CONTINUOUS MESH ADAPTATION MODELS FOR CFD

Youssef Mesri, [1] **Frédéric Alauzet,** [3] **Adrien Loseille,** [3]
Laurent Hascoët, [1] **Bruno Koobus,** [1,2] **Alain Dervieux** [1]

Abstract

Continuous mesh adaptation models can be derived from *a priori* error estimates. For a class of approximations, these estimates are expressed in terms of interpolation errors. We show it for P1-Galerkin-based schemes useful in Computational Fluid Dynamics (CFD). Applications with meshes adapted to steady Euler compressible supersonic flow are presented.

Key Words: mesh adaptation, *a priori* error, compressible flow

1 INTRODUCTION

With modern "controlled" mesh generators, it is possible to build automatically a mesh that conforms with a precise prescription of mesh density and stretching. The present paper deals with the choice of error-based criteria for specifying the adapted mesh. There are several ways in evaluating approximation error.

A first standpoint , see [1], gives *a posteriori* estimates of the error on a given mesh from the discrete solution. It detects, thanks to the introduction of an adjoint state, the regions of the computational domain where a local error is larger than a given prescribed tolerance. The region where the error is too large are refined. An extension of this principle to CFD can be found in [2].
A second standpoint consists in trying to find the best ideal mesh density minimizing a continuous model of the error, see [3] and [4]. In the case of approximations based on $P1$ interpolation, the use of Hessian-based metrics appeared to be an very fruitful approach, see for example [5],[6],[7]. In particular it opened widely the possibility to specify stretching in adapted meshes. In [8], a link between continuous error models and convergence for coarse meshes and non-smooth solutions is demonstrated, but even for smooth solutions, theoretical statements for non-elliptic models are yet very short.

1 INRIA Sophia-Antipolis, France
2 Université de Montpellier II, France
3 INRIA Rocquencourt, France

©Copyright: Computational Fluid Dynamics JOURNAL

In this paper we examine first the justification of a local interpolation error for the advection-diffusion model. Continuous piecewise linear finite-element discretisations are considered. A rigorous analysis is proposed. Then we analyse the extension to the Euler system for compressible gas. In both cases, a relation between approximation error and interpolation error is exhibited. Practical models for interpolation error reduction are then presented. Finally, examples of adapted meshes for supersonic steady flows involving complex shocks combinations are given.

2 MAIN NOTATIONS AND ASSUMPTIONS

2.1 Scalar representation of error

Let $\mathcal{M}$ be a parameterisation of the mesh of a computational domain Ω. To $\mathcal{M}$ corresponds a discretisation $\Psi(\mathcal{M},.)$ of a Partial Differential Equation (PDE). The discretised PDE is solved when a discrete function $W(\mathcal{M})$ satisfying:

$$\Psi(\mathcal{M}, W(\mathcal{M})) \;=\; 0.$$

is found. We are interested in reducing the error made in evaluating a discrete objective functional expressed as follows:

$$j(\mathcal{M}, W(\mathcal{M})) \;=\; (\psi, W(\mathcal{M}))$$

where ψ is a smooth given function. At first order, the approximation error on j is approximated by:

$$\overline{j}\,(\mathcal{M}, W(\mathcal{M})) \;=\; (\psi, W(\mathcal{M}) - W^{exact})$$

where W^{exact} is the non-discretised state solution. It is useful for the sequel to introduce the following **adjoint system**:

$$\text{find } p \text{ such that} \tag{1}$$

$$\left(\frac{\partial \Psi}{\partial U} v, p\right) = (\psi, v)\,, \text{for all } \; v \tag{2}$$

then the error on functional writes:

$$\overline{j}\,(\mathcal{M}, W) \;=\; \left(\frac{\partial \Psi}{\partial W}(W(\mathcal{M}) - W^{exact}), p\right)\;. \tag{3}$$

In order to study the functional error we shall analyse the right-hand side of (3). Before this, we need make more precise the mesh parametrisation.

2.2 Mesh parametrisation by a metric

An elegant way to parametrise the meshes consists of considering metric tensors. A **metric tensor** (or, more simply, a **metric**) $\mathcal{M}$ in $\mathbb{R}^n$ is a tensor field defined for any (x, y, z) of the computational domain. $\mathcal{M}(x, y, z)$ is a $n \times n$ symmetric definite (not degenerated) positive matrix; hence $\mathcal{M}$ can be split into the product of a rotation $\mathcal{S}_{\mathcal{M}}$ times a diagonal matrix times the inverse rotation:

$$\mathcal{M}(x, y) = \mathcal{S}_{\mathcal{M}} \begin{pmatrix} \frac{1}{m_{\mathcal{M},\theta}}^2 & 0 & 0 \\ 0 & \frac{1}{m_{\mathcal{M},\zeta}}^2 & 0 \\ 0 & 0 & \frac{1}{m_{\mathcal{M},\eta}}^2 \end{pmatrix} \mathcal{S}_{\mathcal{M}}^{-1} \tag{4}$$

where $\mathcal{S}_{\mathcal{M}}$, $m_{\mathcal{M},\theta}$, $m_{\mathcal{M},\zeta}$ and $m_{\mathcal{M},\eta}$ depend on x, y and z.
The **isotropic metric** subclass is defined by $m_{\mathcal{M},\theta} = m_{\mathcal{M},\zeta} = m_{\mathcal{M},\eta} = m$, i.e.:

$$\mathcal{M}_{iso} \;=\; m^{-2}\; Id_3\;,\; m > 0\;.$$

The **length** $L_{\mathcal{M}}(\overrightarrow{cd})$ of a vector $\overrightarrow{cd}$ in metric $\mathcal{M}$ is defined as follows:

$$L_{\mathcal{M}}(\overrightarrow{v}) = \int_0^1 \sqrt{\overrightarrow{v}\,.\mathcal{M}.\overrightarrow{v}}\,(x'\overrightarrow{c} + (1-x')\,\overrightarrow{d}\,)\;dx'.$$

We shall say that a mesh is defined by a metric $\mathcal{M}$, if it is a unit mesh for the metric, i.e. any edge $\overrightarrow{e}$ of the mesh is exactly of length $L_{\mathcal{M}}(\overrightarrow{e}\,)$ equal to 1. According to this definition, the coefficients $m_{\mathcal{M},\theta}$, $m_{\mathcal{M},\zeta}$, and $m_{\mathcal{M},\eta}$ are the **local mesh sizes** (or spatial discretisation steps) of $\mathcal{M}$ in each of the three directions $\theta_{\mathcal{M}}$, $\zeta_{\mathcal{M}}$ and $\eta_{\mathcal{M}}$ defined by the rotation $\mathcal{S}_{\mathcal{M}}$.
The rest of this paper will concentrate on a particular mode of mesh convergence. From a particular $\mathcal{M}$ we derive a set of isotropic metrics $\mathcal{M}_h$ that are homothetic to $\mathcal{M}$:

$$\mathcal{M}_h \;=\; h^{-2}\mathcal{M}\;,\; 0 < h < h_{max}. \tag{5}$$

In contrast to usual notations, making h tend to zero will refer to a sequence of meshes that are homothetic to $\mathcal{M}$, according to (5), but with finer and finer mesh size. It is assumed that the meshes are of good quality, which implies that angle conditions satisfied by meshes described by the model metric $\mathcal{M}$ are also satisfied by meshes described by the homothetic metric $\mathcal{M}_h$. We examine how in these conditions the numerical error behaves.

3 ADVECTION-DIFFUSION MODEL

We propose first a rigourous analysis of the error behaviour for a simplified model problem, the advection-diffusion. Then, in Section 4, we shall follow the same lines for addressing the case of the Euler system.

3.1 Governing equations

We consider the following advection-diffusion equation in $\Omega \subset \mathbb{R}^3$:

$$\nabla.(\overrightarrow{V}\,u) - \nabla.(\nu \nabla u) \;=\; f \;in\; \Omega \tag{6}$$

Here $\overrightarrow{V} : \Omega \to \mathbb{R}^3$ denotes a given velocity smooth enough, $\overrightarrow{V} \in (C^1)^d$. For well-posedness it is useful to assume that $\nabla.\overrightarrow{V} \geqslant \alpha_0$ a.e. in Ω , for a positive α_0. The diffusion coefficient ν is supposed to be positive. Right-hand side f is a smooth enough function on Ω.

Dirichlet conditions: The problem is examined first for Dirichlet boundary condition:

$$u = 0 \text{ on } \partial\Omega\;. \tag{7}$$

We write the weak formulation to the problem (6):

$$\text{find } u \in H_0^1(\Omega) \text{ such as} \tag{8}$$

$$a(u, v) = f, \text{for all } \; v \in H_0^1(\Omega) \tag{9}$$

where the bilinear form a is defined by:

$$a(u, v) = (\nabla.(\overrightarrow{V}\,u), v) + \nu(\nabla u, \nabla v) \tag{10}$$

With the above assumptions, $a(u, v)$ is a coercive continuous bilinear form in H_0^1, then existence and uniqueness of the solution of (9) hold.

3.2 Approximation of Convection-Diffusion equation by FEM-discretization

3.2.1 Global error estimate in $H^1(\Omega)$

Let $\mathcal{T}_h$ be a mesh described by $\mathcal{M}_h$. We assume for simplicity that $\Omega = \cup_{K \in \mathcal{T}_h} K$. Let $V_h = \{\varphi \in H^1(\Omega) : \varphi | K \in P_1(K), \forall K \in \mathcal{T}_h\}$ the usual

continuous piecewise linear finite element space. Let Π_h the usual interpolation operator in V_h, applying on continuous functions ($\Pi_h\phi(x) = \phi(x)\ \forall\ x\ vertex$) We look for $u_h \in V_h$ such that

$$a(u_h, \varphi_h) \ = \ b(\varphi_h) \quad \forall \varphi_h \in V_h. \tag{11}$$

Second-order convergence holds in a weak sense. Let us recall this in short. We assume that the solution u is smooth enough:

$$u \ \in \ H^2(\Omega). \tag{12}$$

The classical interpolation error analysis applies to our family of meshes:

$$\begin{aligned} |u - \Pi_h u|_{H^1} \ &\leq \ K_1 h |u|_{H^2} \\ |u - \Pi_h u|_{L^2} \ &\leq \ K_2 h^2 |u|_{H^2} \end{aligned} \tag{13}$$

Substracting (11) from the variational formulation of (6) results in the following relation for the error $e = u_h - u$:

$$a(e, \varphi_h) \ = \ 0 \quad \forall \varphi_h \in V_h. \tag{14}$$

We denote by e_h the implicit error, that is the deviation $e_h = u_h - \Pi_h u$ and we set $\varphi_h = e_h$ in (14):

$$a(e, e_h) \quad = \quad 0 \tag{15}$$

then we write:

$$a(e_h, e_h) \quad = \quad a(\Pi_h u - u, e_h) \tag{16}$$

and using a Poincaré inequality, we get:

$$\|u_h - \Pi_h u\|_{H^1} \leqslant C(\overrightarrow{V})h\|u\|_{H^2} \tag{17}$$

$$\|u_h - u\|_{H^1} \leqslant C(\overrightarrow{V})h\|u\|_{H^2} \tag{18}$$

3.2.2 Global error estimate in $L^2(\Omega)$

For this, we adapt the Aubin-Nitsche method. We introduce an adjoint function $\overline{\phi} \in H_0^1 \cap H^2$ which satisfies for any $\psi \in H_0^1 \cap H^2$

$$a(\psi, \overline{\phi}) = (\psi, u_h - \Pi_h u) \tag{19}$$

then

$$\begin{aligned} |u_h - \Pi_h u|_{L^2} &= a(u_h - \Pi_h u, \overline{\phi}) \\ &= a(u_h - \Pi_h u, \overline{\phi} - \Pi_h\overline{\phi}) \\ &\quad + a(u_h - \Pi_h u, \Pi_h\overline{\phi}) \\ &= a(u_h - \Pi_h u, \overline{\phi} - \Pi_h\overline{\phi}) \\ &\quad + a(u - \Pi_h u, \Pi_h\overline{\phi}) \\ &= A(h) + B(h) \end{aligned} \tag{20}$$

with

$$\begin{aligned} A(h) &= a(u_h - \Pi_h u, \overline{\phi} - \Pi_h\overline{\phi}) \\ B(h) &= a(u - \Pi_h u, \Pi_h\overline{\phi}) \ . \end{aligned} \tag{21}$$

By Cauchy-Schwarz and (18) we get

$$|A(h)| \leq K_1^2 h^2 \tag{22}$$

Besides,

$$\begin{aligned} |B(h)| &\leq \big|\big(\nabla(\Pi_h\overline{\phi} - \overline{\phi}), \nabla(u - \Pi_h u)\big)\big| \\ &\quad + \big|\big(\nabla\overline{\phi}, \nabla(u - \Pi_h u)\big)\big| \\ &\leq K_1^2 h^2 |u|_{H^2} + \big|\big(-\Delta\overline{\phi}, u - \Pi_h u\big)\big| \\ &\leq K_1^2 h^2 |u|_{H^2} + |\Delta\overline{\phi}|_2 |u - \Pi_h u|_2 \\ &\leq K_1^2 h^2 |u|_{H^2} + |\Delta\overline{\phi}|_2 K_2 h^2 |u|_{H^2} \ . \end{aligned} \tag{23}$$

Taking $K = 2K_1^2 + |\Delta\overline{\phi}|_2 K_2$ we get:

$$|u - u_h|_2 \leq K h^2 |u|_{H^2} \ . \tag{24}$$

This states the second-order convergence of the error e, and this and (13) imply the same convergence for e_h. Returning to the error expression, we have:

$$\begin{aligned} \nu(\nabla e_h, \nabla\Pi_h\varphi) + (\overrightarrow{V}\ \nabla e_h, \Pi_h\varphi) = \\ \nu(\nabla(u - \Pi_h u), \nabla\Pi_h\varphi) \\ +(\overrightarrow{V}\ \nabla(u - \Pi_h u), \Pi_h\varphi) \ . \end{aligned} \tag{25}$$

This equality shows the determinant **impact of the interpolation error** on the global one. It also writes:

$$\begin{aligned} \nu(\nabla e_h, \nabla\Pi_h\varphi) + (\overrightarrow{V}\ \nabla e_h, \Pi_h\varphi) = \\ -\nu(u - \Pi_h u, \Delta\Pi_h\varphi) \\ +(u - \Pi_h u, \nabla.(\overrightarrow{V}\ \Pi_h\varphi)) \ . \end{aligned} \tag{26}$$

Then, in order to reduce the error on the functional, and for a smooth ϕ, it is sufficient to make the interpolation error $u - \Pi_h u$ tend to zero in L^2.
Remark: it is possible to continue the analysis as in [9]) and to get:

$$\begin{aligned} \nu(\nabla(u - \Pi_h u), \nabla\Pi_h\varphi) + (\overrightarrow{V}\ \nabla(u - \Pi_h u), \Pi_h\varphi) \\ = \nu h^2 (g(u, d), \varphi) + o(\nu h^2). \end{aligned}$$

where the parentheses in RHS have to be understood as a distribution product.□

3.3 Inflow/outflow condition

Another important type of boundary condition is the following one: let $\overrightarrow{n}$ be the normal to $\partial\Omega$, outward pointing, we assume that the boundary $\partial\Omega$ is split in three smooth enough sub-components:

$$\begin{aligned}\Sigma_0 &= \{x \in \partial\Omega,\ \overrightarrow{V}.\overrightarrow{n} = 0\},\\ \Sigma_+ &= \{x \in \partial\Omega,\ \overrightarrow{V}.\overrightarrow{n} > 0\},\\ \Sigma_- &= \{x \in \partial\Omega,\ \overrightarrow{V}.\overrightarrow{n} < 0\}. \end{aligned} \quad (27)$$

Let u_- a constant function given on Σ_-, an inflow/outflow version of the advection-diffusion model writes:

$$a(u,v) = b(v) \quad (28)$$

with

$$a(u,v) = -\int_\Omega \nabla v(\overrightarrow{V}\,u)dx + \int_\Omega \nu\nabla v.\nabla u dx + \int_{\Sigma_+} vu\overrightarrow{V}.\overrightarrow{n}\,d\sigma \quad (29)$$

$$b(v) = \int_\Omega fv dx - \int_{\Sigma_-} vu_-\overrightarrow{V}.\overrightarrow{n}\,d\sigma \quad (30)$$

This interpretes as follows:

$$\begin{aligned}&\nabla.\overrightarrow{V}\,u - \nabla.(\nu\nabla u) = f \ in\ \Omega\\ &\frac{\partial u}{\partial n} = 0 \ \text{ on } \Sigma_0\ .\\ &\frac{\partial u}{\partial n} = 0 \ \text{ on } \Sigma_+\ .\\ &\nu\frac{\partial u}{\partial n} - u\,\overrightarrow{V}.\overrightarrow{n} = -u_-\,\overrightarrow{V}.\overrightarrow{n} \ \text{ on } \Sigma_-\ .\end{aligned}$$

For ν tending to zero, the last condition is an inflow condition for the advection problem.
Similarly to the Dirichlet case, we get:

$$\begin{aligned}(\overrightarrow{V}\,e_h,\nabla\varphi_h) - \nu(\nabla e_h,\nabla\varphi_h) - \int_{\Sigma_+} e_h\varphi_h\overrightarrow{V}.\overrightarrow{n}\,d\sigma =\\ ((u-\Pi_h u),\nabla.(\overrightarrow{V}\,\varphi_h)) - \nu(\nabla(u-\Pi_h u),\nabla\varphi_h)\\ -\int_{\Sigma_+}(u-\Pi_h u)\varphi_h\overrightarrow{V}.\overrightarrow{n}\,d\sigma \end{aligned} \quad (31)$$

where $e_h = u_h - \Pi_h u$ the **implicit error**. This will be small if the **interpolation error** norm $|u-\Pi_h u|_{L^2}$ is small.

4 NUMERICAL CFD MODEL

In the case of the Euler equations, we are not able to produce a convergence result, but we shall apply the above strategy for estimating the right-hand side of the error equation.

4.1 Governing equations

The Euler equations, which express the conservation of mass, momentum, and energy for the flow of inviscid, compressible fluids, may be written in the following integral conservation law form

$$\frac{\partial}{\partial t}\int_V W dV + \oint_S \mathcal{F}\cdot\mathbf{ndS} = 0 \quad (32)$$

where W is the vector of conserved variables and $\mathcal{F}$ the flux of W across the bounding surface S with outward unit normal $\mathbf{n}$ of a any control volume V. The column vector W and flux vector $\mathcal{F}$ are given by

$$W = \begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ \rho E\end{pmatrix}; \qquad \mathcal{F}(W) = \begin{pmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}u + p\mathbf{i}_x\\ \rho\mathbf{u}v + p\mathbf{i}_y\\ \rho\mathbf{u}w + p\mathbf{i}_z\\ \rho\mathbf{u}H\end{pmatrix} \quad (33)$$

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 $H = E + \frac{p}{\rho}$. If the fluid is assumed to be a thermally perfect ideal gas, then the closure equation linking the pressure p and the conserved quantities ρ and E is provided by the equation of state

$$p = \rho(\gamma-1)[E - \frac{1}{2}(u^2+v^2+w^2)] \quad (34)$$

in which γ stands for the ratio of specific heats at constant pressure and volume.

4.2 Spatial discretization

The Euler system is solved by means of a vertex-centered Mixed-Element-Volume approximation on unstructured meshes, as in [13]. The consistent part is a Galerkin formulation. The stabilising part relies on a Roe Riemann solver combined with a MUSCL reconstruction with Van Albada type limiters. This produces a space accuracy of order two. Let us mention that for solution of the steady system, an explicit multi-stage pseudo-time integration which does not influence the spatial accuracy is applied. The proposed analysis is performed in this paper for pure Galerkin formulation, which means that we do not consider the effect of Godunov-type stabilization. W is a non-scalar field on the computational domain Ω and satisfy, for any non-scalar test function ϕ in the functional space $(H^1(\Omega))^5$.

$$\int_\Omega \phi\nabla.\mathcal{F}(W)d\Omega - \int_{\partial\Omega}\phi\overline{\mathcal{F}}(W).nd\partial\Omega = 0.$$

In the corresponding discretization, the test function is taken into a V_h included in V:

$$\int_\Omega \phi_h \nabla.\mathcal{F}_h(W_h)d\Omega - \int_{\partial\Omega} \phi_h \overline{\mathcal{F}_h}\,(W_h).nd\partial\Omega \;=\; 0$$

where $\mathcal{F}_h(W)$ is the interpolate of $\mathcal{F}$, i.e. $\mathcal{F}_h(W_h) = \Pi_h\mathcal{F}(W_h)$, and same for $\overline{\mathcal{F}_h}\,(W_h)$.
Replacing ϕ by ϕ_h in the continuous system and choosing then $\phi_h = \Pi_h\phi$ in both systems we get:

$$\int_\Omega \phi_h \nabla.\mathcal{F}_h(W_h)d\Omega - \int_{\partial\Omega} \phi_h \overline{\mathcal{F}_h}\,(W_h).nd\partial\Omega \;= \int_\Omega \phi_h \nabla.\mathcal{F}(W)d\Omega - \int_{\partial\Omega} \phi_h \overline{\mathcal{F}}\,(W).nd\partial\Omega$$

In the above systems, the boundary data are involved inside the $\overline{\mathcal{F}}\,(W)$ and $\overline{\mathcal{F}_h}\,(W)$ boundary terms. For the sake of simplicity, we assume that these terms can be split in W-dependant terms, denoted respectively by $\overline{\mathcal{F}}^{\,out}(W)$ and $\overline{\mathcal{F}_h}^{\,out}(W_h)$, and constant terms, denoted $\overline{\mathcal{F}}^{\,in}$ and $\overline{\mathcal{F}_h}^{\,in}$. Therefore:

$$\int_\Omega \phi_h \nabla.\mathcal{F}_h(W_h)d\Omega - \int_{\partial\Omega} \phi_h \overline{\mathcal{F}_h}^{\,out}(W_h).nd\partial\Omega \;= \int_\Omega \phi_h \nabla.\mathcal{F}(W)d\Omega - \int_{\partial\Omega} \phi_h \overline{\mathcal{F}}^{\,out}(W).nd\partial\Omega$$

4.3 Error estimate

We assume now that both W and ϕ are several times continuously differentiable. In order to estimate the error, we introduce on both sides the same expression with interpolations:

$$\int_\Omega \phi_h \nabla.(\mathcal{F}_h(W_h) - \Pi_h\mathcal{F}(W))d\Omega - \int_{\partial\Omega} \phi_h (\overline{\mathcal{F}_h}^{\,out}(W_h) - \Pi_h\overline{\mathcal{F}}^{\,out}(W)).nd\partial\Omega \;= \int_\Omega \phi_h \nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))d\Omega - \int_{\partial\Omega} \phi_h (\overline{\mathcal{F}}^{\,out}(W) - \Pi_h\overline{\mathcal{F}}^{\,out}(W)).nd\partial\Omega.$$

The left-hand side will be inverted and the right-hand side will be expanded to get the error estimate.

4.4 Interpolation errors

We recall that $\Pi_h\mathcal{F}(W) = \mathcal{F}_h(W)$. The left-hand side writes:

$$LHS \;=\; \int_\Omega \phi_h \nabla.(\mathcal{F}_h(W_h) - \mathcal{F}_h(W))d\Omega - \int_{\partial\Omega} \phi_h (\overline{\mathcal{F}_h}^{\,out}(W_h) - \overline{\mathcal{F}}_h^{\,out}(W)).nd\partial\Omega.$$

We linearize it as follows:

$$LHS \;=\; \int_\Omega \phi_h \nabla.(\Pi_h \frac{\partial\mathcal{F}}{\partial W}(W_h - W))d\Omega - \int_{\partial\Omega} \phi_h (\Pi_h \frac{\partial\overline{\mathcal{F}_h}^{\,out}}{\partial W}(W_h - W)).nd\partial\Omega.$$

Where the derivatives $\frac{\partial\mathcal{F}}{\partial W}$ and $\frac{\partial\overline{\mathcal{F}_h}^{\,out}}{\partial W}$ are evaluated from vertex values of W. We denote this in short:

$$LHS \;=\; \mathcal{A}_h(W)(W_h - \Pi_h W)$$

We assume that the corresponding linearized operator, which is the Jacobian $\mathcal{A}_h(W)$ of the discretized Euler system is invertible. This means that the **implicit error** $W_h - \Pi_h W$ is obtained as the unique solution of:

$$W_h - \Pi_h W \;=\; (\mathcal{A}_h(W))^{-1} \;\; RHS\;.$$

In the right-hand side:

$$RHS \;=\; \int_\Omega \phi_h \nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))d\Omega - \int_{\partial\Omega} \phi_h (\overline{\mathcal{F}}^{\,out}(W) - \Pi_h\overline{\mathcal{F}}^{\,out}(W)).nd\partial\Omega$$

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

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

with:

$$RHS_1 \;=\; \int_\Omega (\Pi_h\phi - \phi)\nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))d\Omega - \int_{\partial\Omega} (\Pi_h\phi - \phi)(\overline{\mathcal{F}}^{\,out}(W) - \Pi_h\overline{\mathcal{F}}^{\,out}(W)).nd\partial\Omega\;.$$

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:

$$RHS_2 \;=\; \int_\Omega \phi \nabla.(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))d\Omega - \int_{\partial\Omega} \phi (\overline{\mathcal{F}}^{\,out}(W) - \Pi_h\overline{\mathcal{F}}^{\,out}(W)).nd\partial\Omega$$

and we transform it as follows:

$$RHS_2 \;=\; -\int_\Omega (\nabla\phi).(\mathcal{F}(W) - \Pi_h\mathcal{F}(W))d\Omega$$

$$+\int_{\partial\Omega}\phi(\mathcal{F}(W)-\Pi_h\mathcal{F}(W)).nd\partial\Omega-$$
$$\int_{\partial\Omega}\phi(\overline{\mathcal{F}}^{out}(W)-\Pi_h\overline{\mathcal{F}}^{out}(W)).nd\partial\Omega$$

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 [9]. 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 [9], this terms 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:

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

4.5 Provisional conclusion

The above study shows that the implicit error $W_h - \Pi_h W$ is a function of the interpolation error $W-\Pi_h W$. An interesting option consists then in reducing the interpolation error. This option is studied in Sections 5 and 6.

5 INTERPOLATION ERROR REDUCTION

The above analysis motivates the application of strategies relying on the minimization of L^2 and L^∞ models for interpolation error. This is an attracting option since anisotropic meshes are easily specified via the so-called anisotropic metric parametrization. This kind of strategy is already rather popular, we recall its main features. See [6, 11] for more details.

5.1 Mesh adaptation iteration

For stationary problems, the mesh adaptation scheme aims at finding a fixed point for the mesh-solution couple. In other words, the goal is to converge towards the stationary solution of the problem and similarly towards the corresponding invariant adapted mesh.

At each stage, a numerical solution is computed on the current mesh and has to be analyzed by means of an error estimate. The mesh adaptation is based on the edge length computation with respect to a discrete anisotropic metric specified at the mesh vertices. This metric is defined *via* a geometric error estimate that translates the solution variations into elements sizes and directions. Next, an adapted mesh is generated with respect to this metric. Finally, the solution is interpolated linearly on the new mesh. This procedure is repeated until the convergence of the solution and of the mesh is achieved.

5.2 Metric computation

We will focus here on the construction of the metric tensor based on the interpolation error in L^2 or L^∞ norm. On a particular element of the mesh, both norms are evaluated in a similar way using a geometric error estimate. They are then defined at every mesh vertex. For CFD simulations, we propose a specific error estimate normalization.

A geometric error estimate. As for the elliptic problems, we shall assume here that controlling the interpolation error allows us to control the approximation error. Hence, we deliberately based our anisotropic geometric error estimate on the interpolation error. The error estimate aims at defining a discrete metric field that prescribes size and stretching requirements for the mesh adaptation procedure. Consequently, in an adapted mesh the interpolation error is equidistributed in all directions. More precisely, for each mesh element K, the anisotropic error interpolation bound involves the second derivatives of the variable u:

$$\| u-\Pi_h u \|_{\infty,K} \leq c_d \max_{x\in K}\max_{e\in E_K}\langle \overrightarrow{e}\;,\;|H_u(x)|\;\overrightarrow{e}\;\rangle = \varepsilon_K\,, \tag{35}$$

where c_d is a constant related to the dimension, E_K is the set of edges of K and $|H_u| = \mathcal{R}|\Lambda|\mathcal{R}^{-1}$ is the absolute value of the Hessian of the variable u ($\mathcal{R}$ being the matrix of eigenvectors and $|\Lambda| = diag(|\lambda_i|)$ being the absolute value of the matrix of eigenvalues).

Metric construction. A discrete metric approximation which uses the mesh vertices as support is considered. Let us denote by h_{min} (resp. h_{max}) the minimal (resp. maximal) mesh element size and ε the desired interpolation error. Then, according to Relation (35), we define at each mesh vertex the anisotropic metric tensor $\mathcal{M}$ as:

$$\mathcal{M} = \mathcal{R}\,\widetilde{\Lambda}\,\mathcal{R}^{-1}\,,\;\text{ where }\;\widetilde{\Lambda} = diag(\widetilde{\lambda_i}\,)$$
$$\text{and }\;\widetilde{\lambda_i}\;=\min\left(\max\left(\frac{c\,|\lambda_i|}{\varepsilon}\,,\frac{1}{h_{max}^2}\right),\frac{1}{h_{min}^2}\right).$$

Introducing a minimal (resp. maximal) element size is a practical way to avoid unrealistic metrics. It also allows us to control the time stepping in the computational scheme. In other words, in view of equidistributing the interpolation error over the mesh, we have modified the scalar product that underlies the notion of distance used in mesh generation algorithms (where the local metric $\mathcal{M}$ replaces the usual Euclidean metric).

Error estimate in CFD. Physical phenomena can involve large scale variations (multi-scale phenomena, recirculation, weak and strong shocks, etc.). It is thus difficult to capture the weakest phenomena *via* mesh adaptation, and even harder to do it when, for instance in CFD, shocks are located in the flow. Capturing such weak phenomena is crucial for obtaining an accurate solution by taking into account all phenomena interactions in the main flow area.

A local error estimation can overcome this problem [5]. Relation (35) is normalized using the local absolute value of the variable u :

$$\left\| \frac{u - \Pi_h u}{|u|_\epsilon} \right\|_{\infty,K} \leq c \max_{x \in K} \max_{\overrightarrow{e} \in E_K} \langle \overrightarrow{e} \, , \, \frac{|H_u(x)|}{|u(x)|_\epsilon} \overrightarrow{e} \, \rangle \, , \tag{36}$$

where $|u|_\epsilon = max(|u|, \epsilon \, \|u\|_{\infty,\Omega})$ with $\epsilon \ll 1$ a constant. The term $\epsilon \, \|u\|_{\infty,\Omega}$ introduces a cut off to avoid numerical problems.

However, in the context of anisotropic mesh adaptation for compressible flow, capturing weak phenomena by means of Relation (36) leads to isotropically refined strong shocks. This is due to the discretization of the solution that introduces "virtual" oscillations in the parallel direction of the shock. These oscillations have a magnitude of the same order of weak phenomena. To preserve the anisotropy, we propose to filter these oscillations with the local gradient of the solution. To this end, we suggest the following error estimate:

$$\left\| \frac{u - \Pi_h u}{\gamma |u|_\epsilon + (1-\gamma) \, \overline{h} \, \|\nabla u\|_2} \right\|_{\infty,K} \leq c \max_{x \in K} \max_{\overrightarrow{e} \in E_K} \langle \overrightarrow{e} \, , \, \frac{|H_u(x)|}{\gamma |u(x)|_\epsilon + (1-\gamma) \, \overline{h} \, \|\nabla u(x)\|_2} \overrightarrow{e} \, \rangle \, , \tag{37}$$

where $\overline{h}$ is the diameter (*i.e.,* the length of its largest edge) of element K and γ is a parameter belongs to $[0, 1]$ that will be considered close to zero if strong shocks are involved in the flow.

5.3 Anisotropic mesh adaptation

In our approach, the adaptation of the current mesh is based on the specification of a discrete anisotropic metric tensor at each vertex. For these purposes, the standard Euclidean scalar product is modified according to a proper metric tensor field $\mathcal{M}$. The aim is then to generate a mesh such that all edges have a length of (or close to) one in the prescribed metric and such that all elements are almost regular. Let P be a vertex and let $\mathcal{M}(P)$ be the metric at P, the length of the edge PX with respect to $\mathcal{M}(P)$ is defined as:

$$l_{\mathcal{M}(P)}(PX) = \langle \overrightarrow{PX} \, , \, \overrightarrow{PX} \, \rangle^{\frac{1}{2}}_{\mathcal{M}(P)} = \sqrt{{}^t\overrightarrow{PX} \, \mathcal{M}(P) \, \overrightarrow{PX}} \, .$$

As the metric is not uniform over the domain, we need to consider the metrics at the edge endpoints as well as all intermediate metrics along the edge. To achieve this, we assume that an edge PX has a local parametrization $PX = P + t\overrightarrow{PX}$ and we introduce its average length as:

$$l_{\mathcal{M}}(\overrightarrow{PX}) = \int_0^1 \sqrt{{}^t\overrightarrow{PX} \, \mathcal{M}(P + t\overrightarrow{PX}) \, \overrightarrow{PX}} \; dt \, . \tag{38}$$

Assuming that the metric is normalized, the desired adapted mesh is then a *unit mesh*, as all edges must have a length close to one.

Here, we consider the generation of adapted meshes in three dimensions as a two-steps process. At first the surface mesh is adapted using local modifications [10], then the volume mesh is adapted using a constrained Delaunay algorithm [12]. Notice that, during the point insertion phase of the volume mesh generation, most of the vertices of the previous mesh are reused for CPU concerns. This could reduce interpolation errors but in practice kept vertices are mostly in non critical area.

5.4 Solution interpolation

Solution interpolation is a key point in the mesh adaptation algorithm. The aim is to recover the solution field after generating a new adapted mesh. As we have a discrete solution field, we need an interpolation scheme to transfer this information from the current mesh to the newly adapted mesh.

During the interpolation stage, two problems have to be taken into account. First, locating the new vertices in the background mesh by identifying the elements containing them. This can be solved by moving inside the (oriented) mesh by using its topology thanks to a barycentric coordinates based algorithm [11]. Once the localization has been solved, an interpolation scheme is used to extract the information from the solution field. In our case, as the solution is considered piecewise linear by elements (because the solution is defined only at the mesh vertices) we use a classical P_1 interpolation scheme.

6 SOME NUMERICAL RESULTS

To illustrate the efficiency of the proposed approach, we will now present two application examples of three-dimensional CFD simulations.

6.1 Supersonic business jet

The first example concerns the simulation of a supersonic flow around a future business jet at Mach 1.8 with an angle of incidence of 3 degrees at an altitude of $15,200$ meters. The design and conception of this future business plane (Dassault Aviation) has led to investigate the control of the sonic boom phenomenon. Beside classical studies aimed at reducing the drag and at increasing the lift of the airplane, the shape optimisation process for supersonic civil aircrafts includes another component: the need to reduce the noise at the ground level. In our case, the goal was to analyze the impact on the sonic boom of the optimisation of the front part of the airplane geometry. Numerically, this study combines a good near field flow computation (including the computation of the aircraft signature: pressure fields) and a good sonic boom prediction (far field propagation). In the preliminary stage of this project, mesh adaptation already proved to be a very efficient tool.

The variable used to adapt the mesh is the Mach number. The mesh has been adapted 9 times, every 250 time steps. Figures 1 and 2 present the final adapted mesh with the corresponding Mach number distribution for this simulation. In these figures, the Mach cones are clearly identified in front of the fuselage on the adapted mesh. The initial mesh contains $41,137$ vertices, $20,586$ boundary triangles and $216,916$ tetrahedra. The final mesh (iteration 9) contains $798,756$ vertices, $38,492$ boundary triangles and $4,714,162$ tetrahedra. The CPU times required to compute the whole simulation is 44 hours on a 600MHz workstation with 1Gb of memory. Notice that the meshing time represents only 2% of the solver CPU times.

6.2 Anisotropic ONERA M6 Wing

The second example concerns a classic numerical simulation of transonic air flow around the ONERA M6 wing. An Euler solution is computed for Mach number equal to 0.8395 with an angle of attack of 3.06 degrees. This transonic simulation case features a well-known lambda-shock. The initial mesh is a relatively coarse mesh containing $7,815$ vertices, $5,848$ boundary triangles and $37,922$ tetrahedra. The variable used to adapt the mesh is the Mach number. The mesh has been adapted 9 times, every 250 time steps. Figure 3 (resp. 4) shows the adaptation in the isotropic (resp. anisotropic) case. The final isotropic mesh (iteration 9) contains $231,113$ vertices and $1,316,631$ tetrahedra and the final anisotropic mesh contains $23,516$ vertices and $132,676$ tetrahedra. In this example, the maximal aspect ratio of the anisotropic elements is about 10. Nevertheless, the anisotropic metric leads to a dramatic reduction of the number of degrees of freedom, roughly one order less than in the isotropic case, for the same error level. The CPU times required to generate the final surface (resp. volume) mesh is 31 (resp. 132) seconds, and to compute the Euler solution over 250 time steps is $2,782$ seconds in the isotropic case, on a 3Ghz workstation. Similarly, the CPU times required to generate the surface (resp. volume) mesh is 3 (resp. 25) seconds, and to compute the Euler solution over 250 time steps is 318 seconds in the anisotropic case.

7 CONCLUSION

Continuous models for mesh adaptation are useful for the analysis of approximation errors. This paper explores the possibilities of P1-Galerkin based approximations in order to perform the analysis of CFD methods in terms of interpolation error. Variations of the criterion theoretically derived are also proposed for several numerical experiments. In a forthcoming paper we use this analysis for a larger class of objective functionals.

Acknowledgment

This work was partly supported by the Comite d'Orientation pour l'Avion Supersonique.

REFERENCES

[1] 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.

[2] D.A. Venditi, D.L. Darmofal. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *Journal of Computational Physics*, 187:22-46, 2003.

[3] I. Babuška and T. Strouboulis. *The Finite Element Method and its reliability.* Oxford Scientific Publications, New York, 2001.

[4] F. Courty, D. Leservoisier, P.-L. George, and A. Dervieux. Continuous metrics and mesh optimization. *Applied Numerical Mathematics*, 56:117-145, 2004.

[5] M.J. Castro-Diaz, F. Hecht, B. Mohammadi and O. Pironneau, Anisotropic Unstructured Mesh Adaptation for Flow Simulations. *Int. J. Numer. Meth. Fluids*, 25:475-491 (1997).

[6] P.J. Frey and F. Alauzet, Anisotropic mesh adaptation for CFD computations *Comput. Methods*

Appl. Mech. Engrg., 194:5068-5082 (2005).

[7] E. Schall, D. Leservoisier, A. Dervieux, B. Koobus, Mesh adaptation as a tool for certified computational aerodynamics. *I.J. Num. Meth. Fluids* 45:179-196, 2004

[8] A. Dervieux, D. Leservoisier, P.-L. George, Y. Coudière, About theoretical and practical impact of mesh adaptations on approximation of functions and of solution of PDE, *Int. J. Numer. Meth. Fluids*, 43:507-516, 2003

[9] F. Courty, T. Roy, B. Koobus, M. Vázquez, A. Dervieux. Error analysis for P1-exact schemes, Finite Element for Flow Problems April 4-6,2005, Swansea, Wales, UK, extended paper submitted for I.J. Num. Methods in Fluids

[10] P.J. Frey, About surface remeshing, in *Proc. of 9th Int. Meshing Rountable*, New Orleans, LO, USA, 123-136, (2002).

[11] P.J. Frey and P.L. George, *Mesh generation. Application to finite elements.* Hermès Science Publ., Paris, Oxford (2000).

[12] P.L. George, Tet meshing: construction, optimization and adaptation. in *8th International Meshing Roundtable*, South Lake Tahoe, CA, USA, (1999).

[13] B. Stoufflet, J. Periaux, F. Fezoui, A. Dervieux. 3-D Hypersonic Euler Numerical Simulation around Space Vehicles using Adapted Finite Elements, *25th AIAA Aerospace Meeting*, Reno (1987), AIAA Paper 86-0560

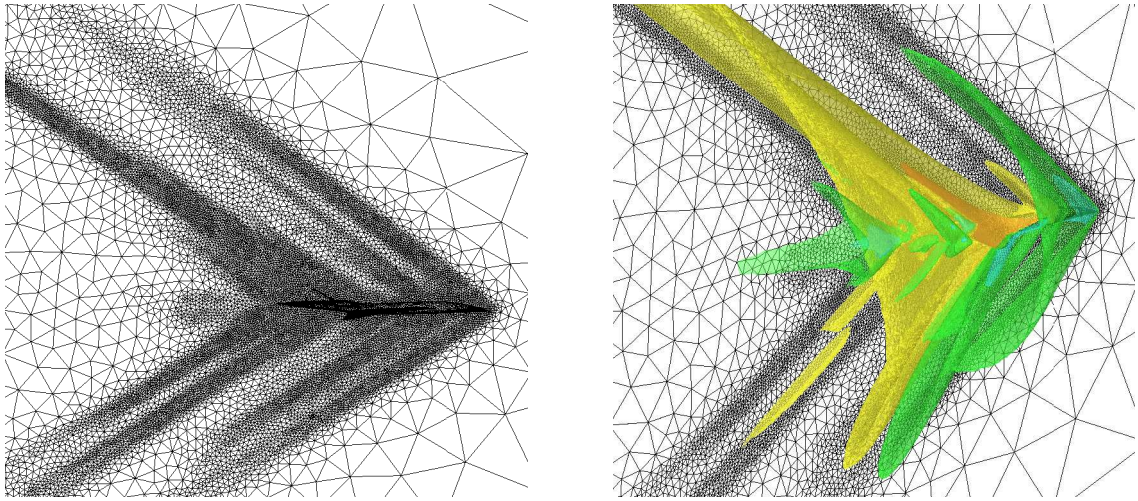

Fig.1: *Supersonic business jet test case: isotropic surface mesh at iteration 9 of the adaptation scheme. Right, isosurface of Mach number for the final solution, Mach cones are clearly identified*

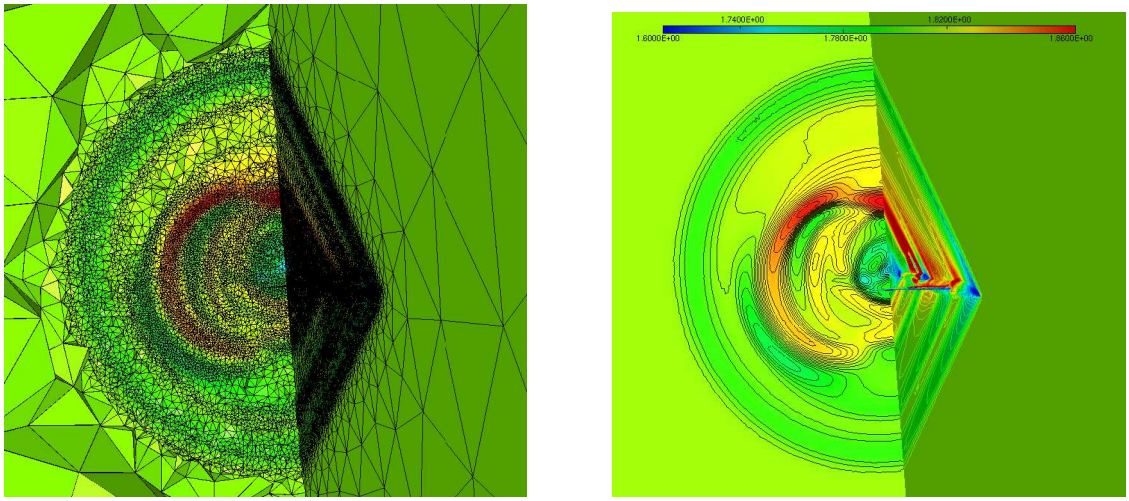

Fig.2: *Supersonic business jet test case: cut through the isotropic volume mesh at iteration 9 of the adaptation scheme. Right, isoline of Mach number in the cut plane.*

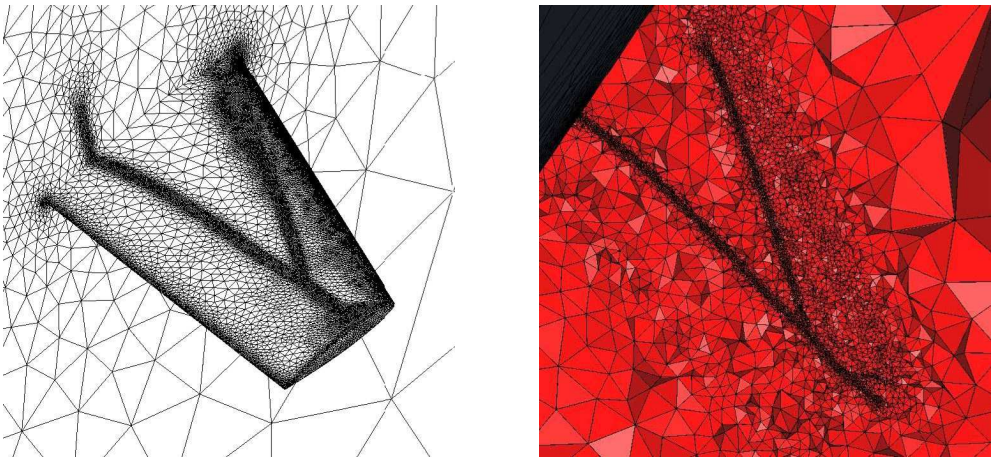

Fig.3: *ONERA M6 wing test case: isotropic surface and cut through the volume mesh at iteration 9 of the adaptation scheme.*

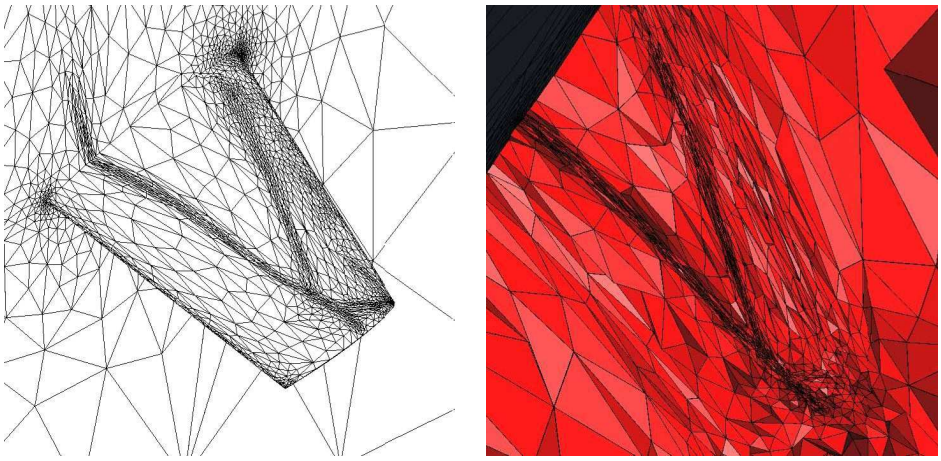

Fig.4: *ONERA M6 wing test case: anisotropic surface and cut through the volume mesh at iteration 9 of the adaptation scheme.*