V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

FULLY ANISOTROPIC GOAL-ORIENTED MESH ADAPTATION FOR UNSTEADY FLOWS

A. Belme*, A. Dervieux† and F. Alauzet††

*,†INRIA Sophia Antipolis,
2004 route de Lucioles, 06902 Sophia-Antipolis, France
e-mail: {Anca.Belme, Alain.Dervieux}@sophia.inria.fr
††INRIA Roquencourt
Domaine de Voluceau, 78150 Rocquencourt, France
e-mail: Frederic.Alauzet@inria.fr

Key words: Anisotropic mesh adaptation, Goal-oriented mesh adaptation, Unsteady flows, Euler equations, Adjoint state

Abstract. *We present a new algorithm for combining a fully anisotropic goal-oriented mesh adaptation with the transient fixed point method for unsteady problems. The minimization of the error on a functional provides both the density and the anisotropy (stretching) of the optimal mesh. This method is used for specifying the mesh for a time subinterval from the state and the adjoint. The global transient fixed point iterates the re-evaluation of meshes and the states over the whole time interval until convergence. Applications to unsteady blast-wave Euler flows are presented.*

1 INTRODUCTION

Engineering problems commonly 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 often unknown. CFD solutions may be computed with an unnecessarily large number of grid points (and associated high cost) to ensure that the outputs are computed within the required accuracy.

One of the powerful methods for increasing the accuracy and reducing the computation complexity is mesh adaptation, whose purpose 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. The technique adopted in this work is the anisotropic mesh adaptation for unsteady flows introduced by Alauzet *et al.* in [2] combined with a goal-oriented mesh adaptation method. The choice of anisotropic mesh is motivated by its strong influence on the accuracy of many CFD predictions. This technique allows (i) to significantly reduce the number of degrees of freedom, thus impacting favorably the CPU time, (ii) to automatically capture the anisotropy of the physical phenomena, and (iii) to access to high order asymptotic convergence.

Recent works have shown a fertile development of metric-based, or Hessian-based methods [11, 13, 16, 20], which rely on an ideal representation of the *interpolation error* and of the *mesh*. Getting rid of error iso-distribution and prefering L^p error minimization allow to take into account discontinuities with higher-order convergence [22].

However, these methods are limited to the minimization of some interpolation errors for some solution fields. If for many applications, this simplifying standpoint is an advantage, there are also many applications where Hessian-based adaptation is far from optimal regarding the way the degrees of freedom are distributed in the computational domain. Indeed, metric-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 a PDE. Further, in many engineering applications, a specific scalar output needs to be accurately evaluated, e.g. lift, drag, heat flux. To address this need, the *goal-oriented* mesh adaptation focuses 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* [8, 15, 23, 24]. Extracting informations concerning mesh anisotropy from an *a posteriori* estimate is a difficult task. Starting from *a priori estimates*, Loseille *et al.* proposed in [21] a fully anisotropic goal-oriented mesh adaptation technique for steady compressible Euler flows. This latter method combines goal oriented error estimate and the application of Hessian-based analysis to truncation error.

Mesh adaptation for unsteady flows is an active research field and brings an attracting increase in simulation efficency. Complexity of the algorithms is larger than for steady case: time discretisation needs also to be adapted, possibly in a manner that combines well with mesh adaptation. Meshes can be moved as in [7], locally refined [10] or rebuild

as in [2, 18]. Many error sensors, *a posteriori* estimates and goal-oriented methods have been considered in the literature, see a typical contribution in [8]. Hessian-based methods are essentially applied with a non-moving mesh system. A transient fixed-point mesh adaptation method was proposed in [2]. The Hessian criteria at the different time steps was synthetized with the metric intersection method [2, 17]. In [17] it is proposed to minimize the $L^\infty(L^p)$ norm (supremum in time of the L^p spatial norm) of the Hessian-based criterion.

This paper addresses the extension to unsteady flows of the fully anisotropic goal-oriented mesh adaptation method. To this end, we combine the anisotropic goal-oriented error estimate of [21] and the transient fixed-point mesh adaptation algorithm of [2, 17]. This work rises several methodological issues. It is necessary in particular to master the complexity in development and the computational (memory and time) cost of the unsteady adjoint.

In the sequel we propose a numerical description of the problem for the case of the unsteady Euler equations. It is followed by the introduction and description of the unsteady adjoint state solver. We give a goal-oriented *a priori* estimate for the unsteady model in Section 4. Section 5 describes the global adjoint-based transient fixed-point algorithm. Numerical application to the simulation of a blast wave and some concluding remarks end this paper.

2 UNSTEADY EULER MODEL

2.1 Euler equations

The unsteady Euler equations in the computational domain $\Omega \subset R^3$ write:

$$\Psi(W) = W_t + \nabla.\mathcal{F}(W) = 0 \quad \text{in } \Omega, \tag{1}$$

where $W = {}^t(\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables. The Euler flux $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ writes:

$$\mathcal{F}_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho Hu \end{pmatrix}, \mathcal{F}_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho Hv \end{pmatrix}, \mathcal{F}_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho Hw \end{pmatrix},$$

where ρ, p and H hold respectively for the density, the thermodynamical pressure and the total enthalpy. Symbols u, v and w stand for the Cartesian components of velocity vector $\mathbf{u} = (u, v, w)$. For a calorically perfect gas, we have $p = (\gamma - 1)(\rho E - \frac{1}{2}\rho\|\mathbf{u}\|_2^2)$, where γ is constant. We can write this system and its boundary conditions under a compact variational formulation in the functional space $W \in V = [H^1(\Omega)]^5$ as follows:

$$\forall \phi \in V, \quad (\Psi(W)\,,\,\phi) = \int_\Omega \phi W_t \,\mathrm{d}\Omega + \int_\Omega \phi\, \nabla.\mathcal{F}(W)\,\mathrm{d}\Omega - \int_\Gamma \phi\, \hat{\mathcal{F}}(W).\mathbf{n}\,\mathrm{d}\Gamma \;=\; 0\;, \tag{2}$$

where Γ is the boundary of the computational domain Ω, $\mathbf{n}$ the outward normal to Γ and the boundary flux $\hat{\mathcal{F}}$ contains the different boundary conditions, which involve inflow, outflow and slip boundary conditions.

2.2 Discrete model

We use a vertex-centered finite volume formulation applied to unstructured tetrahedral meshes. A detailed description of the version implemented in our in-house flow solver `Wolf` can be found in [3]. We assume that the computational domain Ω is approximated by a discretised domain Ω_h equipped by a tetrahedral mesh $\mathcal{H}$. To each vertex is associated a control volume or finite volume cell, denoted C_i. The boundary of C_i is defined by the rule of medians. Higher-order interpolations on both side of cell interfaces are done by applying a MUSCL type method using downstream and upstream tetrahedra. Stabilisation is obtained by introducing the cell-interface extrapolations into a HLLC approximate Riemann solver. Monotone shock capturing is obtained with a generalization of the Superbee limiter with three entries [12].

In [21], we have reformulated this particular finite-volume scheme under the form of a finite element variational formulation. Let us introduce the following approximation space:

$$V_h \;=\; \left\{\phi_h \in V \cap \mathcal{C}^0(\bar{\Omega}) \mid \phi_{h|K} \text{ is affine } \forall K \in \mathcal{H}\right\}.$$

The interpolation operator of the previous section is chosen as the usual $\mathcal{P}^1$ operator:

$$\Pi_h :\; V \cap \mathcal{C}^0(\bar{\Omega}) \rightarrow\; V_h \;\text{ such that }\; \Pi_h\varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i),$$

for all vertices $\mathbf{x}_i$ of $\mathcal{H}$. The weak discrete formulation writes:

$$\forall \phi_h \in V_h,\; \left(\Psi_h(W_h)\,,\, \phi_h\right) = 0,$$

with

$$\begin{aligned}\left(\Psi_h(W_h)\,,\, \phi_h\right) \;&=\; \int_{\Omega_h} \phi_h W_{h,t}\, \mathrm{d}\Omega_h + \int_{\Omega_h} \phi_h \nabla.\mathcal{F}_h(W_h)\, \mathrm{d}\Omega_h \\ &-\; \int_{\Gamma_h} \phi_h \hat{\mathcal{F}}_h(W_h).\mathbf{n}\, \mathrm{d}\Gamma_h \;+ \int_{\Omega_h} \phi_h\, D_h(W_h)\mathrm{d}\Omega_h\end{aligned} \tag{3}$$

with $\mathcal{F}_h = \Pi_h\mathcal{F}$ and $\hat{\mathcal{F}}_h = \Pi_h\hat{\mathcal{F}}$ and $\Gamma_h = \partial\Omega_h$. The numerical diffusion term D_h is at least a third order term with respect to mesh size, everywhere limiters do not apply. Even for shocked flows, we have found it is interesting to neglect it, see [21]. This option is also followed in this paper.

2.2.1 Time advancing

An explicit scheme is used to advance the Euler equations in time. Once the equations have been discretized in space, a set of ordinary differential equations in time is obtained:

$W_t - \Phi(W) = 0$. To discretize the previous relation, a Strong-Stability-Preserving (SSP) Runge-Kutta scheme is considered. 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. This holds for the first order Runge-Kutta scheme which is considered here:

$$W^{n+1} \;=\; W^n + \Delta t\, \Phi(W^n) \tag{4}$$

This scheme is stable for a Courant number of 1.

3 UNSTEADY ADJOINT STATE AND LAGRANGE MULTIPLIERS

Starting from an initial solution W^0 of the state equation (1), the solution at time t_n, *i.e.,* $W(t_n)$, is computed using a time integration scheme. We first consider the first order Runge-Kutta method given by Relation (4):

$$\tilde{\Psi}^n(W^n, W^{n-1}) = \frac{W^n - W^{n-1}}{\delta t^n} + \Phi(W^{n-1}) = 0, \tag{5}$$

with $\delta t^n = t_{n+1} - t_n$ the (n+1)-th time step and Φ the numerical flux function. The problem of minimizing the error on target funtional $j(W) = (g, W)$, subject to Euler system (1) can be rewritten into an unconstrained problem via the Lagrange method. The extended cost function (also known as the Lagrangian functional) reads:

$$\mathcal{L} = j - \sum_{n=0}^{N} [(W^{*,n})^T \tilde{\Psi}^n] \tag{6}$$

with $W^{*,0}, W^{*,1}, ..., W^{*,N}$ the $N+1$ vectors of Lagrange multipliers (which are the time-dependent adjoint states). A necessary condition for an extrema is that the gradient of $\mathcal{L}$ with respect to $W^0, ..., W^N$ and $W^{*,0}, ..., W^{*,N}$ vanishes. Since the states $W^1, ..., W^N$ are calculated starting from W^0 using the constraint given by Relation (5), we get easily:

$$\frac{\partial \mathcal{L}}{\partial W^{*,n}} \;=\; 0, \quad \text{for } n = 0, ..., N.$$

The Lagrange multipliers $W^{*,n}$ must now be chosen such that

$$\frac{\partial \mathcal{L}}{\partial W^{n}} \;=\; 0, \quad \text{for } n = 0, ..., N,$$

which, in case of two-levels schemes as (5), leads to:

$$\begin{cases} \dfrac{\partial j}{\partial W^N} - (W^{*,N})^T \dfrac{\partial \tilde{\Psi}^N}{\partial W^N} = 0 \\ \dfrac{\partial j}{\partial W^n} - (W^{*,n})^T \dfrac{\partial \tilde{\Psi}^n}{\partial W^n} - (W^{*,n+1})^T \dfrac{\partial \tilde{\Psi}^{n+1}}{\partial W^n} = 0, \quad \forall\ n = 0, ..., N-1. \end{cases}$$

This can be written equivalently as:

$$\left\{\begin{array}{rcl} W^{*,N} &=& \Big(\dfrac{\partial\tilde{\Psi}^N}{\partial W^N}\Big)^{-T}\Big(\dfrac{\partial j}{\partial W^N}\Big)^T \\ W^{*,n} &=& \Big(\dfrac{\partial\tilde{\Psi}^n}{\partial W^n}\Big)^{-T}\Big[\Big(\dfrac{\partial j}{\partial W^n}\Big)^T - \Big(\dfrac{\partial\tilde{\Psi}^{n+1}}{\partial W^n}\Big)^{-T}(W^{*,n+1})^T\Big], \quad \forall\ n=0,...,N-1. \end{array}\right. \tag{7}$$

Since $W^1,...,W^N$ have been calculated from the initial state W^0, the vector of Lagrange multipliers $W^{*,n}$ can be computed backwards using the whole set of flow solutions $W^0,...,W^N$. System (7) is known as the system of adjoint equations for Model (1). In this context, the Lagrange multipliers are also known as the adjoint variables. The next step is to study how the Jacobians $\frac{\partial\tilde{\Psi}^n}{\partial W^n}$ and $\frac{\partial\tilde{\Psi}^{n+1}}{\partial W^n}$ can be evaluated. In our case, for the first order Runge-Kutta scheme, we have:

$$\frac{\partial\tilde{\Psi}^n}{\partial W^n} = \frac{1}{\delta t^n}I \quad \text{and} \quad \frac{\partial\tilde{\Psi}^{n+1}}{\partial W^n} = -\frac{1}{\delta t^{n+1}}I + \frac{\partial}{\partial W^n}\Phi(W^n)$$

and the adjoint system (7) becomes:

$$\left\{\begin{array}{rcl} W^{*,N} &=& \delta t^N\Big(\dfrac{\partial j}{\partial W^N}\Big)^T \\ W^{*,n} &=& \delta t^n\Big[\Big(\dfrac{\partial j}{\partial W^n}\Big)^T + \dfrac{(W^{*,n+1})^T}{\delta t^{n+1}} - \Big(\dfrac{\partial\Phi}{\partial W^n}(W^n)\Big)^T(W^{*,n+1})^T\Big], \ \forall\ n=0,...,N-1. \end{array}\right.$$

4 ERROR ANALYSIS

4.1 Formal analysis

Let Ω be a subdomain of $\mathbb{R}^3$ with a smooth enough boundary. Let V_1 be the space of real square integrable functions defined on Ω, and $V_1^{'}$ its dual for the L^2 scalar product, which lies in $\mathcal{D}(\Omega)$, the space of usual distributions. Let $\mathcal{V}_1 = V_1 \cap \mathcal{C}^0(\bar{\Omega})$ be a subspace of V_1 of smoother functions than those of V_1. Since we are interested in systems of PDE, we consider the spaces of n-uples of functions/distributions in the previous spaces:

$$V = (V_1)^n \quad ; \quad V' = (V_1')^n \quad ; \quad \mathcal{V} = (\mathcal{V}_1)^n$$

equipped with the induced topologies.

In order to define a continuous system of PDE, we introduce a mapping $\Psi : V \to V'$:

$$w = (w_1, w_2, .., w_n) \ \mapsto \ \Psi(w) = (\Psi(w)_1, \Psi(w)_2, .., \Psi(w)_n) \tag{8}$$

We consider a continuous system of PDE called the *state equation* which is written as follows:

$$\text{Find} \ \ w \in V \ \ \text{such that} \ \ \Psi(w) = 0 \ \ \text{in} \ \ V' \ . \tag{9}$$

We can also write the state equation under a variational statement:

$$w \in V \ , \ \forall \varphi \in V \ , \ (\Psi(w), \varphi) = 0, \tag{10}$$

where the operator $(,)$ holds for a $V' \times V$ product. The functional under study is also defined on V, *i.e.,* $j : V \to \mathbb{R}$ defined by:

$$w \ \mapsto \ j(w) = (g, w) \ . \tag{11}$$

The *continuous adjoint* w^* is solution of:

$$w^* \in V \ , \ \forall \psi \in V \ , \ \left(\frac{\partial \Psi}{\partial w}(w)\psi, w^* \right) = (g, \psi) \ . \tag{12}$$

We now describe the discretized PDE system. Let V_h be a subspace of $\mathcal{V}$ of finite dimension N. A central assumption is that we can write the discrete state equation as follows:

$$w_h \in V_h \ , \ \forall \varphi \in V \ , \ (\Psi_h(w_h), \varphi) = 0.$$

In particular, we can write:

$$(\Psi_h(w), \varphi_h) - (\Psi_h(w_h), \varphi_h) = (\Psi_h(w), \varphi_h) - (\Psi(w), \varphi_h) = ((\Psi_h - \Psi)(w), \varphi_h). \tag{13}$$

For the *a priori* analysis, we assume that solutions w and w^* are sufficiently regular:

$$w \in V \cap \mathcal{C}^0(\bar{\Omega}) \ \text{ and } \ w^* \in V \cap \mathcal{C}^0(\bar{\Omega}),$$

and that we have an interpolation operator:

$$\Pi_h : V \cap \mathcal{C}^0(\bar{\Omega}) \ \to \ V_h.$$

The objective is to estimate the following approximation error on the functional:

$$\delta j = j(w) - j(w_h) \ .$$

In [21] the following *a priori* formal estimate is proposed:

$$\delta j \approx ((\Psi_h - \Psi)(w), w^*) \ . \tag{14}$$

The next session is devoted to the application of Estimate (14) to the unsteady Euler system.

4.2 Application to unsteady Euler equations

The main term of the *a priori* error estimation of δj writes:

$$\delta j = (g, W - W_h) \approx ((\Psi_h - \Psi)(W), W^*),$$

where W^* is the continuous adjoint state, solution of:

$$\frac{\partial \Psi}{\partial W} W^* = g.$$

Considering $\Psi_h(W)$ the extension of the discrete residual, we obtain a more general formulation and with the use of Π_h the interpolation operator previously defined we have:

$$\begin{aligned}(\Psi_h(W)\,,\,\phi) = \int_{\Omega_h} \Pi_h\phi\ \Pi_h W_t \,\mathrm{d}\Omega_h \quad &+ \quad \int_{\Omega_h} \Pi_h\phi\ \nabla.\,(\Pi_h\mathcal{F}(\Pi_h W)) \quad \mathrm{d}\Omega_h \\ &- \quad \int_{\Gamma_h} \Pi_h\phi\ \Pi_h\hat{\mathcal{F}}(\Pi_h W).\mathbf{n}\,\mathrm{d}\Gamma_h. \end{aligned} \tag{15}$$

where the dissipation term D_h has been neglected. From Relations (2) and (15), after integrating by parts, the *a priori* error estimate becomes:

$$\begin{aligned}\delta j \approx \int_{\Omega_h} W^*\left(\Pi_h W - W\right)_t \mathrm{d}\Omega_h \quad &+ \quad \int_{\Omega_h} \nabla W^*\left(\mathcal{F}(W) - \Pi_h\mathcal{F}(W)\right) \mathrm{d}\Omega_h \\ &- \quad \int_{\Gamma_h} W^*\left(\bar{\mathcal{F}}(W) - \Pi_h\bar{\mathcal{F}}(W)\right)).\mathbf{n}\,\mathrm{d}\Gamma_h. \end{aligned} \tag{16}$$

We observe that this estimate of δj is expressed in terms of interpolation errors for the fluxes and for the time derivative (which can be considered like a source term for Equations (1)) and in terms of the continuous functions W and W^*.

Error bound with a safety principle The integrands in (16) contain positive and negative parts which can compensate for some particular meshes. In our strategy, we prefer to avoid these parasitic effects. To this end, all integrands are bounded by their absolute values:

$$\begin{aligned}(g, W_h - W) \leq \int_{\Omega_h} |W^*|\,|(W - \Pi_h W)_t|\,\mathrm{d}\Omega_h \\ + \int_{\Omega_h} |\nabla W^*|\,|\mathcal{F}(W) - \Pi_h\mathcal{F}(W)|\,d\Omega_h \\ + \int_{\Gamma_h} |W^*|\,|(\bar{\mathcal{F}}(W) - \Pi_h\bar{\mathcal{F}}(W)).\mathbf{n}|\,d\Gamma_h. \end{aligned} \tag{17}$$

In other words, we prefer to locally over-estimate the error.

4.3 Optimal metric for the interpolation error

We recall here the continuous mesh framework [19] introduced to find the optimal mesh minimizing the interpolation error estimate for a given mesh size or a given error threshold. Indeed, it allows us to define proper differentiable optimization [1, 6] or to use the calculus of variations that is undefined on the class of discrete meshes. This framework lies in the class of metric-based methods.

Continuous mesh model and continuous linear interpolation error. A continuous mesh $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ of Ω is a Riemannian metric field [9]. For all $\mathbf{x}$ of Ω, $\mathcal{M}(\mathbf{x})$ is a symmetric tensor 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 $(h_i(\mathbf{x}))_{i=1,3} = (\lambda_i^{-\frac{1}{2}}(\mathbf{x}))_{i=1,3}$. With this definition, $\mathbf{M}$ admits the more practical local decomposition:

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

where

- the node density d is equal to: $d = (h_1h_2h_3)^{-1} = (\lambda_1\lambda_2\lambda_3)^{\frac{1}{2}} = \sqrt{\det(\mathcal{M})}$,
- the three anisotropic quotients r_i are equal to: $r_i = h_i^3\,(h_1h_2h_3)^{-1}$.

The anisotropic quotients represent the overall anisotropic ratio of a tetrahedron taking into account all the possible directions. It is a complementary measure to anisotropic ratio given by $\max_i(h_i)/\min_i(h_i)$. By integrating the node density, we define the complexity $\mathcal{C}$ of a continuous mesh which is the continuous counterpart of the total number of vertices:

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

This real-value parameter is useful to quantify the global level of accuracy of the continuous mesh $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\boldsymbol{\Omega}}$. It has been shown in [19] that $\mathbf{M}$ defines a class of equivalence of discrete meshes. The equivalence relation is based on the notion of **unit mesh** with respect to $\mathbf{M}$.

This model is also particularly well suited to the study of the interpolation error. Indeed, there exists a unique continuous interpolation error that models the (infinite) set of interpolation errors computed on the class of unit meshes. See [19] for the proof along with equivalence between discrete and continuous formulations. For a smooth function u, the continuous linear interpolate $\pi_{\mathcal{M}}u$ is a function of the Hessian H_u of u and verifies:

$$(u - \pi_{\mathcal{M}}u)(\mathbf{x}) \;=\; \frac{1}{10}\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,|H_u(\mathbf{x})|\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}))$$

$$= \frac{1}{10} d(\mathbf{x})^{-\frac{2}{3}} \sum_{i=1}^{3} r_i(\mathbf{x})^{\frac{2}{3}t} \mathbf{v_i}(\mathbf{x}) \, |H_u(\mathbf{x})| \, \mathbf{v_i}(\mathbf{x}), \tag{18}$$

where $|H_u|$ is deduced from H_u by taking the absolute values of its eigenvalues. $\pi_{\mathcal{M}}$ replaces the discrete operator Π_h in this continuous framework.

Optimal mesh. The problem of mesh adaptation consists in finding the mesh $\mathcal{H}$ of Ω that minimizes the linear interpolation error $u - \Pi_h u$ controlled in $\mathbf{L}^p$ norm. The problem is thus stated in an *a priori* way:

$$\text{Find } \mathcal{H}_{\mathbf{L}^p} \text{ having } N \text{ nodes such that } E_{\mathbf{L}^p}(\mathcal{H}_{\mathbf{L}^p}) = \min_{\mathcal{H}} \|u - \Pi_h u\|_{\mathbf{L}^p(\Omega_h)}. \tag{19}$$

As it, Problem (19) is a global combinatorial problem which turns out to be intractable practically. Indeed, both topology and vertices locations need to be optimized. This ill-posed problem can be reformulated in the continuous mesh framework [19, 20]. It is then possible to set the well-posed global optimization problem of finding the optimal continuous mesh minimizing the continuous interpolation error in $\mathbf{L}^p$ norm:

$$\text{Find } \mathbf{M}_{\mathbf{L}^p} \text{ such that } \mathcal{E}_{\mathbf{L}^p}(\mathbf{M}_{\mathbf{L}^p}) = \min_{\mathbf{M}} \left(\int_\Omega (u(\mathbf{x}) - \pi_{\mathcal{M}} u(\mathbf{x}))^p \, \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}}, \tag{20}$$

under the constraint

$$\mathcal{C}(\mathbf{M}) = \int_\Omega \sqrt{\det \mathcal{M}(\mathbf{x})} \mathrm{d}\mathbf{x} = N. \tag{21}$$

which models the total number of nodes. According to [19], if $\mathcal{H}$ is a unit mesh with respect to $\mathbf{M}$ and u is a smooth function, then the following bound holds:

$$\|u - \Pi_{\mathcal{H}} u\|_{\mathbf{L}^p(\Omega_h)} \leq \|u - \pi_{\mathcal{M}} u\|_{\mathbf{L}^p(\Omega)} = \left(\int_\Omega \left(\mathrm{trace}(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x}) |H_u(\mathbf{x})| \mathcal{M}^{-\frac{1}{2}}(\mathbf{x})) \right)^p \mathrm{d}\mathbf{x} \right)^{\frac{1}{p}}, \tag{22}$$

Writing the optimality conditions provides the unique (by convexity) optimal continuous mesh $\mathbf{M}_{\mathbf{L}^p} = (\mathcal{M}_{\mathbf{L}^p}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ solution of Problem (20) under Constraint (21):

$$\mathcal{M}_{\mathbf{L}^p} = D_{\mathbf{L}^p} \left(\det |H_u| \right)^{-\frac{1}{2p+3}} |H_u| \quad \text{with} \quad D_{\mathbf{L}^p} = N^{\frac{2}{3}} \left(\int_\Omega (\det |H_u|)^{\frac{p}{2p+3}} \right)^{-\frac{2}{3}}. \tag{23}$$

4.4 Optimal goal-oriented error estimate metric

We propose to work in the continuous mesh framework [19] in order to find the optimal mesh minimizing the *a priori* goal-oriented error estimate (17) for a given mesh size or a given error threshold.

Error model minimization. Working in this framework enables us to write Estimate (17) in a continuous form:

$$(g, W_h - W) \approx E(t, \mathbf{M}) = \int_\Omega |W^*| \, |(W - \pi_{\mathcal{M}} W)_t| \, \mathrm{d}\Omega + \int_\Omega |\nabla W^*| \, |\mathcal{F}(W) - \pi_{\mathcal{M}} \mathcal{F}(W)| \, d\Omega + \int_\Gamma |W^*| \, |(\bar{\mathcal{F}}(W) - \pi_{\mathcal{M}} \bar{\mathcal{F}}(W)).\mathbf{n}| \, d\Gamma, \tag{24}$$

where states $W = W(t)$ and $W^* = W^*(t)$ are taken at time level t and where $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is a continuous mesh defined by a Riemannian metric space and $\pi_{\mathcal{M}}$ is the continuous linear interpolate defined hereafter. For the sequel, it will be useful to split the description of the mesh into the volumic mesh, described by $\mathbf{M}$ and the surfacic mesh for the boundary, described by $\bar{\mathbf{M}}$. Further, we consider the space error for a set $\{t_0, ..., t_m\}$ of time levels. We then introduce:

$$E(\mathbf{M}, \bar{\mathbf{M}}) = \sum_k \int_\Omega g_k |(1 - \pi_{\mathcal{M}}) u_k| \mathrm{d}\Omega + \sum_k \int_\Gamma \bar{g}_k |(1 - \pi_{\bar{\mathcal{M}}}) \bar{u}_k| d\Gamma \ . \tag{25}$$

In the different couples (g_k, u_k), we account for all the integrands in the volumic integral, that is the term from the time derivative, and the 15 terms resulting from the multiplication of interpolation error of Euler fluxes by adjoint spatial derivatives. And this, for all the $m+1$ time levels. In the different couples $(\bar{g}_k, \bar{u}_k)$, we account for the five terms resulting from the multiplication of interpolation error of boundary Euler fluxes by adjoint components, this, for all $m+1$ time levels.

We can now formulate the following (continuous) mesh optimization problem:

$$\text{Find } (\mathbf{M}_{opt}, \bar{\mathbf{M}}_{opt}) = \text{Argmin}_{\mathbf{M},\bar{\mathbf{M}}} \, E(\mathbf{M}, \bar{\mathbf{M}}), \tag{26}$$

under the constraint

$$\mathcal{C}(\mathbf{M}) + \mathcal{C}(\bar{\mathbf{M}}) = N.$$

The constraint is added to the previous problem in order to bound mesh fineness. In this continuous framework, we impose the total number of nodes (in volume and in surface) to be equal to a specified positive integer N. We now detail the continuous mesh and continuous interpolation framework.

Optimal goal-oriented metric. The optimal metric is composed of a volume tensor field $\mathcal{M}_{go}$ defined in Ω and a surface one $\bar{\mathcal{M}}_{go}$ defined on Γ. We have:

- for each vertex $\mathbf{x}$ of Ω, a 3×3 matrix arising from the volume contribution of the sum of the Hessian of each component of the Euler fluxes weighted by the gradient of the adjoint state and the Hessian of the state time derivative weighted by the adjoint state:

$$\mathbf{H}(\mathbf{x}, t_n) = \sum_{n=1}^{m} \sum_{j=1}^{5} \left([\Delta t]_j(\mathbf{x}, t_n)) + [\Delta x]_j(\mathbf{x}, t_n) + [\Delta y]_j(\mathbf{x}, t_n) + [\Delta z]_j(\mathbf{x}, t_n) \right), \quad (27)$$

where

$$\begin{aligned}
[\Delta t]_j(\mathbf{x}, t_n) &= \left| W_j^*(\mathbf{x}, t_n) \right| \cdot |H((W_{j,t}))(\mathbf{x}, t_n)|, \\
[\Delta x]_j(\mathbf{x}, t_n) &= \left| \frac{\partial W_j^*}{\partial x}(\mathbf{x}, t_n) \right| \cdot |H(\mathcal{F}_1(W_j))(\mathbf{x}, t_n)|, \\
[\Delta y]_j(\mathbf{x}, t_n) &= \left| \frac{\partial W_j^*}{\partial y}(\mathbf{x}, t_n) \right| \cdot |H(\mathcal{F}_2(W_j))(\mathbf{x}, t_n)|, \\
[\Delta z]_j(\mathbf{x}, t_n) &= \left| \frac{\partial W_j^*}{\partial z}(\mathbf{x}, t_n) \right| \cdot |H(\mathcal{F}_3(W_j))(\mathbf{x}, t_n)|,
\end{aligned}$$

with W_j^* denoting the j^{th} component of the adjoint vector W^* and $H(\mathcal{F}_i(W_j))$ the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$,

- for each vertex $\mathbf{x}$ of Γ, a 2×2 matrix arising from the surface contribution:

$$\bar{\mathbf{H}}(\mathbf{x}, t_n) = \sum_{n=1}^{m} \sum_{j=1}^{5} \left| W^*(\mathbf{x}, t_n) \right| \cdot \left| \bar{H} \left(\sum_{i=1}^{3} \bar{\mathcal{F}}_i(W)(\mathbf{x}, t_n) \cdot n_i \right) \right|, \quad (28)$$

where $\mathbf{n} = (n_1, n_2, n_3)$ is the outward normal of Γ.

The standard $\mathbf{L}^1$ norm normalization is then applied independently on each metric Goal Oriented ("*go*") tensor field:

$$\begin{aligned}
\mathcal{M}_{go}(\mathbf{x}) &= C \, \det(|\mathbf{H}(\mathbf{x}, t_n)|)^{-\frac{1}{5}} |\mathbf{H}(\mathbf{x}, t_n)| \quad \text{and} \\
\bar{\mathcal{M}}_{go}(\mathbf{x}) &= \bar{C} \, \det(|\bar{\mathbf{H}}(\mathbf{x}, t_n)|)^{-\frac{1}{4}} |\bar{\mathbf{H}}(\mathbf{x}, t_n)|.
\end{aligned} \quad (29)$$

Constants C et $\bar{C}$ depends on the desired complexity N.

In [21], it is shown that C and $\bar{C}$ are the solutions of the nonlinear algebraic problem.

$$\begin{aligned}
a \, C^{-\frac{3}{5}} + b \, C^{-\frac{1}{2}} &= N \\
\bar{C} &= C \\
\text{with } a &= \int_\Omega g^{\frac{3}{5}} \det(|\mathbf{H}(\mathbf{x}, t_n)|)^{\frac{1}{5}}, \\
\text{and } b &= \int_\Gamma \bar{g}^{\frac{1}{2}} \det(|\bar{\mathbf{H}}(\mathbf{x}, t_n)|)^{\frac{1}{4}}.
\end{aligned} \quad (30)$$

Note that the metric $\bar{\mathcal{M}}_{go}(\mathbf{x})$ is 2D and needs to be transformed onto a 3D metric prior to the intersection with $\mathcal{M}_{go}(\mathbf{x})$ to get the final goal-oriented metric. This is done by setting an $h_{\max}$ size along the normal direction to the surface in $\bar{\mathcal{M}}_{go}(\mathbf{x})$ leading to $\mathcal{M}_{gosurf}$. Finally, the global optimal continuous mesh $\mathbf{M}_{opt} = (\mathcal{M}_{opt}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is defined by:

$$\mathcal{M}_{opt}(\mathbf{x}) = \begin{cases} \mathcal{M}_{go}(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega \\ \mathcal{M}_{go}(\mathbf{x}) \cap \mathcal{M}_{gosurf}(\mathbf{x}) & \text{for } \mathbf{x} \in \Gamma \end{cases} \tag{31}$$

The continuous problem (26) has been solved from an explicit optimality condition producing the optimal metric field as a function of state and adjoint. In practice, it remains to approximatively solve the coupled system of state, adjoint, and mesh-optimality. This is done with a goal-oriented adaptive loop which we describe in Section 5.

5 UNSTEADY MESH ADAPTATION ALGORITHM

5.1 Computing the goal-oriented metric

The optimal metric defined in Section 4.4 by Relations (27,28,29,31) is a function of the adjoint state, of the gradient of the adjoint state and of the Hessians of the Euler fluxes and time derivative. 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 by element solution field. In this paper, the recovery method is based on the double $\mathbf{L}^2$-projection. Its formulation along with some comparisons to other methods is available in [3]. Goal-oriented mesh adaptation requires to adapt the surface mesh of the surface 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. We use `Yams` [14] for the adaptation of the surface and the volume mesh.

5.2 Unsteady mesh adaptation model

The transient fixed-point mesh adaptation algorithm introduced in [2] consists in splitting the simulation time interval $[0, T]$ into n mesh-adaptation sub-intervals:

$$[0, T] \;=\; [0 = t_0, t_1] \cup ... \cup [t_i, t_{i+1}] \cup ... \cup [t_{n-1}, t_n].$$

The idea is to build, for each sub-interval $[t_i, t_{i+1}]$, a mesh which is adapted to the chosen time evolving features of the solution occurring during the sub-interval. This is done by means of a fixed-point iteration procedure which enables to enhance the capture of the phenomena and the mesh adaptation at these features at each new iterations. The mesh-solution couple is improved in this way until convergence. Then, the next sub-interval is treated.

The transient fixed-point mesh adaptation algorithm cannot be used efficiently when an adjoint-based criterion is adopted. Instead, we define a *global transient fixed-point mesh adaptation algorithm*. More precisely, the mesh-solution couple convergence is now global on the whole time-frame $[0, T]$. At each fixed-point iteration (α), the global mesh is represented by a n-uple of meshes $\mathcal{M}^{(\alpha)} = (\mathcal{M}_1^{(\alpha)}, ..., \mathcal{M}_n^{(\alpha)})$, each $\mathcal{M}_k^{(\alpha)}$ being used to advance the solution on the sub-interval $[t_{k-1}, t_k]$. It results in a complex iterative algorithm since forward/backward computations are performed to solve the unsteady state/adjoint systems.

The following steps are carried out:

1. Set α=0 and define an initial global mesh $\mathcal{M}^{(0)}$ for $[0, T]$.
2. Compute state $W^{(\alpha)}$ on time interval $[0, T]$ with $\mathcal{M}^{(\alpha)}$, *i.e.,* using $\mathcal{M}_k^{(\alpha)}$ for advancing $W^{(\alpha)}$ from t_{k-1} to t_k.
3. For decreasing k, *i.e.* $k = n, n-1, ..., 1$:
   - compute backwards the adjoint state $W_k^{*,(\alpha)}$ on interval $[t_k, t_{k-1}]$ (in practice $[t_k, t_{k-1}]$ contains several time steps)
   - compute from $W_k^{(\alpha)}$ and $W_k^{*,(\alpha)}$ the new individual mesh $\mathcal{M}_k^{(\alpha+1)}$ corresponding to time sub-interval k.
4. Let $\mathcal{M}^{(\alpha+1)} = (\mathcal{M}_1^{(\alpha+1)}, ..., \mathcal{M}_n^{(\alpha+1)})$.
5. Repeat from step 2, with $\alpha = \alpha + 1$.

Memory consumption is saved by storing the state solution $W^{(\alpha)}$ only at some checkpoints defined by user. We have chosen to store $W^{(\alpha)}$ at beginning of sub-intervals $[t_{k-1}, t_k]$ and recompute it at the beginning of Step 4.

The error carried out by the tranfers between two different meshes is mastered by using the conservative transfer introduced in [4].

The fixed-point iteration is stopped when the deviation between two successive solutions at time 0 is sufficiently small.

6 NUMERICAL EXPERIMENTS

In this section we present some preliminary results on the 2D test case of a channel flow past a semi-circular object, shown in Figure (1).
We consider a "blast-like" initialization inside a circle of radius $r_0 = 0.15$ around $x_0 = (1.2, 0.0)$, given by: $\rho = 10.0$, $v = (0, 0)$ and $e = 25.0$.

The cost function j was the impulse over the target surface S in Figure (1):

$$j(W) = \frac{1}{2} \int_S (p - p_\infty)^2 ds.$$

The goal-oriented mesh adaptation technique previously defined, of deriving an optimal mesh to observe the functional has been applied to Euler model and the flowfields were integrated in time using the first-order Runge-Kutta scheme. The presence of multiple shock reflections is clearly evident. Slip boundary conditions has been considered involving the rebound of the shock wave.

Backwards progression of the adjoint state is illustrated in Figure (2). When adjoints are zero no mesh adaptation is performed since the metric defined to prescibe the mesh is constructed of terms weigthed by the adjoint and his gradient (cf. Section 4.4). Hence the flowfield is advanced at least until the target surface is reached.

The evolution of the adaptive meshes sequence is shown in Figure (3). We can observe that the chosen area of refinement by the goal oriented error estimate evolves during the computation depending on the information currently available.

Figure 1: Channel flow 2D mesh

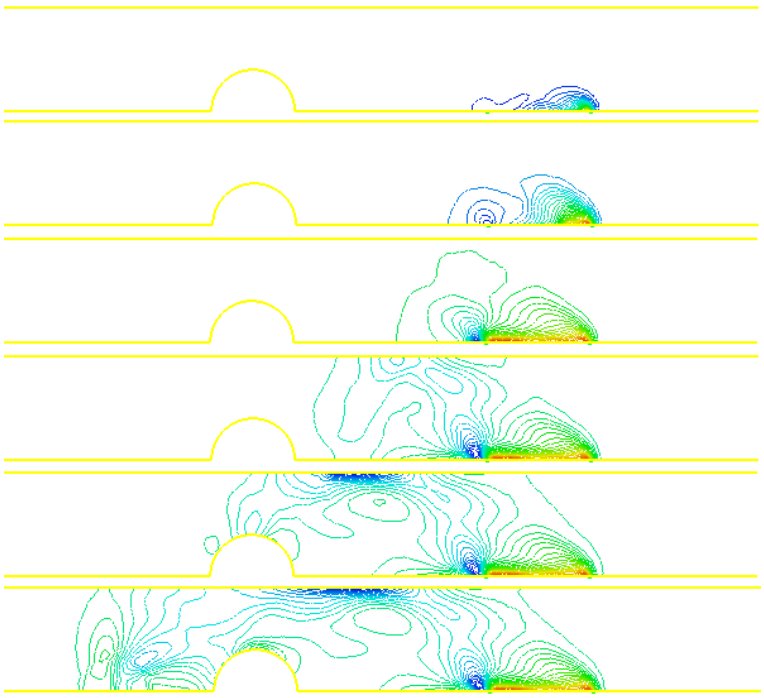

Figure 2: Adjoint solutions evolving backwards in time

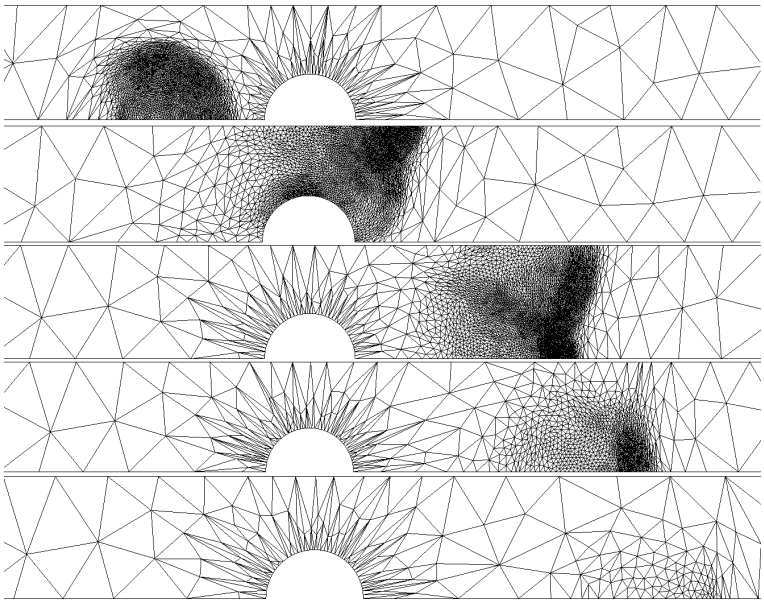

Figure 3: Evolution of the meshes in time

7 CONCLUSIONS

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 methods specifies both mesh density and anisotropy by variational calculus. Extension to unsteadiness is applied in an implicit mesh-solution coupling which needs a non-linear iteration, the fixed point. In contrast to the Hessian-based fixed-point, the iteration covers the whole time interval, including forward steps for evaluating the state and backward ones for the adjoint. The new algorithm is applied to a blast wave Euler test case and shows on this calculation the favourable behavior expected from an adjoint-based method, that is an automatic selection of the mesh necessary for the target output. Several important issues were difficult to address in this short paper and will be addressed in a longer publication. Among them, the strategies for choosing the splitting in time sub-intervals and the accurate integration of time errors in the mesh adaptation process with a more general formulation of the mesh optimisation problem is examined in [5].

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

[4] F. Alauzet and M. Mehrenberger. P1-conservative solution interpolation on unstructured triangular meshes. RR-6804, INRIA, January 2009.

[5] F. Alauzet and G. Olivier. An L^p-L^∞ space-time anisotropic mesh adaptation strategy for time dependent problems. In *Proceedings of ECCOMAS CFD*, 2010.

[6] 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.

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

[8] 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.

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

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

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

[12] 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.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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*, 2010. Submitted.

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

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

[20] 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.

[21] 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.

[22] 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.

[23] D.A. Venditti and D.L. Darmofal. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *J. Comput. Phys.*, 176(1):40–69, 2002.

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