, (\hat{\mathcal{F}}(W) - \Pi_h \hat{\mathcal{F}}(W))) . \mathbf{n} \, \mathrm{d}\Gamma \, \mathrm{d}t \, . \end{aligned} \tag{30}$$

Integrating by parts leads to:

$$\begin{aligned}\delta j \quad &\approx \quad \int_0^T \int_\Omega W^* \, (W - \Pi_h W)_t \, \mathrm{d}\Omega \, \mathrm{d}t \; - \; \int_0^T \int_\Omega \nabla W^* \, (\mathcal{F}(W) - \Pi_h \mathcal{F}(W)) \, \mathrm{d}\Omega \, \mathrm{d}t \\ &- \quad \int_0^T \int_\Gamma W^* \, (\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W))) . \mathbf{n} \, \mathrm{d}\Gamma \, \mathrm{d}t \, . \end{aligned} \tag{31}$$

with $\bar{\mathcal{F}} = \hat{\mathcal{F}} - \mathcal{F}$. We observe that this estimate of δj is expressed in terms of interpolation errors of the Euler fluxes and of the time derivative weighted by continuous functions W^* and ∇W^*.

Error bound with a safety principle. The integrands in Error Estimation (31) contain positive and negative parts which can compensate for some particular meshes. In our strategy, we prefer to not rely on these parasitic effects and to slightly overestimate the error. To this end, all integrands are bounded by their absolute values:

$$\begin{aligned}(g, W_h - W) \quad &\leq \quad \int_0^T \int_\Omega |W^*| \, |(W - \Pi_h W)_t| \, \mathrm{d}\Omega \, \mathrm{d}t + \int_0^T \int_\Omega |\nabla W^*| \, |\mathcal{F}(W) - \Pi_h \mathcal{F}(W)| \, \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \quad \int_0^T \int_\Gamma |W^*| \, |(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W)) . \mathbf{n}| \, \mathrm{d}\Gamma \, \mathrm{d}t \, . \end{aligned} \tag{32}$$

9.2. *Continuous error model*

Working in this framework enables us to write Estimate (32) in a continuous form:

$$\begin{aligned}|(g, W_h - W)| \approx \mathbf{E}(\mathbf{M}) \quad &= \quad \int_0^T \int_\Omega |W^*| \, |(W - \pi_{\mathcal{M}} W)_t| \, \mathrm{d}\Omega \, \mathrm{d}t \; + \; \int_0^T \int_\Omega |\nabla W^*| \, |\mathcal{F}(W) - \pi_{\mathcal{M}} \mathcal{F}(W)| \, \mathrm{d}\Omega \, \mathrm{d}t \\ &+ \quad \int_0^T \int_\Gamma |W^*| \, |(\bar{\mathcal{F}}(W) - \pi_{\mathcal{M}} \bar{\mathcal{F}}(W)) . \mathbf{n}| \, \mathrm{d}\Gamma \, \mathrm{d}t . \end{aligned} \tag{33}$$

We observe that the third term introduce a dependency of the error with respect to the boundary surface mesh. In the present paper, we discard this term and refer to [32] for a discussion of the influence of it. Then, introducing the continuous interpolation error, we can write the simplified error model as follows:

$$\mathbf{E}(\mathbf{M}) \quad = \quad \int_0^T \int_\Omega \operatorname{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t) \, \mathbf{H}(\mathbf{x}, t) \, \mathcal{M}^{-\frac{1}{2}}(\mathbf{x}, t) \right) \, \mathrm{d}\Omega \, \mathrm{d}t$$

$$\text{with}\quad \mathbf{H}(\mathbf{x},t) = \sum_{j=1}^{5} \Big([\Delta t]_j(\mathbf{x},t) + [\Delta x]_j(\mathbf{x},t) + [\Delta y]_j(\mathbf{x},t) + [\Delta z]_j(\mathbf{x},t) \Big)\,, \tag{34}$$

in which

$$[\Delta t]_j(\mathbf{x},t) = \left|W_j^*(\mathbf{x},t)\right| \cdot \left|H(W_{j,t})(\mathbf{x},t)\right|, \quad [\Delta x]_j(\mathbf{x},t) = \left|\frac{\partial W_j^*}{\partial x}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_1(W_j))(\mathbf{x},t)\right|,$$

$$[\Delta y]_j(\mathbf{x},t) = \left|\frac{\partial W_j^*}{\partial y}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_2(W_j))(\mathbf{x},t)\right|, \quad [\Delta z]_j(\mathbf{x},t) = \left|\frac{\partial W_j^*}{\partial z}(\mathbf{x},t)\right| \cdot \left|H(\mathcal{F}_3(W_j))(\mathbf{x},t)\right|.$$

Here, W_j^* denotes the j^{th} component of the adjoint vector W^*, $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$, and $H(W_{j,t})$ the Hessian of the j^{th} component of the time derivative of W. The mesh optimization problem writes:

$$\text{Find } \mathbf{M}_{opt} = \text{Argmin}_{\mathbf{M}}\, \mathbf{E}(\mathbf{M}), \tag{35}$$

under the constraint of bounded mesh fineness:

$$C_{st}(\mathbf{M}) = N_{st}, \tag{36}$$

where N_{st} is a specified total number of nodes. Since we consider an unsteady problem, the space-time (st) complexity used to compute the solution takes into account the time discretization. The above constraint then imposes the total number of nodes in the time integral, that is:

$$C_{st}(\mathbf{M}) = \int_0^T \tau(t)^{-1} \left(\int_\Omega d_{\mathcal{M}}(\mathbf{x},t)\mathrm{d}\mathbf{x} \right) \mathrm{d}t \tag{37}$$

where $\tau(t)$ is the time step used at time t of interval $[0,T]$.

9.3. Spatial minimization for a fixed t

Let us assume that at time t, we seek for the optimal continuous mesh $\mathbf{M}_{go}(t)$ which minimizes the instantaneous error, *i.e.*, the spatial error for a fixed time t:

$$\tilde{\mathbf{E}}(\mathbf{M}(t)) = \int_\Omega \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t)\,\mathbf{H}(\mathbf{x},t)\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x},t) \right) \mathrm{d}\mathbf{x}$$

under the constraint that the number of vertices is prescribed to

$$C(\mathbf{M}(t)) = \int_\Omega d_{\mathcal{M}(t)}(\mathbf{x},t)\,\mathrm{d}\mathbf{x} = N(t). \tag{38}$$

Similarly to [32], solving the optimality conditions provides the *optimal goal-oriented ("go") instantaneous continuous mesh* $\mathbf{M}_{go}(t) = (\mathcal{M}_{go}(\mathbf{x},t))_{\mathbf{x}\in\Omega}$ at time t defined by:

$$\mathcal{M}_{go}(\mathbf{x},t) = N(t)^{\frac{2}{3}}\, \mathcal{M}_{go,1}(\mathbf{x},t)\,, \tag{39}$$

where $\mathcal{M}_{go,1}$ is the optimum for $C(\mathbf{M}(t)) = 1$:

$$\mathcal{M}_{go,1}(\mathbf{x},t) = \left(\int_\Omega (\det \mathbf{H}(\bar{\mathbf{x}},t))^{\frac{1}{5}} \mathrm{d}\bar{\mathbf{x}} \right)^{-\frac{2}{3}} (\det \mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\ \mathbf{H}(\mathbf{x},t). \tag{40}$$

The corresponding optimal instantaneous error at time t writes:

$$\tilde{\mathbf{E}}(\mathbf{M}_{go}(t)) = 3\,N(t)^{-\frac{2}{3}} \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}} = 3\,N(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,. \tag{41}$$

For the sequel of this paper we denote: $\mathcal{K}(t) = \left(\int_\Omega (\det \mathbf{H}(\mathbf{x},t))^{\frac{1}{5}} \mathrm{d}\mathbf{x}\right)^{\frac{5}{3}}$.

9.4. Temporal minimization

To complete the resolution of optimization Problem (35-36), we perform a temporal minimization in order to get the optimal space-time continuous mesh. In other words, we need to find the optimal time law $t \rightarrow N(t)$ for the instantaneous mesh size. We consider the simpler case where the time step τ is specified by the user as a function of time $t \rightarrow \tau(t)$.

Temporal minimization for specified τ. Let us consider the case where the time step τ is specified by a function of time $t \rightarrow \tau(t)$. After the spatial optimization, the space-time error writes:

$$\mathbf{E}(\mathbf{M}_{go}) = \int_0^T \tilde{\mathbf{E}}(\mathbf{M}_{go}(t))\,\mathrm{d}t = 3\int_0^T N(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,\mathrm{d}t \tag{42}$$

and we aim at minimizing it under the following space-time complexity constraint:

$$\int_0^T N(t)\tau(t)^{-1}\,\mathrm{d}t = N_{st}. \tag{43}$$

In other words, we concentrate on seeking for *the optimal distribution of $N(t)$ when the space-time total number of nodes N_{st} is prescribed.* Let us apply the one-to-one change of variables:

$$\tilde{N}(t) = N(t)\tau(t)^{-1} \quad \text{and} \quad \tilde{\mathcal{K}}(t) = \tau(t)^{-\frac{2}{3}}\,\mathcal{K}(t)\,.$$

Then, our temporal optimization problem becomes:

$$\min_{\mathbf{M}}\ \mathbf{E}(\mathbf{M}) = \int_0^T \tilde{N}(t)^{-\frac{2}{3}}\,\tilde{\mathcal{K}}(t)\,\mathrm{d}t \quad \text{under constraint} \quad \int_0^T \tilde{N}(t)\,\mathrm{d}t = N_{st}\,.$$

The solution of this problem is given by:

$$\tilde{N}_{opt}(t)^{-\frac{5}{3}}\,\tilde{\mathcal{K}}(t) = const \quad \Rightarrow \quad N_{opt}(t) = C(N_{st})\,(\tau(t)\,\mathcal{K}(t))^{\frac{3}{5}}\,.$$

Here, constant $C(N_{st})$ can be obtained by introducing the above expression in space-time complexity Constraint (43), leading to:

$$C(N_{st}) = \left(\int_0^T \tau(t)^{-\frac{2}{5}}\,\mathcal{K}(t)^{\frac{3}{5}}\,\mathrm{d}t\right)^{-1} N_{st}\,,$$

which completes the description of the optimal space-time metric for a prescribed time step. Using Relation (39), the analytic expression of the optimal space-time goal-oriented metric $\mathbf{M}_{go}$ writes:

$$\mathcal{M}_{go}(\mathbf{x},t) = N_{st}^{\frac{2}{3}}\left(\int_0^T \tau(t)^{-\frac{2}{5}}\left(\int_\Omega (\det\mathbf{H}(\bar{\mathbf{x}},t))^{\frac{1}{5}}\,\mathrm{d}\bar{\mathbf{x}}\right)\mathrm{d}t\right)^{-\frac{2}{3}}\tau(t)^{\frac{2}{5}}\,(\det\mathbf{H}(\mathbf{x},t))^{-\frac{1}{5}}\,\mathbf{H}(\mathbf{x},t)\,. \tag{44}$$

We get the following optimal error:

$$\mathbf{E}(\mathbf{M}_{go}) = 3\,N_{st}^{-\frac{2}{3}}\left(\int_0^T \tau(t)^{-\frac{2}{5}}\left(\int_\Omega (\det\mathbf{H}(\mathbf{x},t))^{\frac{1}{5}}\,\mathrm{d}\mathbf{x}\right)\mathrm{d}t\right)^{\frac{5}{3}}\,. \tag{45}$$

9.5. *Temporal minimization for time sub-intervals*

The previous analysis provides the optimal size of the adapted meshes for each time level. Hence, this analysis requires the mesh to be adapted at each flow solver time step. But, in practice this approach involves a very large number of remeshing which is CPU consuming and spoils solution accuracy due to many solution transfers from one mesh to a new one. In consequence, a new adaptive strategy has been proposed in [2, 7] where the number of remeshing is controlled (thus drastically reduced) by generating adapted meshes for several solver time steps. The idea is to split the simulation time interval into n_{adap} *sub-intervals* $[t_{i-1}, t_i]$ for $i = 1, .., n_{adap}$. Each spatial mesh $\mathbf{M}^i$ is then kept constant during each sub-interval $[t_{i-1}, t_i]$. We could consider this partition as a *time discretization of the mesh adaptation problem*. In other words, the number of nodes N^i of the i^{th} adapted mesh $\mathbf{M}^i$ on sub-interval $[t_{i-1}, t_i]$ should for example be taken equal to:

$$N^i = \frac{\int_{t_{i-1}}^{t_i} N_{opt}(t)\tau(t)^{-1}\mathrm{d}t}{\int_{t_{i-1}}^{t_i} \tau(t)^{-1}\mathrm{d}t}.$$

Here, we propose a different option in which we get an optimal discrete answer.

Spatial minimization on a sub-interval. Given the continuous mesh complexity N^i for the single adapted mesh used during time sub-interval $[t_{i-1}, t_i]$, we seek for the optimal continuous mesh $\mathbf{M}^i_{go}$ solution of the following problem:

$$\min_{\mathbf{M}^i} \mathbf{E}^i(\mathbf{M}^i) = \int_\Omega \operatorname{trace}\left((\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x})\, \mathbf{H}^i(\mathbf{x})\, (\mathcal{M}^i)^{-\frac{1}{2}}(\mathbf{x}) \right) \mathrm{d}\mathbf{x} \quad \text{such that} \quad \mathcal{C}(\mathbf{M}^i) = N^i\,,$$

where matrix $\mathbf{H}^i$ on the sub-interval can be defined by either using an $\mathbf{L}^1$ or an $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^1}(\mathbf{x}) = \int_{t_{i-1}}^{t_i} \mathbf{H}(\mathbf{x}, t)\, \mathrm{d}t \quad \text{or} \quad \mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t \in [t_{i-1}, t_i]} \mathbf{H}(\mathbf{x}, t)\,,$$

with $\Delta t_i = t_i - t_{i-1}$. Processing as previously, we get the spatial optimality condition:

$$\mathcal{M}^i_{go}(\mathbf{x}) = (N^i)^{\frac{2}{3}}\ \mathcal{M}^i_{go,1}(\mathbf{x}) \quad \text{with } \mathcal{M}^i_{go,1}(\mathbf{x}) = \left(\int_\Omega (\det \mathbf{H}^i(\bar{\mathbf{x}}))^{\frac{1}{5}} \mathrm{d}\bar{\mathbf{x}} \right)^{-\frac{2}{3}} (\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\, \mathbf{H}^i(\mathbf{x}).$$

The corresponding optimal error $\mathbf{E}^i(\mathbf{M}^i_{go})$ writes:

$$\mathbf{E}^i(\mathbf{M}^i_{go}) = 3\, (N^i)^{-\frac{2}{3}} \left(\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}} \mathrm{d}\mathbf{x} \right)^{\frac{5}{3}} = 3\, (N^i)^{-\frac{2}{3}}\, \mathcal{K}^i\,.$$

To complete our analysis, we shall perform a temporal minimization. Again, we consider the case where the time step τ is specified by a function of time.

Temporal minimization for specified τ. After the spatial minimization, the temporal optimization problem reads:

$$\min_{\mathbf{M}} \mathbf{E}(\mathbf{M}) = \sum_{i=1}^{n_{adap}} \mathbf{E}^i(\mathbf{M}^i_{go}) = 3 \sum_{i=1}^{n_{adap}} (N^i)^{-\frac{2}{3}}\, \mathcal{K}^i \quad \text{such that} \quad \sum_{i=1}^{n_{adap}} N^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \right) = N_{st}\,.$$

We set the one-to-one mapping:

$$\tilde{N}^i = N^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \right) \quad \text{and} \quad \tilde{\mathcal{K}}^i = \mathcal{K}^i \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \right)^{\frac{2}{3}}\,,$$

then the optimization problem reduces to:

$$\min_{\mathbf{M}} \sum_{i=1}^{n_{adap}} (\tilde{N}^i)^{-\frac{2}{3}}\, \tilde{\mathcal{K}}^i \quad \text{such that} \quad \sum_{i=1}^{n_{adap}} \tilde{N}^i = N_{st}\,.$$

The solution is:

$$\tilde{N}^i_{opt} = C(N_{st})\, (\tilde{\mathcal{K}}^i)^{\frac{3}{5}} \quad \text{with} \quad C(N_{st}) = N_{st} \left(\sum_{i=1}^{n_{adap}} (\tilde{\mathcal{K}}^i)^{\frac{3}{5}} \right)^{-1}$$

$$\Rightarrow \quad N^i = N_{st} \left(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}} \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \right)^{\frac{2}{5}} \right)^{-1} (\mathcal{K}^i)^{\frac{3}{5}} \left(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \right)^{-\frac{3}{5}}\,.$$

and we deduce the following optimal continuous mesh $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ and error:

$$\mathcal{M}^i_{go}(\mathbf{x}) \;=\; N_{st}^{\frac{2}{3}} \Big(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}} \Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \Big)^{\frac{2}{5}} \Big)^{-\frac{2}{3}} \Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \Big)^{-\frac{2}{5}} (\det \mathbf{H}^i(\mathbf{x}))^{-\frac{1}{5}}\, \mathbf{H}^i(\mathbf{x}) \tag{46}$$

$$\mathbf{E}(\mathbf{M}_{go}) \;=\; 3\, N_{st}^{-\frac{2}{3}} \Big(\sum_{i=1}^{n_{adap}} (\mathcal{K}^i)^{\frac{3}{5}} \Big(\int_{t_{i-1}}^{t_i} \tau(t)^{-1} \mathrm{d}t \Big)^{\frac{2}{5}} \Big)^{\frac{5}{3}}\,, \tag{47}$$

with $(\mathcal{K}^i)^{\frac{3}{5}} = \displaystyle\int_\Omega (\det \mathbf{H}^i(\mathbf{x}))^{\frac{1}{5}} \mathrm{d}\mathbf{x}$.

Remark 1: The choice of the time sub-intervals $\{[t_{i-1}, t_i]\}_{i=1,n_{adap}}$ for a given n_{adap} is a mesh adaptation problem: what is the subdivision of interval $[0, T]$ in n_{adap} sub-intervals which will minimize the error on optimal mesh/metric $\mathcal{M}$? Since we take a constant metric in the sub-interval, the error main term in approximating $\mathcal{M}$ is the following integral of the absolute value of the time-derivative of $\mathcal{M}$:

$$\sum_{i=1}^{n_{adap}} \int_{t_{i-1}}^{t_i} \left| \frac{\partial \mathcal{M}(t)}{\partial t} \right| \mathrm{d}t.$$

which can be minimized for instance by isodistribution of the error by sub-interval.
Remark 2: The parameter n_{adap}, number of time sub-intervals is important for the efficiency of the adaptation algorithm. When a too large value is prescribed for n_{adap}, the error in mesh-to-mesh interpolation may increase. A compromise needs to be found by the user.

10. Theoretical Mesh Convergence Analysis

Our motivation in developing mesh adaptation algorithms is to get approximation algorithms with better convergence to continuous target data. By better, we mean that we want to reach the asymptotic high order convergence with a lower number of nodes and also for solutions involving discontinuities. Both improvements have been previously obtained in the context of steady inviscid flows [5]. The present paper focuses on mesh adaptation only controlling the spatial approximation error in the context of unsteady flows. From this standpoint, we just forget about time approximation error by specifying a time step or by considering an explicit time advancing context close to the one of references [2, 7, 23]. We assume that accuracy of the time advancing scheme is second-order at least. Then it can be derived from basic arguments that the time approximation error is essentially smaller than or equal to the spatial approximation error, controlled by the metric-based method, which justifies to adapt the mesh only to spatial error [2, 7, 23].

In order to measure the theoretical convergence order of the mesh adaptation algorithm, we need to compare the global mesh effort -the complexity N_{st}- with the corresponding error level. We recall that an adaptative or an uniform discretisation approach both using N degrees of freedom has a convergence of order α if the approximation error $|u - u_N|$ satisfies:

$$|u - u_N| \leq const.\ N^{-\frac{\alpha}{d}}$$

where d is the computational domain Euclidian dimension.

In the following, the convergence order of numerical solutions computed with the presented adaptive platform is addressed for the case of smooth flows.

10.1. Smooth flow fields

For some Hessian-based methods, *i.e.,* the $\mathbf{L}^\infty - \mathbf{L}^p$ approach of [23], an analysis is proposed for predicting the order of convergence to the continuous solution. In the goal-oriented method discussed in this paper, the adaptive state solution W_h generally does *not* converge to the continuous one W. However, in the case of regular solutions, the expression of the optimal error model indicates that the functional output indeed converges, and with a predictable order.

Smooth context with specified time step. Here, we are adapting the mesh with the purpose of reducing the spatial error. The space dimension is 3. Now, in this case, we have established the following estimate:

$$\mathbf{E}(\mathbf{M}_{go}) = O(N_{st}^{-\frac{2}{3}}) \quad \text{as} \quad N_{st} \to \infty.$$

for the case of an adaptation at each time step (45) and also for the case of an adaptation with a fixed mesh at each sub-interval (47), therefore:

Lemma 10.1. *The modeled spatial error on functional for a specified time-step converges at second-order rate.*

Smooth context with Courant-based time step. According to the argument recalled above, we are adapting the mesh $\mathbf{M}(t)$ in order to, by the magic of Courant condition, reduce both space and time error. The space-time dimension is 4. Now, in this case, we have established the following estimate:

$$\mathbf{E}(\mathbf{M}_{go}) = O(N_{st}^{-\frac{1}{2}}) \quad \text{as} \quad N_{st} \to \infty.$$

for the case of an adaptation at each time step, (**??**), and also for the case of an adaptation with a fixed mesh at each sub-interval, (**??**), therefore:.

Lemma 10.2. *The space-time modeled error on functional for Courant-based time step converges at second-order rate.*

Remark 4: Adapting for a time sub-interval instead of adapting at each time steps does not degrade the asymptotic convergence order which is a very good news. Nevertheless, such a series of adapted meshes is only space-time sub-optimal as the constant in the error term is larger than the adaptation at each time step.

Remark 5: Between sub-intervals, transfers of the solution fields from the previous mesh to the new one are necessary. The choice of the transfer operator has certainly some impact on the global accuracy (see for example [6]) together with how frequently it is applied. Reference [6] suggests to consider conservative interpolation for compressible flows.

11. From theory to practice

The continuous mesh adaptation problem takes the form of the following continuous optimality conditions:

$$\begin{aligned}
&W \in \mathcal{V}\,,\ \forall \varphi \in \mathcal{V}\,,\ \ (\Psi(\mathcal{M}, W), \varphi) = 0 && \text{“Navier-Stokes system”}\\
&W^* \in \mathcal{V}\,,\ \forall \psi \in \mathcal{V}\,,\ \left(\frac{\partial \Psi}{\partial W}(\mathcal{M}, W)\psi, W^*\right) = (g, \psi) && \text{“adjoint system”}\\
&\mathcal{M}(\mathbf{x}, t) = \mathcal{M}_{go}(\mathbf{x}, t) && \text{“adapted mesh”.}
\end{aligned} \tag{48}$$

In practice, it remains to approximate the above three-field coupled system by a discrete one and then solve it. For discretising the state and adjoint PDE's, we take the spatial schemes introduced previously which are explicit Runge-Kutta time advancing schemes. Such time discretization methods have non-linear stability properties like TVD which are particularly suitable for the integration of system of hyperbolic conservation laws where discontinuities appear. Discretising the last equation consists in specifying the mesh according to a discrete metric deduced from the discrete states.

In order to remedy all the problematics relative to mesh adaptation for time-dependent simulations stated in the introduction, an innovative strategy based on a fixed-point algorithm has been initiated in [3] and fully developed in [2]. The fixed-point algorithm aims at avoiding the generation of a new mesh at each solver iteration which would imply serious degradation of the CPU time and of the solution accuracy due to the large number of mesh modifications. It is also an answer to the lag problem occurring when, from the solution at t^n, a new adapted mesh is generated at level t^n to compute next solution at level t^{n+1}. In that latter case, since the mesh is not adapted to the solution evolution between time levels t^n and t^{n+1}, the mesh is always late with respect to the physics. The fixed point approach has been successfully applied to bi-fluids three-dimensional problems [23], to a blast simulation in a three-dimensional city [2] and to moving bodies simulations [7]. The basic idea consists in splitting the simulation time frame $[0, T]$ into n_{adap} adaptation sub-intervals:

$$[0,\ T]\ =\ [0 = t_0,\ t_1] \cup \ldots \cup [t_i,\ t_{i+1}] \cup \ldots \cup [t_{n_{adap}-1},\ t_{n_{adap}}]\,,$$

and to keep the same adapted mesh for each sub-interval. Consequently, the time-dependent simulation is performed with only n_{adap} different adapted meshes. The mesh used on each sub-interval is adapted to control the solution accuracy from t_{i-1} to t_i. We examine now how to apply this program.

11.1. Choice of the goal-oriented metric

The optimal adapted meshes for each sub-interval are generated according to analysis of Section 9.5. In this work, the following particular choice has been made:

- the Hessian metric for sub-interval i is based on a control of the temporal error in $\mathbf{L}^\infty$ norm:

$$\mathbf{H}^i_{\mathbf{L}^\infty}(\mathbf{x}) = \Delta t_i \max_{t \in [t_i, t_{i+1}]} \mathbf{H}(\mathbf{x}, t) = \Delta t_i\, \mathbf{H}^i_{\max}(\mathbf{x})\,,$$

- function $\tau : t \to \tau(t)$ is constant and equal to 1,
- all sub-intervals have the same time length Δt.

The optimal goal-oriented metric $\mathbf{M}_{go} = \{\mathbf{M}^i_{go}\}_{i=1,..,n_{adap}}$ then simplifies to:

$$\mathcal{M}^i_{go}(\mathbf{x}) = N_{st}^{\frac{2}{3}} \left(\sum_{i=1}^{n_{adap}} \Big(\int_\Omega (\det \mathbf{H}^i_{\max}(\mathbf{x}))^{\frac{1}{5}}\, \mathrm{d}\mathbf{x} \Big) \right)^{-\frac{2}{3}} (\Delta t)^{\frac{1}{3}} \left(\det \mathbf{H}^i_{\max}(\mathbf{x}) \right)^{-\frac{1}{5}}\ \mathbf{H}^i_{\max}(\mathbf{x})\,.$$

Remark 7: We notice that we obtain a similar expression of the optimal metric to the one proposed in [7], but it is presently obtained in a goal-oriented context and by means of a space-time error minimization.

11.2. Global fixed-point mesh adaptation algorithm

To converge the non-linear mesh adaptation problem, *i.e.,* converging the couple mesh-solution, we propose a fixed-point mesh adaptation algorithm. The parameter N_{st} representing the total computational effort is prescribed by the user and will influence the size of all the meshes defined during the time interval. That is, to compute any metric fields $\mathbf{M}^i_{go}$, we have to evaluate a global normalization factor which requires the knowledge of all $\mathbf{H}^i_{\max}$. Thus, the whole simulation from 0 to T must be performed to be able to evaluate all metrics $\mathbf{M}^i_{go}$. Similarly to [7], a global fixed point strategy covering the whole time-frame $[0, T]$, called *Global adjoint fixed-point mesh adaptation algorithm*, is considered:

```
//- Fixed-point loop to converge global space-time mesh adaptation
For j=1,$n_{nptfx}$
  //- Solve state once to get checkpoints
  For i=1,$n_{adap}$
    • $\mathcal{W}^j_{0,i}$ = ConservativeSolutionTransfer($\mathcal{H}^j_{i-1}, \mathcal{W}^j_{i-1}, \mathcal{H}^j_i$)
    • $\mathcal{W}^j_i$ = SolveStateForward($\mathcal{W}^j_{0,i}, \mathcal{H}^j_i$)
  End for
  //- Solve state and adjoint backward and store samples
  For i=$n_{adap}$, 1
    • $(\mathcal{W}^*)^j_i$ = AdjointStateTransfer($\mathcal{H}^j_{i+1}, (\mathcal{W}^*_0)^j_{i+1}, \mathcal{H}^j_i$)
    • $\{\mathcal{W}^j_i(k), (\mathcal{W}^*)^j_i(k)\}_{k=1,n_k}$ = SolveStateAndAdjointBackward($\mathcal{W}^j_{0,i}, (\mathcal{W}^*)^j_i, \mathcal{H}^j_i$)
    • $|\mathbf{H}_{\max}|^j_i$ = ComputeGoalOrientedHessianMetric($\mathcal{H}^j_i, \{\mathcal{W}^j_i(k), (\mathcal{W}^*)^j_i(k)\}_{k=1,n_k}$)
  End for
  • $C^j$ = ComputeSpaceTimeComplexity($\{|\mathbf{H}_{\max}|^j_i\}_{i=1,n_{adap}}$)
  • $\{\mathcal{M}^j_i\}_{i=1,n_{adap}}$ = ComputeUnsteadyGoalOrientedMetrics($C^j, \{|\mathbf{H}_{\max}|^j_i\}_{i=1,n_{adap}}$)
  • $\{\mathcal{H}^{j+1}_i\}_{i=1,n_{adap}}$ = GenerateAdaptedMeshes($\{\mathcal{H}^j_i\}_{i=1,n_{adap}}, \{\mathcal{M}^j_i\}_{i=1,n_{adap}}$)
End for
```

Let us describe this algorithm sketched in Figure 4. It consists in splitting the time interval $[0, T]$ into the n_{adap} mesh-adaptation time sub-intervals: $\{[t_{i-1}, t_i]\}_{i=1,..,n_{adap}}$ with $t_0 = 0$ and $t_{n_{adap}} = T$. On each sub-interval a different mesh is used. A time-forward computation of the state solution is first performed with a out-of-core storage of all checkpoints, which are taken to be $\{\mathcal{W}_h(t_i)\}_{i=1,..,n_{adap}}$. Between each sub-interval, the solution is interpolated on the new mesh using the conservative interpolation of [6]. Then, starting from the last sub-interval and proceeding until the first one, we recompute and store in memory all solution states of the sub-interval from the previously stored checkpoint in order to evaluate time-backward the adjoint state throughout the sub-interval. At the same time, we evaluate the Hessian metrics $\mathbf{H}^i_{\max}$ required to generate new adapted meshes for each sub-interval. To this end, n_k Hessian metric sample are computed on each sub-interval and intersected [2] to obtain $\mathbf{H}^i_{\max}$. At the end of the computation, the global space-time mesh complexity is evaluated, producing weights for the goal-oriented metric fields for each sub-interval. Finally, all new adapted meshes are generated according to the prescribed metrics. The time-forward, time-backward and mesh update steps are repeated into the $j = 1, .., n_{ptfx}$ global fixed-point loop. Convergence of the fixed-point is obtained in typically 5 global iterations.

This mesh adaptation loop has been fully parallelized. The solution transfer, the solver and the Hessians computation have been parallelized using a p-thread paradigm at the element loop level [4]. As regards the computation of the metrics and the generation of the adapted meshes, we observe that they can be achieved in a decoupled manner once the complete simulation has been performed, leading to an easy parallelization of these stages. Indeed, if n_{adap} processors are available, each metric and mesh can be done on one processor.

11.3. Computing the goal-oriented metric

The optimal goal-oriented metric is a function of the adjoint state, the adjoint state gradient, the state time derivative Hessian and the Euler fluxes Hessians. In practice, these continuous states are approximated by the discrete states and derivative recovery is applied to get gradients and Hessians. The discrete adjoint state W^*_h is taken to represent the adjoint state W^*. The gradient of the adjoint state ∇W^* is replaced by $\nabla_R W^*_h$ and the Hessian of each component of the flux vector $H(\mathcal{F}_i(W))$ is obtained from $H_R(\mathcal{F}_i(W_h))$. ∇_R (resp. H_R) stands for the operator that recovers numerically the first (resp. second) order derivatives of an initial piecewise linear solution field. In this paper, the recovery method is based on the $\mathbf{L}^2$-projection formula. Its formulation along with some comparisons to other methods is available in [5].

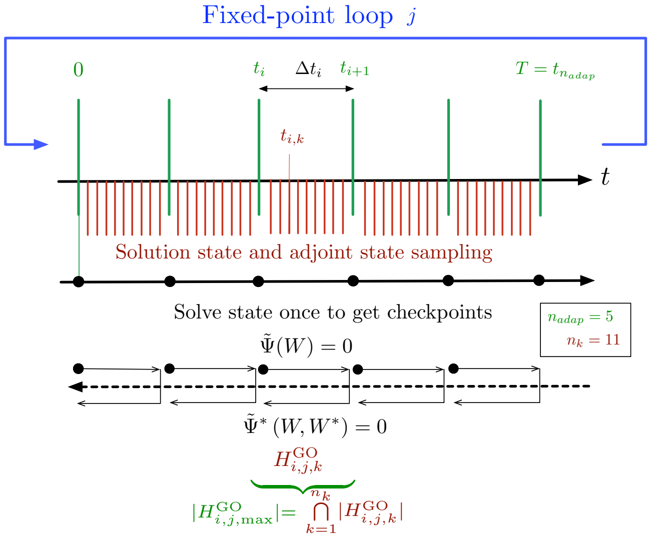

Fixed-point loop j
0
t_i
Δt_i
t_{i+1}
$T = t_{n_{adap}}$
$t_{i,k}$
t
Solution state and adjoint state sampling
Solve state once to get checkpoints
$n_{adap} = 5$
$n_k = 11$
$\tilde{\Psi}(W) = 0$
$\tilde{\Psi}^*(W, W^*) = 0$
$H^{\mathrm{GO}}_{i,j,k}$
$|H^{\mathrm{GO}}_{i,j,\max}| = \bigcap_{k=1}^{n_k} |H^{\mathrm{GO}}_{i,j,k}|$

Figure 4: Global Fixed-Point algorithm for unsteady goal-oriented anisotropic mesh adaptation

12. Numerical Experiments

The adaptation algorithms described in this paper have been implemented the CFD code `Wolf`. As regards meshing, goal-oriented mesh adaptation requires to update the surface mesh of Γ on which the functional is observed. This standpoint is needed in order to ensure a valid coupling between the volume mesh and the surface mesh. This implies to consider a more complex re-meshing phase. In our case, a local remeshing strategy has been considered. We use `Yams` [18] for the adaptation in 2D and `Feflo.a` [34] in 3D.

13. Conclusion

We have designed a new mesh adaptation algorithm which prescribes the spatial mesh of an unsteady simulation as the optimum of a goal-oriented error analysis. This method specifies both mesh density and mesh anisotropy by variational calculus. Accounting for unsteadiness is applied in a time-implicit mesh-solution coupling which needs a non-linear iteration, the fixed point. In contrast to the Hessian-based fixed-point of [2, 23] which iterates on each sub-interval, the new iteration covers the whole time interval, including forward steps for evaluating the state and backward ones for the adjoint. This algorithm has been successful applied to 2D and 3D blast wave Euler test cases and to the calculation of a 2D acoustic wave. Results demonstrate the favorable behavior expected from an adjoint-based method, which gives an automatic selection of the mesh regions necessary for the target output.

Several important issues for fully space-time computation have been addressed in this paper. Among them, the strategies for choosing the splitting in time sub-intervals and the accurate integration of time errors in the mesh adaptation process have been proposed, together with a more general formulation of the mesh optimization problem.

Time discretization error is not considered in this study. Solving this question is not so important for the type of calculation that are shown in this paper, but can be of paramount impact in many other cases, in particular when implicit time advancing is considered. In a future work, the authors plan to consider a space-time error analysis in the context of the proposed method.

14. Acknowledgements

This work has been supported by French National Research Agency (ANR) through COSINUS program (project ECINADS n^o ANR-09-COSI-003). HPC resources from GENCI-[CINES] (Grant 2010-x2010026386 and 2010-c2009025067) are also gratefully acknowledged.

References

[1] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008.

[2] F. Alauzet, P.J. Frey, P.L. George, and B. Mohammadi. 3D transient fixed point mesh adaptation for time-dependent problems: Application to CFD simulations. *J. Comp. Phys.*, 222:592–623, 2007.

[3] F. Alauzet, P.L. George, B. Mohammadi, P.J. Frey, and H. Borouchaki. Transient fixed point based unstructured mesh adaptation. *Int. J. Numer. Meth. Fluids*, 43(6-7):729–745, 2003.

[4] F. Alauzet and A. Loseille. On the use of space filling curves for parallel anisotropic mesh adaptation. In *Proceedings of the 18th International Meshing Roundtable*, pages 337–357. Springer, 2009.

[5] F. Alauzet and A. Loseille. High order sonic boom modeling by adaptive methods. *J. Comp. Phys.*, 229:561–593, 2010.

[6] F. Alauzet and M. Mehrenberger. P1-conservative solution interpolation on unstructured triangular meshes. *Int. J. Numer. Meth. Engng*, 84(13):1552–1588, 2010.

[7] F. Alauzet and G. Olivier. Extension of metric-based anisotropic mesh adaptation to time-dependent problems involving moving geometries. In *49th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2011-0896, Orlando, FL, USA, Jan 2011.

[8] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Log-Euclidean metrics for fast and simple calculus on diffusion tensors. *Magn. Reson. Med.*, 56(2):411–421, 2006.

[9] M.J. Baines. *Moving finite elements*. Oxford University Press, Inc., New York, NY, 1994.

[10] R. Becker and R. Rannacher. A feed-back approach to error control in finite element methods: basic analysis and examples. *East-West J. Numer. Math.*, 4:237–264, 1996.

[11] Y. Belhamadia, A. Fortin, and E. Chamberland. Three-dimensional anisotropic mesh adaptation for phase change problems. *J. Comp. Phys.*, 201:753–770, 2004.

[12] A. Belme, A. Dervieux, and F. Alauzet. Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows. *J. Comp. Phys., submitted*, 2012.

[13] M. Berger. *A panoramic view of Riemannian geometry*. Springer Verlag, Berlin, 2003.

[14] M. Berger and P. Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comp. Phys.*, 82(1):67–84, 1989.

[15] C.L. Bottasso. Anisotropic mesh adaption by metric-driven optimization. *Int. J. Numer. Meth. Engng*, 60:597–639, 2004.

[16] P.-H. Cournède, B. Koobus, and A. Dervieux. Positivity statements for a Mixed-Element-Volume scheme on fixed and moving grids. *European Journal of Computational Mechanics*, 15(7-8):767–798, 2006.

[17] J. Dompierre, M.G. Vallet, M. Fortin, Y. Bourgault, and W.G. Habashi. Anisotropic mesh adaptation: towards a solver and user independent CFD. In *AIAA 35th Aerospace Sciences Meeting and Exhibit*, AIAA-1997-0861, Reno, NV, USA, Jan 1997.

[18] P.J. Frey. Yams, a fully automatic adaptive isotropic surface remeshing procedure. RT-0252, INRIA, November 2001.

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

[20] M.B. Giles and N.A. Pierce. An introduction to the adjoint approach to design. *Flow, Turbulence and Combustion*, 65:393–415, 2000.

[21] M.B. Giles and E. Suli. *Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality*, pages 145–236. Cambridge University Press, 2002.

[22] C. Gruau and T. Coupez. 3D tetrahedral, unstructured and anisotropic mesh generation with adaptation to natural and multidomain metric. *Comput. Methods Appl. Mech. Engrg.*, 194(48-49):4951–4976, 2005.

[23] D. Guégan, O. Allain, A. Dervieux, and F. Alauzet. An l^∞-l^p mesh adaptive method for computing unsteady bi-fluid flows. *Int. J. Numer. Meth. Engng*, 84(11):1376–1406, 2010.

[24] F. Hecht and B. Mohammadi. Mesh adaptation by metric control for multi-scale phenomena and turbulence. In *35th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-1997-0859, Reno, NV, USA, Jan 1997.

[25] T. Leicht and R. Hartmann. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *J. Comp. Phys.*, 229(19):7344–7360, 2010.

[26] X. Li, M.S. Shephard, and M.W. Beal. 3D anisotropic mesh adaptation by mesh modification. *Comput. Methods Appl. Mech. Engrg.*, 194(48-49):4915–4950, 2005.

[27] R. Löhner. Adaptive remeshing for transient problems. *Comput. Methods Appl. Mech. Engrg.*, 75:195–214, 1989.

[28] A. Loseille and F. Alauzet. Continuous mesh model and well-posed continuous interpolation error estimation. RR-6846, INRIA, March 2009.

[29] A. Loseille and F. Alauzet. Optimal 3D highly anisotropic mesh adaptation based on the continuous mesh framework. In *Proceedings of the 18th International Meshing Roundtable*, pages 575–594. Springer, 2009.

[30] A. Loseille and F. Alauzet. Continuous mesh framework. Part I: well-posed continuous interpolation error. *SIAM J. Numer. Anal.*, 49(1):38–60, 2011.

[31] A. Loseille and F. Alauzet. Continuous mesh framework. Part II: validations and applications. *SIAM J. Numer. Anal.*, 49(1):61–86, 2011.

[32] A. Loseille, A. Dervieux, and F. Alauzet. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *J. Comp. Phys.*, 229:2866–2897, 2010.

[33] A. Loseille, A. Dervieux, P.J. Frey, and F. Alauzet. Achievement of global second-order mesh convergence for discontinuous flows with adapted unstructured meshes. In *37th AIAA Fluid Dynamics Conference and Exhibit*, AIAA-2007-4186, Miami, FL, USA, Jun 2007.

[34] A. Loseille and R. Löhner. Adaptive anisotropic simulations in aerodynamics. In *48th AIAA Aerospace Sciences Meeting and Exhibit*, AIAA-2010-169, Orlando, FL, USA, Jan 2010.

[35] C.C Pain, A.P. Humpleby, C.R.E. de Oliveira, and A.J.H. Goddard. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Comput. Methods Appl. Mech. Engrg.*, 190:3771–3796, 2001.

[36] P.W. Power, C.C. Pain, M.D. Piggott, F. Fang, G.J. Gorman, A.P. Umpleby, and A.J.H. Goddard. Adjoint a posteriori error measures for anisotropic mesh optimization. *Comput. Math. Appl.*, 52:1213–1242, 2006.

[37] A. Tam, D. Ait-Ali-Yahia, M.P. Robichaud, M. Moore, V. Kozel, and W.G. Habashi. Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. *Comput. Methods Appl. Mech. Engrg.*, 189:1205–1230, 2000.

[38] D.A. Venditti and D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *J. Comp. Phys.*, 187(1):22–46, 2003.

[39] R. Verfürth. *A review of A Posteriori Error Estimation and Adaptative Mesh-Refinement techniques*. Wiley Teubner Mathematics, New York, 1996.

[40] M. Wintzer, M. Nemec, and M.J. Aftosmis. Adjoint-based adaptive mesh refinement for sonic boom prediction. In *AIAA 26th Applied Aerodynamics Conference*, AIAA-2008-6593, Honolulu, HI, USA, Aug 2008.