Anisotropic Norm-Oriented Mesh Adaptation for A Compressible Inviscid Flow

Adrien Loseille *
GAMMA3 Team, INRIA Paris Rocquencourt, France

Alain Dervieux†
ECUADOR Team, INRIA Sophi-Antipolis, France

Frédéric Alauzet ‡
GAMMA3 Team, INRIA Paris Rocquencourt, France

The paper gives an unified formalism that encompasses the two most common mesh adaptation strategies: Hessian-based and goal-oriented. The first one is based on the control of the interpolation error of a solution field. The second one relies on the control of the approximation error of a scalar-output functional. Both of them have been widely used in aeronautics and derived in an anisotropic context by using a metric-based approach. If Hessian-based mesh adaptation is completely generic, it does not account for discretization error of the PDE at hand, contrary to the goal-oriented approach. The scope of this paper is to extend metric-based mesh adaptation to control a norm of the approximation error. This allows us to control simultaneously multiple functionals of interest as lift, drag, moment, without the need to solve multiple adjoint states. The procedure is based on the derivation of a corrector term that is then used as a functional for adjoint based-mesh adaptation. The estimate is derived within the continuous mesh framework, yielding naturally a fully anisotropic estimate.

Introduction

Adaptive methods in aeronautics has been used for many different purposes. The first one is generally to improve the prediction of complex phenomena (sonic-boom, contact discontinuity, blast, vortices, ...) while minimizing the CPU cost. Then, it may be used to guarantee the optimal (second) order of convergence of the numerical scheme, especially when discontinuities (shocks waves) are present in the flow field. In addition, adaptivity is also concerned with the assessment of the numerical solution. We distinguish the following class of adaptive methods according to these purposes.

A first set of methods is based on the minimization of the interpolation error of one or several sensors depending on the CFD solution.[5,13,22,24,31,41,50,55,56] Given a numerical solution W_h, a solution of higher regularity $R_h(W_h)$ is recovered, so that the following interpolation error estimate[14,38] hold:

$$\|R_h(W_h) - \Pi_h R_h(W_h)\|_{\mathbf{L}^p} \leq N^{-\frac{2}{3}} \left(\int_\Omega \det \left(|H_{R_h(W_h)}| \right)^{\frac{p}{2p+3}} \right)^{\frac{2p+3}{3p}}$$

where $H_{R_h(W_h)}$ is the Hessian of the recovered solution and N an estimate of the desired number of nodes. If anisotropic mesh prescription is naturally deduced in this context, interpolation-based methods do not take into account the features of the PDE. However, in some simplified context and assumptions (elliptic PDE, specific recovery operator), we have:

$$\|W - W_h\| \leq \frac{1}{1-\alpha} \|R_h(W_h) - \Pi_h R_h(W_h)\| \text{ with } \alpha > 1\,,$$

so that good convergence to the exact solution may be observed.[39] Indeed, if $R_h(W_h)$ is a better approximate of W in the following meaning:

$$\|W - W_h\| \ \leq \ \frac{1}{1-\alpha} \|R_h(W_h) - W_h\| \ \text{ where } \ 0 \leq \alpha < 1,$$

*Researcher, Adrien.Loseille@inria.f
†Researcher, Alain.Dervieux@inria.fr
‡Researcher, Frederic Alauzet@inria.f

and if the reconstruction operator R_h has the property:

$$\Pi_h R_h(W_h) = W_h,$$

we can then bound the approximation error of the solution by the interpolation error of the reconstructed function $R_h(W_h)$:

$$\|W - W_h\| \leq \frac{1}{1-\alpha}\|R_h(W_h) - \Pi_h R_h(W_h)\|.$$

Note that from a practical point of view, $R_h(W_h)$ is never recovered, only its first and second derivatives are estimated. Standard recovery techniques include least-square or green formula, and the ZZ operator.

A second set of methods tends to couple adaptivity with the assessment of the numerical prediction of the flow. Goal-oriented optimal methods,[29,33,40,51,57] minimize the error committed on the evaluation of a scalar functional. A usual functional is the observation of the pressure field on an observation surface γ:

$$|j(W) - j_h(W_h)|, \text{ with } j(W) = \int_\gamma \left(\frac{p - p_\infty}{p_\infty}\right)^2,$$

W (respectively W_h) is (respectively numerical) solution of the Euler equation. They do take into account the features of the PDE, through the use of an adjoint state that gives the sensitivity of W to observed j. In,[10,36] in order to be able to solve the goal-oriented mesh optimization problem, we introduce an a priori analysis which restricts to the main asymptotic term of the local error. if a super-convergence,[27,28] of $|j(W) - j_h(W_h)|$ may be observed in some cases, goal-oriented optimal methods are specialized for a given output, and in particular do not provide a convergent solution field. Indeed, the convergence of $\|W - W_h\|$ is not predicted. In addition, if the observation of multiple functionals is possible (trough multiple adjoint states), the optimally of the mesh and the convergence properties of the approximation error may be lost.

In each case, the aforementioned adaptive strategies address specifically one goal. Consequently, it is still a challenge to find an adaptive framework that encompass all the desired requirements: anisotropic mesh prescription, asymptotic optimal order of convergence, assessment of the convergence of the numerical solution to the continuous one, control of multiple functionals of interest, ... This paper is a contribution with a first attempt to formally predict all the different requirements. Our approach is based on the design of a norm-oriented optimal method, which takes into account the PDE features, and produces an approximate solution field which does converge to the exact one. This is done by estimating a residual term $\Pi_h W - W_h$. This term naturally arise when the functional of interest is the norm $\|\Pi_h W - W_h\|_{\mathbf{L}^2}$. The estimate is then used as a functional with the standard goal-oriented approach. To do so, we derive some correctors that estimate the implicit error. We also discuss the two standard strategies with *a priori* and *a posteriori* estimates. Contrary to the goal-oriented mesh adaptation, the functional may be now any function of approximation error. Consequently, we can observed functional of interest that the difference between exact and numerical. In addition, multiple functional of interest can be observed simultaneously. For instance, the norm-functional can be:

$$j(W, W_h) = (\text{drag}(W) - \text{drag}(W_h))^2 + (\text{lift}(W) - \text{lift}(W_h))^2.$$

By linearizing the right of side, we see that the (corrector) estimate for the norm-functional depends only of $\Pi_h W - W_h$ and will produce only one right hand side for the goal-oriented estimation.

The paper is organized as follows. In Section I, we then derive formally an error analysis for multi-scale, goal-oriented and norm-oriented mesh adaptation. In Section II, we recall the main flow features used in the numerical examples. In Section III, we derive correctors for the case of the Euler equations within a linear and non linear setting. In Section III, we illustrate the corrector along with the norm oriented approach on 3D examples.

I. Flow and mesh models

I.A. Flow equations

The compressible Euler equations for mass, momentum and energy conservation reads (with no source terms):

$$\left\{\begin{array}{rcl} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) & = & 0\,, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p & = & 0, \\ \dfrac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p)\mathbf{u}) & = & 0, \end{array}\right.$$

where ρ denotes the density, $\mathbf{u}$ the velocity, e the total energy per mass and p the pressure. This system can be rewritten under vectorial form:

$$W_t + F_1(W)_x + F_2(W)_y + F_3(W)_z = 0,$$

where W is the nondimensionalized conservative variables vector:

$$W = (\rho, \rho u, \rho v, \rho w, \rho E)^T$$

$\mathbf{F}(W) = (F_1(W), F_2(W), F_3(W))$ are the convective (Euler) flux functions:

$$\begin{array}{rcl} F_1(W) & = & (\rho u,\ \rho u^2 + p,\ \rho uv,\ \rho uw,\ u(\rho E + p))^T \\ F_2(W) & = & (\rho v,\ \rho uv,\ \rho v^2 + p,\ \rho vw,\ v(\rho E + p))^T \\ F_3(W) & = & (\rho w,\ \rho uw,\ \rho vw,\ \rho w^2 + p,\ w(\rho E + p))^T. \end{array}$$

A weak formulation of this system writes for $W \in V = \left[H^1(\Omega)\right]^5$ as follows:

$$\forall \phi \in V, \quad (\Psi(W)\,,\, \phi) = \int_\Omega \nabla \phi \cdot \mathcal{F}(W)\ \mathrm{d}\Omega + \int_\Gamma \phi\, \bar{\mathcal{F}}(W).\mathbf{n}\ \mathrm{d}\Gamma\ =\ 0, \tag{1}$$

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 boundary conditions.

I.B. Spatial discretization

Equation (1) is discretized by a vertex-centered upwind finite-volume formulation applying to unstructured tetrahedrizations. The interested reader is invited to find a detailed presentation in.[2] For the sake of the variational analysis, it is interesting to present is as a stabilization of the Galerkin approximation.

Let $\mathcal{H}$ be a tetrahedral mesh of Ω. We denote by Ω_h and Γ_h the linear approximate of Ω and Γ defined by $\mathcal{H}$. Let us introduce the following approximation space:

$$V_h\ =\ \left\{\phi_h \in V \cap \mathcal{C}^0\ \middle|\ \phi_{h|K} \text{ is affine } \forall K \in \mathcal{H}\right\}.$$

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

$$\Pi_h :\ V \cap \mathcal{C}^0 \rightarrow\ V_h\ ;\ \Pi_h \varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i),\ \forall\, i\ ,\ \text{vertex}.$$

The weak discrete formulation writes:

$$\begin{array}{c} \forall \phi_h \in V_h,\ (\Psi_h(W_h)\,,\, \phi_h) = 0, \\ (\Psi_h(W_h)\,,\, \phi_h) = \displaystyle\int_{\Omega_h} \nabla \phi_h \cdot \mathcal{F}_h(W_h)\ \mathrm{d}\Omega_h + \int_{\Gamma_h} \phi_h \bar{\mathcal{F}}_h(W_h).\mathbf{n}\ \mathrm{d}\Gamma_h\ =\ 0 \\ \mathcal{F}_h = \Pi_h \mathcal{F}\ ;\ \bar{\mathcal{F}}_h = \Pi_h \bar{\mathcal{F}}. \end{array} \tag{2}$$

Taking as in (2) the $\mathcal{P}^1$-interpolation of the fluxes $\mathcal{F}_h$ as a discretisation principle. This produces a finite-element scheme which is identical to the central-differenced finite-volume scheme built on the so-called median dual cells, built around vertices by limiting them by plans through mid-edges, face centroids, element centroids. In practice,

this family of Mixed-Element-Volume schemes cannot be used in a non-dissipative purely centered version. In,[17,47] MUSCL versions are described and analysed. For our analysis, we shall simply consider that the scheme under study is enriched with artificial stabilisation terms. We write this as follows:

$$\forall \phi_h \in V_h, \quad \int_{\Omega_h} \nabla \phi_h \cdot \mathcal{F}_h(W_h)\, \mathrm{d}\Omega_h + \int_{\Gamma_h} \phi_h \bar{\mathcal{F}}_h(W_h).\mathbf{n}\, \mathrm{d}\Gamma_h = -\int_{\Omega_h} \phi_h\, D_h(W_h)\mathrm{d}\Omega_h,$$

It is now useful to introduce the linearised operators $\mathcal{A}$ and $\mathcal{A}_h$ expressed in terms of Jacobians of $\mathcal{F}$ and $\bar{\mathcal{F}}$ computed at W, resp. $\mathcal{F}_h$ and $\bar{\mathcal{F}}_h$ computed at W_h:

$$\begin{aligned} &\forall W \in V \quad , \quad \forall \delta W \in V \quad , \quad \mathcal{A}(W)\delta W \in (V)' \quad \text{and} \quad \forall \phi \in V, \\ &(\mathcal{A}(W)\delta W\, ,\, \phi) = \int_\Omega \nabla \phi \cdot \frac{\partial \mathcal{F}}{\partial W}(W)(\delta W)\, \mathrm{d}\Omega + \int_\Gamma \phi \frac{\partial \bar{\mathcal{F}}(W)}{\partial W}(\delta W).\mathbf{n}\, \mathrm{d}\Gamma = 0 \end{aligned} \tag{3}$$

$$\begin{aligned} &\forall W_h \in V_h \quad , \quad \forall \delta W_h \in V_h \quad , \quad \mathcal{A}_h(W_h)\delta W_h \in (V_h)' \quad \text{and} \quad \forall \phi_h \in V_h, \\ &(\mathcal{A}_h(W_h)\delta W_h\, ,\, \phi_h) = \int_{\Omega_h} \nabla \phi_h \cdot \frac{\partial \mathcal{F}_h}{\partial W}(W_h)(\delta W_h)\, \mathrm{d}\Omega_h + \int_{\Gamma_h} \phi_h \frac{\partial \bar{\mathcal{F}}_h(W_h)}{\partial W}(\delta W_h).\mathbf{n}\, \mathrm{d}\Gamma_h = 0 \end{aligned} \tag{4}$$

which both we assume to be invertible. We shall use in the sequel the notations $\mathcal{A}^{-1}RHS$ and resp. $\mathcal{A}_h^{-1}RHS_h$ for the results of solving the corresponding systems with RHS and resp. RHS_h as right-hand sides. According to,[47] the diffusion term is of higher order as soon as it is applied to the interpolation of a smooth enough field W on a sufficiently regular mesh:

$$|\int_{\Omega_h} \phi_h\, D_h(W_h)\mathrm{d}\Omega_h| \;\leq\; h^3 K(W) |\phi_h|_{L^2}\ .$$

As a result, the dissipation term will be neglected in the estimates of the sequel.

I.C. Continuous mesh framework formalism

We use in the sequel the continuous mesh framework to drive our analysis. In,[37] we prove that any mesh can be represented by a continuous Riemannian metric field $\mathcal{M}$. The link between continuous mesh and discrete mesh is based on the unit mesh concept.[25] Given a Riemannian metric field $\mathcal{M}$, a unit-mesh is a mesh having length computed in $\mathcal{M}$ and volume computed in $\mathcal{M}$ close to one:

$$\begin{aligned} &\text{for all edges of } \mathbf{e} = \mathbf{AB}, \qquad \ell_{\mathcal{M}}(\mathbf{e}) = \int_0^1 \sqrt{{}^t\mathbf{AB}\mathcal{M}((1-t)A+tB)\mathbf{AB}\mathrm{d}t} \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right], \\ &\text{for all elements } K, |K|_{\mathcal{M}} \approx \frac{\sqrt{2}}{12}. \end{aligned} \tag{5}$$

From a practical point of view, generating a anisotropic unit mesh $\mathcal{H}$ with respect to $\mathcal{M}$ requires to use any anisotropic mesh generators, see[11,15,16,21,26,33,35,41,49,54] in 3D. Conversely, given a mesh $\mathcal{H}$, the following metric field is a continuous representative of $\mathcal{H}$:

$$\mathcal{M}_P = \exp\left(\frac{\sum_{P\in K} |K| \ln(\mathcal{M}_K)}{\sum_{P\in K} |K|}\right),$$

where P is a vertex of $\mathcal{H}$ and $\mathcal{M}_K$ is the unique metric of element K, and $|K|$ the volume of K. Consequently, if u_h denotes a discrete quantity computed on a given mesh, we use equivalently the notation $u_{\mathcal{M}}$, that represents the same quantities represented on any unit mesh with respect to $\mathcal{M}$. In the case of the interpolation error, there is a strict equivalence between continuous and discrete interpolation error.[38] The parametrization of a mesh by $\mathcal{M}$ instead of h is advantageous for *a priori* analysis with anisotropic mesh. Indeed, there exist also quantities of interest as the density, anisotropic ratios, differentiation that are well defined on $\mathcal{M}$.

II. Three mesh adaptation heuristics

II.A. Hessian-based multi-scale adaptation

Let us consider a mesh parametrised by a metric $\mathcal{M}$, according to the continuous mesh theory. A Hessian-based adaptation will rely on the choise of a *sensor* s depending on the state variable W. The P^1 interpolation error $\Pi_{\mathcal{M}} s - s$

can be expressed in terms of second derivatives of s and of metric $\mathcal{M}$:

$$\Pi_{\mathcal{M}} s - s \approx \pi_{\mathcal{M}} s - s$$

where the expression $\pi_{\mathcal{M}} s - s$ holds for the following continuous approximation of the interpolation error:

$$\pi_{\mathcal{M}} s - s \;\equiv\; \text{trace}(\mathcal{M}^{-\frac{1}{2}} H(s) \mathcal{M}^{-\frac{1}{2}}), \tag{6}$$

where $H(s)$ is the Hessian of s. Minimizing $\|\Pi_{\mathcal{M}} s - s\|_{\mathbf{L}^1}$ for a given number N of vertices provides an optimal interpolation-based metric:

$$\mathcal{M}^{opt}_{inter}(s) = \underset{|\mathcal{M}|=N}{\arg\min}\, \text{trace}(\mathcal{M}^{-\frac{1}{2}} H(s) \mathcal{M}^{-\frac{1}{2}})$$

Then:

$$\mathcal{M}^{opt}_{inter}(s) = \mathcal{K}(H_s) \;\text{ with }\; \mathcal{K}(H_s) = D(H_s)\,(\det|H_s|)^{\frac{-1}{3}}\,|H_s| \;\text{ and }\; D(H_u) = N\left(\int_\Omega (\det|H_s|)^{\frac{1}{3}}\right)^{-1} \tag{7}$$

where $D(H_s)$ is a global normalization term set to obtain a continuous mesh with complexity N and $(\det|H_s|)^{\frac{-1}{3}}$ is a local normalization field. Note that replacing the optimal metric in the L^1 norm shows that second-order convergence is obtained for smooth contexts.[38] We can also extend this approach to non-smooth functions, see.[20]

Statement (7) defines directly a continuous optimal metric. In practice, solving (7) is done approximatively, *i.e.* in a discrete context with a couple (mesh,solution) denoted $(\mathcal{H}_h, W_h)$ and iteratively through the following fixed point:
1.- compute state W_h on mesh $\mathcal{H}_h$,
2.- compute sensor $s_h = s(W_h)$ and optimal metric $\mathcal{M}^{opt}_{inter}(s_h) = \mathcal{K}(H_h(s_h))$,
3.- generate a new mesh $\mathcal{H}_h = \mathcal{H}_{\mathcal{M}^{opt}_{inter}(s_h)}$ and go to 1, until convergence

in which the continuous Hessian of s is replaced by an approximate Hessian $H_h(s_h)$, evaluated by a patch-recovery approximation, defined in.[45]

II.B. Goal-oriented multi-scale adaptation

The goal-oriented analysis relies on the minimization of the error committed on a scalar output functional j which we simplify as follows:

$$j(W) - j(W_h) \approx (\frac{\partial j}{\partial W}(W), W - W_h) = (g, W - W_h). \tag{8}$$

We recall in short the results of the analysis in.[40] Introducing the adjoint state $W^* = (\mathcal{A}^{-1})^* g \equiv \mathcal{A}^{-*} g$, we have:

$$\begin{aligned}(g, W_h - W) \;&\leq\; \int_{\Omega_h} |\nabla W^*|\,|\mathcal{F}(W) - \Pi_h \mathcal{F}(W)|\,d\Omega_h \\ &+\; \int_{\Gamma_h} |W^*|\,|(\bar{\mathcal{F}}(W) - \Pi_h \bar{\mathcal{F}}(W)).\mathbf{n}|\,d\Gamma_h.\end{aligned} \tag{9}$$

The boundary term is neglected.

$$j(u) - j(u_h) \approx (\nabla(\mathcal{A}^{-*} g), \Pi_h \mathcal{F} - \mathcal{F}). \tag{10}$$

The above right-hand side transforms in:

$$(\nabla(\mathcal{A}^{-*} g), \Pi_h \mathcal{F} - \mathcal{F}) \approx (1, \text{trace}(\mathcal{M}^{-1/2}\,\nabla(\mathcal{A}^{-*} g) \cdot |H(\mathcal{F}(W))|\,\mathcal{M}^{-1/2})),$$

where $\mathcal{M}$ is a metric field representing the current mesh. Then

$$\mathcal{M}^{opt}_{goal} = \underset{|\mathcal{M}|=N}{\arg\min}\, \text{trace}(\mathcal{M}^{-1/2}\,|\nabla(\mathcal{A}^{-*} g)| \cdot |H(\mathcal{F}(W))|\,\mathcal{M}^{-1/2}))$$

and $|\nabla(\mathcal{A}^{-*} g)| \cdot |H(\mathcal{F}(W))|$ is a positive combination of symmetric matrices. Similarly to (7), we get:

$$\mathcal{M}^{opt}_{goal}(W) = \mathcal{K}(|\nabla(\mathcal{A}^{-*} g)| \cdot |H(\mathcal{F}(W))|). \tag{11}$$

This time, the optimal metric is not explicitly given by (11) since W is the solution of an EDP solved on the continuous mesh $\mathcal{M}^{opt}_{goal}$. In order to adapt the mesh, we have to discretise and converge the following loop:
1.- compute state W_h on mesh $\mathcal{H}_h$,
2.- compute adjoint state W^*_h,
3.- compute optimal metric $\mathcal{M}^{opt}_{goal}(W_h)$,
4.- generate a new mesh $\mathcal{H}_h = \mathcal{H}_{\mathcal{M}^{opt}_{inter}(s)}$ and go to 1, until convergence.

II.C. Norm-oriented adaptation

We are now interested by the minimization with respect to the metric $\mathcal{M}$ of a semi-norm of the deviation $W - W_h$. This is examplified here by the usual L^2 norm:

$$j(\mathcal{M}) = ||W - W_{\mathcal{M}}||^2_{L^2(\Omega)}. \tag{12}$$

The stationarity condition writes:

$$j'(h).\delta\mathcal{M} = (W - W_{\mathcal{M}}, \frac{\partial}{\partial\mathcal{M}}(W - W_{\mathcal{M}}).\delta\mathcal{M}). \tag{13}$$

Let g_{opt} be a good approximation of the error at optimum:

$$g_{opt} \approx W - W_{\mathcal{M}_{opt}} \tag{14}$$

then $j'(\mathcal{M}).\delta\mathcal{M}$ is small when $(g_{opt}, \frac{\partial}{\partial\mathcal{M}}(W - W_{\mathcal{M}}).\delta\mathcal{M})$ is small. Reciprocally, if we find for some $\bar{g}$ a $\mathcal{M}$ such that:

$$(\bar{g}, \frac{\partial}{\partial\mathcal{M}}(W - W_{\mathcal{M}}).\delta\mathcal{M}) \quad \text{is small,} \tag{15}$$

$$\bar{g} = W - W_{\mathcal{M}}, \tag{16}$$

then we have solved (13).

For satisfying (15), we minimise the functional $(\bar{g}, W - W_{\mathcal{M}})$. It is minimum for $\mathcal{M} = \overline{\mathcal{M}}$ with :

$$\overline{\mathcal{M}}(W) = \mathcal{K}(|\nabla(A^{-*}\bar{g})| \cdot |H(\mathcal{F}(W))|). \tag{17}$$

In practice, as for the previous algorithms, we first discretise (17). Then the unknown g_{opt}, together with the other unknowns is iteratively advanced in the fixed point iteration:
1.- compute state W_h on $\mathcal{H}_h$,
2.- compute an approximation g_{iter} of $W - W_h$
3.- compute adjoint state $W^* = A^{-*}g_{iter}$,
4.- compute optimal metric $\overline{\mathcal{M}}(W_h)$,
5.- generate a new mesh $\mathcal{H}_h = \mathcal{H}_{\overline{\mathcal{M}}(W_h)}$ and go to 1, until convergence.

The next section describes how to compute the approximation g_{iter} to $W - W_h$, which we naturally call a corrector.

III. Correctors for the Euler equations

In this section, we propose an approximate corrector answering to the criterion (14).

III.A. Finer-grid defect correction corrector for the PDE solution

According to (14) a corrector W'_h approximates the error $W - W_h$ between the exact PDE solution and its approximate on a so-called current mesh $\mathcal{M}_h$. The central question is the computational cost which we are ready to invest for the evaluation of the corrector. A typical option is to bound the cost of evaluation of the corrector to the cost of evaluation of W_h itself. For example we accept to obtain W'_h by solving a discrete system on the current mesh. This cost can be further reduced by using coarser-grid evaluations, which we do not consider here.

Of course we also need that the corrector is accurate enough. The corrector needs be an usuable evaluation of the approximation error. To fix the ideas, we could specify that the error for the corrected approximation $W - W_h - W'_h$ be two times smaller than $W - W_h$. When the approximation of W by W_h is far from mesh convergence, probably we have no chance for evaluating the accuracy of a corrector based on mesh $\mathcal{M}_h$. Let us assume, at the contrary, that the approximation on $\mathcal{M}_h$ is in its asymptotic mesh convergence phase. Then this will be also true for a strictly two-times finer embedding mesh $\mathcal{M}_{h/2}$, and for our second-order accurate scheme applied to a smooth enough problem, we would have:

$$\Psi_h(W_h) = 0 \ \text{ in } \ V'_h, \quad \Psi_{h/2}(W_{h/2}) = 0 \ \text{ in } \ V'_{h/2}, \qquad \Rightarrow \quad W - W_{h/2} \approx \frac{1}{4}(W - W_h). \tag{18}$$

which suggests the following corrector:

$$\frac{4}{3}(W_{h/2} - W_h) \approx W - W_h.$$

Note that the following also holds:

$$\frac{4}{3}(\Pi_h W_{h/2} - \Pi_h W_h) \approx \Pi_h W - W_h. \tag{19}$$

The computational cost of $W_{h/2}$ is a few times larger than expected. We first linearise the finer-grid equation as follows:

$$\frac{\partial \Psi_{h/2}}{\partial W}(W_h)(W_{h/2} - W_h) \approx \Psi_{h/2}(W_{h/2}) - \Psi_{h/2}(W_h)$$

in which we use that $V_h \subset V_{h/2}$. The first term of the RHS sum is zero. Let $R_{h/2\to h}$ the transfer from $V_{h/2}$ to V_h by the accumulation to coarse vertices weighted by barycentric coefficients. The fine-coarse linear defect-correction system writes:

$$\frac{\partial \Psi_h}{\partial W}(W_h)(\bar{W}'_{h,LDC}) \approx -R_{h/2\to h}\Psi_{h/2}(W_h).$$

This is exactly the usual coarse-grid corrector used in linear multi-grid cycles. It has also been used in[1] for building accurate correctors. A nonlinear variant can be defined by avoiding the differentiation of the residual Ψ:

$$\Psi_h(W_h + \bar{W}'_{h,DC}) \approx -R_{h/2\to h}\Psi_{h/2}(W_h).$$

The defect correctors $\bar{W}'_{h,LDC}$ and $\bar{W}'_{h,DC}$ are similar to $\Pi_h W - w_h$ and, according to (19) should be multiplied by $4/3$ and enriched by an approximation of the interpolation error.

$$\begin{aligned} W'_{h,LDC} &= \frac{4}{3}\bar{W}'_{h,LDC} - (\pi_h W_h - W_h) \\ W'_{h,DC} &= \frac{4}{3}\bar{W}'_{h,DC} - (\pi_h W_h - W_h) \end{aligned} \tag{20}$$

where the $\pi_h W_h - W_h$ is the previously introduced recovery-based approximate interpolation error.

Fin de la correction

Partie morte: Passer aux essais numériques

CORRECTOR IN A LINEAR SETTING. In order to apply the previous analysis to the set of Euler equations, we use the finite-volume/Galerkin equivalence.[48] We do not mention boundary conditions that are expressed by boundary integrals ("BI"). The numerical and continuous solutions of the Euler equations are given by the following equations:

$$\iint \phi \,\mathrm{div}\mathcal{F}(W) \;+\; BI = \;0 \text{ and } \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h) \;+\; BI_h = \;0,$$

where the discrete fluxes are simply:

$$\mathcal{F}_h(.) = \Pi_h \mathcal{F}(\Pi_h .).$$

To build the *a posteriori* estimate, we combine previous equations, the added last term is null:

$$\iint \phi \,\mathrm{div}(\mathcal{F}(W) - \mathcal{F}(W_h)) \;+\; BI \;=\; -\iint \phi \,\mathrm{div}\mathcal{F}(W_h) + \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h) \;+\; BI$$

By using a Taylor expansion of $\mathcal{F}$, we get:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}}{\partial W}\delta w \;=\; -\iint \phi \,\mathrm{div}\mathcal{F}(W_h) + \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h),$$

where δw is the implicit error $\delta w = \Pi_h W - W_h$ By using the Green formula, we obtain:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}}{\partial W}\delta w \;=\; +\iint \mathcal{F}(W_h)\nabla\phi - \iint \mathcal{F}_h(W_h)\nabla\phi_h. \tag{21}$$

In practice we take $\phi_h = 0$, then we discretise. Note that BI terms remain. In contrast to the elliptic case, the RHS will not be zero when ϕ is replaced by ϕ_h. To use this estimate, we assume we can recover from W_h a smooth solution $R_h(W_h)$ enjoying the following properties

1. $R_h(W_h)$ is smooth,
2. $R_h(W_h)$ interpolates W_h in the sense that $\Pi_h R_h(W_h) = W_h$.

We can build a quadratic representation of $R_h(W_h)$ from W_h. The proposed quadratic scheme uses $\mathbf{P}^2$ Lagrange test functions in triangle K to reconstruct a quadratic representation of the solution on K. This interpolation requires the solution nodal value at triangle vertices P_0, P_1 and P_2 , and the solution at the triangle mid-edges. We denote by P_3, P_4 and P_5 the middle of edges $\vec{e}_2$, $\vec{e}_0$ and $\vec{e}_1$, respectively. We denote by $(W_i)_{i=0,2}$ the nodal value of W_h and ∇W_i the nodal gradient of W_h. The quadratic scheme is given by:

$$R_h(W_h)(P) = \sum_{i=0}^{2} \psi_i(P)\, W_h(P_i) + \sum_{i=3}^{5} \psi_i(P)\, R_h(W_h)(P_i),$$

with:

$$\begin{cases} \psi_i(P) = \beta_i(P)\,(2\,\beta_i(P) - 1) & \text{for } i = 0, ..., 2\,, \\ \psi_i(P) = 4\,\beta_{[i]}(P)\,\beta_{[i+1]}(P) & \text{for } i = 3, ..., 5. \end{cases}$$

The mid-node values $(R_h(W_h)(P_i))_{i=3,5}$ are recovered from the nodal gradients of vertices $P_0,\ P_1,\ P_2$:

$$\begin{aligned} R_h(W_h)(P_3) &= \tfrac{1}{2}(W_1 + W_2) + \tfrac{1}{4}(\nabla W_1 + \nabla W_2)\cdot P_1P_2, \\ R_h(W_h)(P_4) &= \tfrac{1}{2}(W_0 + W_2) + \tfrac{1}{4}(\nabla W_2 + \nabla W_0)\cdot P_2P_0, \\ R_h(W_h)(P_5) &= \tfrac{1}{2}(W_0 + W_1) + \tfrac{1}{4}(\nabla W_0 + \nabla W_1)\cdot P_0P_1. \end{aligned}$$

With this scheme, we have $R_h(W_h)(P_i) = W_i$ for the nodal values, and R_h is quadratic, such that properties 1. and 2. hold. Then the last term of (40) is still zero but can be used for building an kind of interpolation error:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}}{\partial W}\delta w \;\approx\; +\iint \nabla\phi_h \,(\mathcal{F}(W_h) - \mathcal{F}_h(W_h)). \tag{22}$$

To get the *a priori* estimate, we combine the two states equations, where the added last term is null:

$$\int\int \phi_h \,\mathrm{div}(\mathcal{F}_h(W_h) - \mathcal{F}_h(W)) \;+\; BI_h = \;-\int\int \phi_h \mathrm{div}\mathcal{F}_h(W) + \int\int \phi_h \mathrm{div}\mathcal{F}(W).$$

In a similar way, by using a Taylor expansion and Green formula, we have:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}_h}{\partial W}(-\delta w) \;=\; + \iint \mathcal{F}_h(W)\nabla\phi_h - \iint \mathcal{F}(W)\nabla\phi_h. \tag{23}$$

To derive the corrector equation, we just have to replace W by W_h:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}_h}{\partial W}(-\delta w) \;=\; + \iint \nabla\phi_h \left(\mathcal{F}_h(W_h) - \mathcal{F}(W)\right). \tag{24}$$

We observe that (41) and (43) are close to each other.

Previous correctors are straightforward to implement when an adjoint solver is already integrated in the flow solver. However, they have several weaknesses:

- There is no guarantee that the corrected solution $W_h + \delta w$ will be physical,
- the approximation of the discrete fluxes $\mathcal{F}_h$ can be far the real numerical fluxes $\mathbf{\Phi_{ij}}$.

In addition, it is also very complex to apply the numerical fluxes $\mathbf{\Phi_{ij}}$ to the recovered solution $R_h(W_h)$. Indeed, as the nodal values are the same, it is necessary to take into account the gradient variation (during the extrapolation) in the different flux choices (fourth or sixth order dissipation). Another simpler strategy is used non linear correctors.

CORRECTOR IN A NON LINEAR SETTING. Instead of solving a linear equation to find the corrector, we build an error equation that consists in adding a source term to reduce a defect. This residual is close to the second member found in Equations (41) and (43). In the *a priori* setting, the defect is the residual of numerical flux applied to the continuous solution. Indeed, optimally, we want

$$\mathrm{div}(\mathcal{F}_h(W)) = 0, \quad \textit{(a priori)} \tag{25}$$

when W is the exact solution and $\mathcal{F}_h$ represents the numerical scheme. Consequently, the corrected solution W_c of the current discrete solution W_h is found by solving the non linear equation:

$$\mathrm{div}(\mathcal{F}_h(W_c)) = -S_h(W),$$

where $S_h(W)$ is the source term computed from the divergence of $\mathcal{F}_h(W)$. In our approach W is replaced by the smooth recovery $R_h(W_h)$ and we use locally a finer mesh to take into account the nodal values at mid-edges. $S_h(W)$ is then approximated as:

$$S_h(W) \approx \mathrm{div}(\mathcal{F}_{h/2}(R_h(W_h))).$$

A transfer procedure is used to accumulate $\mathcal{F}_{h/2}(R_h(W_h))$ on the coarser mesh h. In contrast with the linear approach, the only modification in the flow solver is to take into account a residual source term. The corrected solution is found (instead of the implicit error) and the corrected solution is guarantee to be physical.

In the *a posteriori* setting, the defect is the gap between the continuous PDE with respect to the numerical solution. Indeed, ideally, W_h should verify

$$\mathrm{div}(\mathcal{F}(W_h)) = 0, \quad \textit{(a posteriori)} \tag{26}$$

meaning that the numerical solution solves exactly the continuous set of Euler equations. In this case, the (exact) source term is the divergence of $\mathcal{F}(W_h)$. The continuous fluxes $\mathcal{F}$ are approximated on locally refined grids, so that the source term becomes the divergence of $\mathcal{F}_{h/2}(W_h)$. The solution on the finer mesh is interpolation from the coarser mesh and the source term computed on $h/2$ is accumulated back from the finer to the coarser mesh.

From a practical point of view, the finer grids are never generated as we can solve local problem to compute the source terms. The flow solver is then used to inverse the error equation directly. The procedure to derive the corrected solution is then:

Step 1.: Solve the flow problem to get the numerical solution W_h

Step 2.: Compute the source term on finer mesh $-S_h(W)$ or $S(W_h)$.

Step 3.: From W_h, converge again the solution with the residual source term added.

To perform mesh adaptation, the corrector, *i.e.* the RHS of the adjoint equation is computed as $W_c - W_h$.

IV. Formal error analysis within the continuous mesh framework

The norm oriented approach is based on previous developments on anisotropic (Hessian-based) and goal-oriented mesh adaptation. In these two cases, the anisotropic mesh prescription (orientations and sizes) is given in a close form. Each of them are tightly related to interpolation error: on the solution field for Hessian-based methods, and the Euler fluxes for the goal-oriented approach. We first recall formally the derivation of these estimates in the continuous mesh framework. These leads to the definition of two *kernels* (interpolation and goal oriented) providing the optimal mesh. The norm oriented approach is just a combination of these kernels along with the derivation of a solution corrector. In this section, we focus on controlling the implicit error $\Pi_h u - u_h$. Controlling the approximation error will consists in controlling the implicit error (corrector) and interpolation error terms simultaneously as :

$$u - u_h = u - \Pi_h u + \Pi_h u - u_h.$$

Note that the implicit error can be seen as a point-wise error between the exact solution and the numerical one. Knowing the numerical scheme and PDE at hand is then mandatory. In this section, we consider a linear (formal) PDE.

IV.A. Continuous mesh framework formalism

We use in the sequel the continuous mesh framework to drive our analysis. In,[37] we prove that any mesh can be represented by a continuous Riemannian metric field $\mathcal{M}$. The link between continuous mesh and discrete mesh is based on the unit mesh concept.[25] Given a Riemannian metric field $\mathcal{M}$, a unit-mesh is a mesh having length computed in $\mathcal{M}$ and volume computed in $\mathcal{M}$ close to one:

$$\begin{aligned} &\text{for all edges of } \mathbf{e} = \mathbf{AB}, \qquad \ell_{\mathcal{M}}(\mathbf{e}) = \int_0^1 \sqrt{{}^t\mathbf{AB}\mathcal{M}((1-t)A+tB)\mathbf{AB}}\mathrm{d}t \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right], \\ &\text{for all elements } K, |K|_{\mathcal{M}} \approx \frac{\sqrt{2}}{12}. \end{aligned} \tag{27}$$

From a practical point of view, generating a anisotropic mesh with respect to $\mathcal{M}$ requires to use any anisotropic mesh generators, see[11,15,16,21,26,33,35,41,49,54] in 3D. Conversely, given a mesh $\mathcal{H}$, the following metric field is a continuous representative of $\mathcal{H}$:

$$\mathcal{M}_P = \exp\left(\frac{\sum_{P\in K}|K|\ln(\mathcal{M}_K)}{\sum_{P\in K}|K|}\right),$$

where P is a vertex of $\mathcal{H}$ and $\mathcal{M}_K$ is the unique metric of element K, and $|K|$ the volume of K. Consequently, if u_h denotes a discrete quantity computed on a given mesh, we use equivalently the notation $u_{\mathcal{M}}$, that represents the same quantities represented on any unit mesh with respect to $\mathcal{M}$. In the case of the interpolation error, there is a strict equivalence between continuous and discrete interpolation error.[38] The parametrization of a mesh by $\mathcal{M}$ instead of h is advantageous for *a priori* analysis with anisotropic mesh. Indeed, there exist also quantities of interest as the density, anisotropic ratios, differentiation that are well defined on $\mathcal{M}$.

IV.B. Interpolation error

Let us consider a mesh parametrised by a metric $\mathcal{M}$, according to the continuous mesh theory. The P^1 interpolation error $\Pi_h u - u$ can be expressed in terms of second derivatives of u and of metric $\mathcal{M}$:

$$\Pi_h u - u \approx \text{trace}(\mathcal{M}^{-\frac{1}{2}} H(u) \mathcal{M}^{-\frac{1}{2}}), \tag{28}$$

where $H(u)$ is the Hessian of u. Minimizing $\|\Pi_h u - u\|_{\mathbf{L}^1}$ for on node provides the optimal interpolation-based metric:

$$\mathcal{M}^{opt}_{inter}(u) = \det(|H(u)|)^{\frac{1}{3}} |H(u)|,$$

where $|H|$ is derived from H by taking the absolute values of the eigen-values. We denote by $\mathcal{K}_1(u)$, the kernel giving the optimal metric for the interpolation error on u in $\mathbf{L}^1$ norm. Note that replacing the optimal metric in the L^2 norm shows that second-order convergence is obtained for smooth contexts.[38] We can also extend this approach to non-smooth functions, see.[20]

IV.C. Implicit error analysis

We are interested in computing the function u solution of the PDE:

$$Au = f$$

In the PDE discretisation process, we choose a mesh parametrized by h (or $\mathcal{M}$ more generally) and replace the computation of function u by the computation of an array of real numbers $\mathbf{u}_h$, called the degrees of freedom of the discrete solution:

$$\mathbf{u}_h \in \mathbb{R}^N \ ; \ A_h \mathbf{u}_h = \mathbf{f}_h. \tag{29}$$

Before checking how our discrete solution is close to the continuous one, it is useful to build, by interpolation from the degrees of freedom solutions of (29), an approximate function u_h which should be close to the exact one:

$$u_h = R_h \mathbf{u}_h.$$

Or to define an array $T_h u$ to be compared with the degrees of freedom $\mathbf{u}_h$, for example by choosing $T_h = R_h^*$. Two classes of error analysis exist: either *a priori* or *a posteriori*.

In *a posteriori* analysis, we estimate $u - u_h$ in some functional norm by setting:

$$A(u - u_h) = f - Au_h \ \ \Rightarrow \ \ u - u_h = A^{-1}(f - Au_h),$$

where $f - Au_h$ is the continuous residual. It is evident that the difficulty in computing $A^{-1}(f - Au_h)$ is as large as for computing the initial problem $A^{-1}f$. In general, we shall try to replace A^{-1} by its discretisation A_h^{-1}, loosing the exactness of the estimate.

In *a priori* analysis, we write:

$$A_h(T_h u - \mathbf{u}_h) = A_h T_h u - \mathbf{f}_h \ \ \Rightarrow \ \ T_h u - \mathbf{u}_h = A_h^{-1}(A_h T_h u - \mathbf{f}_h).$$

This is computable, knowing u, but the relation between $T_h u - \mathbf{u}_h$ and the error under study $u - u_h$ is less evident. In the case of a finite-element type variational method, we shall be able to state the above *a priori* estimate in terms of functions. Let $V = H^1(\Omega)$ and $\mathcal{V} = V \cap \mathcal{C}^0(\bar{\Omega})$. Let Π_h be the P^1 interpolation:

$$\begin{aligned} \Pi_h : \mathcal{V} &\rightarrow \Pi_h(V), \\ \Pi_h v(X_i) &= v(X_i) \ \forall \ i, \ \text{vertex}, \ \Pi_h v | T \ \text{is affine} \ \forall T, \ element. \end{aligned}$$

We note that $V_h = \Pi_h(V) = \Pi_h(\mathcal{V}) \subset V$. Let us define:

$$\begin{aligned} A_h : V &\rightarrow V' \\ (A_h v, w) &= (Av, \Pi_h w) \end{aligned}$$

Where $(,)$ holds for $(,)_{V \times V'}$. Then the discretized system writes:

$$\begin{aligned} u_h \in V_h \ \text{and} \ \forall v_h \in V_h, \\ (A_h u_h, v_h) = (f, v_h) \end{aligned}$$

in which the following restriction of A_h is invertible:

$$\bar{A}_h : V_h \tilde{\rightarrow} (V_h)' \ \ ; \ \ (\bar{A}_h)^{-1} : (V_h)' \tilde{\rightarrow} V_h$$

Observing that:

$$(A_h u, v_h) = (f, v_h),$$

we derive:

$$(A_h(\Pi_h u - u_h), v_h) = (A_h(\Pi_h u - u), v_h) + (A_h(u - u_h), v_h) = (A_h(\Pi_h u - u), v_h).$$

We observe that $A_h(\Pi_h u - u) \in (V_h)'$ and therefore:

$$\Pi_h u - u_h = (\bar{A}_h)^{-1}(A_h(\Pi_h u - u)) \tag{30}$$

In the case of the elliptic model, we have:

$$a(\Pi_h u - u_h, \varphi_h) = a(\Pi_h u - u, \varphi_h)$$

where

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx.$$

Let us assume that we have a good approximation of the derivatives of u by a recovery procedure from u_h. We can derive an estimate for the gradient of $\Pi_h u - u$, see:[23]

$$\nabla(\Pi_h u - u) \approx \begin{pmatrix} \left(\frac{\partial u}{\partial x}\right)^2 & \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} & \frac{\partial u}{\partial x}\frac{\partial u}{\partial z} \\ \frac{\partial u}{\partial x}\frac{\partial u}{\partial y} & \left(\frac{\partial u}{\partial x}\right)^2 & \frac{\partial u}{\partial y}\frac{\partial u}{\partial z} \\ \frac{\partial u}{\partial x}\frac{\partial u}{\partial z} & \frac{\partial u}{\partial y}\frac{\partial u}{\partial z} & \left(\frac{\partial u}{\partial z}\right)^2 \end{pmatrix} \approx G(h, u_h),$$

where G retains only derivatives of u, recovered from u_h. This can help building a right hand side to (30) and get an error approximate model.

$$\begin{aligned} \Pi_h u - u_h &\approx u' \quad \text{such that} \\ a(u', \varphi_h) &= \int_\Omega G(h, u_h).\nabla\varphi_h dx. \end{aligned} \tag{31}$$

Estimates (30) and (31) are the *a priori* counterparts of the *a posteriori* estimates of Becker-Rannacher[9] used in combination with an adjoint in Giles-Pierce.[28]

IV.D. Scalar output goal-oriented analysis

The goal-oriented analysis relies on the minimization of the error committed on a scalar output functional j which we simplify as follows:

$$j(u) - j(u_h) = (\frac{\partial j}{\partial u}(u), \Pi_h u - u_h) = (g, \Pi_h u - u_h), \tag{32}$$

Then, using (30), we get:

$$j(u) - j(u_h) \approx (A_h^* \bar{A}_h^{-*} g, \Pi_h u - u). \tag{33}$$

The above right-hand side transforms in:

$$(A_h^* \bar{A}_h^{-*} g, \Pi_h u - u) \approx (1, \text{trace}(\mathcal{M}^{-1/2} \, A_h^* \bar{A}_h^{-*} g |H(u)| \, \mathcal{M}^{-1/2})),$$

where $\mathcal{M}$ is a metric field representing the current mesh according to (27) and (28). To minimize $(A_h^* \bar{A}_h^{-*} g, \Pi_h u - u)$, we can consider its stationarity condition, obtained by differentiating it with respect to $\mathcal{M}$. We assume that mesh is fine enough, and the error term $h^{-1/2}$ is an infinitely small, typically h^α. Then its derivative with respect to mesh is a $O(h^{\alpha-1})$, much larger than the variation of $A_h^* \bar{A}_h^{-*} g$. We then neglect the latter variation, the optimal metric (mesh) is then found by evaluating everything on the current (initial) mesh parametrized by h_0:

$$\mathcal{M}_{h_0}^{opt} = \text{argmin}_{\mathcal{M}} \text{trace}(\mathcal{M}^{-1/2} \, A_{h_0}^* \bar{A}_{h_0}^{-*} g |H(u)| \, \mathcal{M}^{-1/2})$$

$$\mathcal{M}_{h_0}^{opt} = \mathcal{K}_{goal}(A_{h_0}^* \bar{A}_{h_0}^{-*} g, u)$$

from which by converging a fixed point we get the solution where adjoints states and Hessian are recovered from the optimal mesh parametrized by h_{opt}:

$$\mathcal{M}_{goal}^{opt} = \mathcal{K}_{goal}(A_{h_{optl}}^* \bar{A}_{h_{opt}}^{-*} g, u).$$

This defines the second kernel.

IV.E. Norm-oriented analysis

metric-based goal-oriented metric is used. The norm of the implicit error is then controlled.

We are interested by the minimization of the following expression with respect to the mesh h:

$$j(h) = ||\Pi_h u - u_h||^2_{L^2(\Omega)}. \tag{34}$$

The stationarity condition writes:

$$j'(h).\delta h = (\Pi_h u - u_h, \frac{\partial}{\partial h}(\Pi_h u - u_h).\delta h). \tag{35}$$

Now:

$$\frac{\partial}{\partial h}(\Pi_h u - u_h).\delta h = \frac{\partial}{\partial h}(\bar{A}_h^{-1} A_h).\delta h\ (\Pi_h u - u) + \bar{A}_h^{-1} A_h\ \frac{\partial}{\partial h}(\Pi_h u - u).\delta h \tag{36}$$

Then a standard metric-based goal-oriented metric is used. The norm of the implicit error is then controlled.

We have to use again the fact that we consider that the error is an infinitely small, typically h^α. Then its derivative with respect to the mesh is a $O(h^{\alpha-1})$, much larger than the other terms. We write:

$$\frac{\partial}{\partial h}(\Pi_h u - u_h).\delta h \approx \bar{A}_h^{-1} A_h\ \frac{\partial}{\partial h}(\Pi_h u - u).\delta h. \tag{37}$$

Replacing $\Pi_h u - u_h$ by $A_h^{-1} A_h(\Pi_h u - u)$, we get

$$\mathcal{M}^{opt}_{norm} = \mathcal{K}_{goal}(A_h^* \bar{A}_h^{-*} \bar{A}_h^{-1} A_h(\Pi_h u - u), u).$$

As for as the goal oriented approach, the optimal mesh is obtained by iterating the procedure. We consider a counter α representing the current iteration and the current mesh, the algorithm is then as follows:

Step 1. : Solve the linearised error system:

$$\bar{A}_{h^{(\alpha)}} u' = A_{h^{(\alpha)}}(\Pi_{h^{(\alpha)}} u - u), \tag{38}$$

where $\Pi_h u - u$ is express ed in terms of metric and Hessian, as in (31).

Step 2. : Solve the adjoint system, where the second member is the corrector obtained in **Step 1.**:

$$\bar{A}^*_{h^{(\alpha)}} u^* = u',$$

Step 3. : Compute the goal-oriented metric to derive the optimal mesh at step α:

$$\mathcal{M}^{(\alpha+1)} = \mathcal{K}_{goal}(A^*_{h^{(\alpha)}} u^*, u),$$

We then iterate **Steps 1-3** until we get a fixed point $\mathcal{M}^{opt}_{norm} = \mathcal{M}^{(\infty)}$. It is then interesting to evaluate the optimal functional, obtained by replacing $\mathcal{M}^{opt}_{norm}$ into the norm-based functional j. Second-order accuracy is formally predicted. Of course this does not rigorously apply to any choice of the initial norm in j. From a practical of view, the additional cost from goal-oriented or multi-scale approaches is to solve an additional linear system (which is the transpose of the adjoint one) with the computation of its second member (and estimate of the interpolation error).

REMARK. Note that it is possible to modify the right of side of **Step 1.** by considering a different error estimate. Indeed, one disadvantage of the *a priori* analysis is that it neglects high-order terms in the local error. A way to avoid this disadvantage is to replace in the square norm one of the error u' by an *a posteriori* one u'_{post}, e.g. the solution of:

$$(\bar{A}_h u'_{post}, v_h) = (Au_h - f, v_h)$$

for any v_h in V_h. We consider now how the RHS of this system

$$(R, \varphi) = (Au_h - f, \varphi)$$

can be computed in the case of the continuous P_1-by-element finite element approximation. It writes:

$$(Au_h - f, v_h) = (-f, v_h) + \sum_{ij} \int_{ij} [\nabla u_h \cdot \mathbf{n}] meas(ij)\ \mathrm{d}\sigma$$

where the sum $\sum_{ij}$ is computed for all the edges ij in 2D (all the faces ijk in 3D) of the mesh and where the bracket $[]$ hold for the jump of the quantity inside through the edge ij (resp. face ijk). Therefore this RHS can be introduced directly in the error equation in order to get u'_{post} and continue the norm-oriented mesh optimisation

V. Numerical flow solver

In this section, we recall the man features of the flow solver. We restrict ourselves to the case of inviscid flows. More details for solving RANS equations can be found in.[46]

V.A. Modeling equations

The compressible Euler equations for mass, momentum and energy conservation reads (with no source terms):

$$\left\{\begin{array}{rcl} \dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) &=& 0\,, \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p &=& 0, \\ \dfrac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p)\mathbf{u}) &=& 0, \end{array}\right.$$

where ρ denotes the density, $\mathbf{u}$ the velocity, e the total energy per mass and p the pressure. This system can be rewritten under vectorial form:

$$W_t + F_1(W)_x + F_2(W)_y + F_3(W)_z = 0,$$

where W is the nondimensionalized conservative variables vector:

$$W = (\rho, \rho u, \rho v, \rho w, \rho E)^T$$

$\mathbf{F}(W) = (F_1(W), F_2(W), F_3(W))$ are the convective (Euler) flux functions:

$$\begin{array}{rcl} F_1(W) &=& (\rho u,\ \rho u^2 + p,\ \rho uv,\ \rho uw,\ u(\rho E + p))^T \\ F_2(W) &=& (\rho v,\ \rho uv,\ \rho v^2 + p,\ \rho vw,\ v(\rho E + p))^T \\ F_3(W) &=& (\rho w,\ \rho uw,\ \rho vw,\ \rho w^2 + p,\ w(\rho E + p))^T. \end{array}$$

V.B. Spatial discretization

The spatial discretization of the fluid equations is based on a vertex-centered finite element/finite volume formulation on unstructured meshes. It combines a HLLC upwind schemes for computing the convective fluxes and the Galerkin centered method for evaluating the viscous terms. Second order space accuracy is achieved through a piecewise linear interpolation based on the Monotonic Upwind Scheme for Conservation Law (MUSCL) procedure which uses a particular edge-based formulation with upwind elements. A specific slope limiter is employed to damp or eliminate spurious oscillations that may occur in the vicinity of discontinuities.

V.B.1. Finite Volume discretization

Let $\mathcal{H}$ a mesh of domain Ω, the vertex-centered finite volume formulation consists in associating with each vertex P_i of the mesh a control volume or finite volume cell, denoted C_i. Discretized domain Ω_h can be written as the union of the elements or the union of the finite volume cells:

$$\Omega_h = \bigcup_{i=1}^{N_T} K_i = \bigcup_{i=1}^{N_S} C_i\,.$$

In order to discretize accurately the flow equations on highly anisotropic meshes, the dual finite volume cells are the containment sphere cells introduced in 2D by Barth[6] and generalized to 3D by Dervieux[30] instead of the classic median cells. It consists in subdividing each tetrahedron into four hexahedra cell around each vertex. The hexahedron cell vertices associated with vertex P_i are (i) the middle of the three edges issued from P_i, (ii) the containment circle center of the three faces containing P_i, (iii) the containment sphere center of the tetrahedron and (iv) the considered vertex P_i. The containment sphere cells of vertex P_i is the union of all its hexahedra cells. The containment sphere center correspond the the sphere circumcenter if it falls inside the element.

The set of equations are integrated on the finite volume cell C_i following a finite volume formulation reads (using the Green formula):

$$|C_i| \frac{dW_i}{dt} + \int_{\partial C_i} \mathbf{F}(W_i) \cdot \mathbf{n}_i \, \mathrm{d}\gamma = \mathrm{B.T.}, \tag{39}$$

where W_i is the mean value of the solution W on the cell C_i, $\mathbf{n}_i$ is the outer normal to the finite volume cell surface ∂C_i, $\mathbf{F}$, are the convective terms and B.T. represents boundary conditions terms.

V.B.2. Discretization of the convective terms

The integration of convective fluxes $\mathbf{F}$ of Equation (39) is done by decomposing the cell boundary in many facets ∂C_{ij}:

$$\int_{\partial C_i} \mathbf{F}(W_i) \cdot \mathbf{n}_i \, \mathrm{d}\gamma \; = \sum_{P_j \in \mathcal{V}(P_i)} \mathbf{F}|_{\partial C_{ij}} \cdot \int_{\partial C_{ij}} \mathbf{n}_i \, \mathrm{d}\gamma \, ,$$

where $\mathcal{V}(P_i)$ is the set of all neighboring vertices linked by an edge to P_i and $\mathbf{F}|_{\partial C_{ij}}$ represents the constant value of $\mathbf{F}(W)$ at interface ∂C_{ij}. The flow is calculated with a numerical flux function, denoted Φ_{ij}:

$$\Phi_{ij} = \Phi_{ij}(W_i, W_j, \mathbf{n}_{ij}) = \mathbf{F}|_{\partial C_{ij}} \cdot \int_{\partial C_{ij}} \mathbf{n}_i \, \mathrm{d}\gamma \, ,$$

where $\mathbf{n}_{ij} = \displaystyle\int_{\partial C_{ij}} \mathbf{n}_i \, \mathrm{d}\gamma$. The numerical flux function approximates the hyperbolic terms on the common boundary ∂C_{ij}. We notice that the computation of the convective fluxes is performed mono-dimensionally in the direction normal to the boundary of the finite volume cell. Therefore, the numerical calculation of the flux function Φ_{ij} at the interface ∂C_{ij} is achieved by the resolution of a one-dimensional Riemann problem in the direction of the normal $\mathbf{n}_{ij}$ by means of an approximate Riemann solver. In this work, the HLLC approximate Riemann solver is used for the mean flow - more details can be found[7] - and linear upwind advection is used for the turbulent variable convection.

HLLC APPROXIMATE RIEMANN SOLVER. The idea of the HLLC flow solver is to consider locally a simplified Riemann problem with two intermediate states depending on the local left and right states. The simplified solution to the Riemann problem consists of a contact wave with a velocity S_M and two acoustic waves, which may be either shocks or expansion fans. The acoustic waves have the smallest and the largest velocities (S_L and S_R, respectively) of all the waves present in the exact solution. If $S_L > 0$ then the flow is supersonic from left to right and the upwind flux is simply defined from $\mathbf{F}(W_l)$ where W_l is the state to the left of the discontinuity. Similarly, if $S_R < 0$ then the flow is supersonic from right to left and the flux is defined from $\mathbf{F}(W_r)$ where W_r is the state to the right of the discontinuity. In the more difficult subsonic case when $S_L < 0 < S_R$ we have to calculate $\mathbf{F}(W_l^*)$ or $\mathbf{F}(W_r^*)$. Consequently, the HLLC flux is given by:

$$\Phi_{lr}^{hllc}(W_l, W_r, \mathbf{n}_{lr}) = \begin{cases} \mathbf{F}(W_l) \cdot \mathbf{n}_{lr} & \text{if } S_L > 0 \\ \mathbf{F}(W_l^*) \cdot \mathbf{n}_{lr} & \text{if } S_L \leq 0 < S_M \\ \mathbf{F}(W_r^*) \cdot \mathbf{n}_{lr} & \text{if } S_M \leq 0 \leq S_R \\ \mathbf{F}(W_r) \cdot \mathbf{n}_{lr} & \text{if } S_R < 0 \end{cases}$$

Now, let us specify how W_l^* and W_r^* are evaluated. We denote by $\eta = \mathbf{u} \cdot \mathbf{n}$. Assuming that $\eta^* = \eta_l^* = \eta_r^* = S_M$, the following evaluations are proposed[7] (the subscript $_l$ or $_r$ are omitted for clarity):

$$W^* = \frac{1}{S - S_M} \begin{pmatrix} \rho \, (S - \eta) \\ \rho \mathbf{u} \, (S - \eta) + (p^* - p)\mathbf{n} \\ \rho E \, (S - \eta) + p^* S_M - p\eta \end{pmatrix} \quad \text{where} \quad p^* = \rho \, (S - \eta)(S_M - \eta) + p \, .$$

A key feature of this solver is in the definition of the three waves velocity. For the contact wave we consider:

$$S_M = \frac{\rho_r \eta_r (S_R - \eta_r) - \rho_l \eta_l (S_L - \eta_l) + p_l - p_r}{\rho_r (S_R - \eta_r) - \rho_l (S_L - \eta_l)} \, ,$$

and the acoustic wave speeds based on Roe average:

$$S_L = \min(\eta_l - c_l, \tilde{\eta} - \tilde{c}) \quad \text{and} \quad S_R = \max(\eta_r + c_r, \tilde{\eta} + \tilde{c}) \, .$$

With such waves velocities, the HLLC Riemann solver has the following properties. It automatically (i) satisfies the entropy inequality, (ii) resolves isolated contacts exactly, (iii) resolves isolated shocks exactly, (iv) preserves positivity.

HIGH-ORDER ACCURATE VERSION. The MUSCL type reconstruction method has been designed to increase the order of accuracy of the scheme.[34] The idea is to use extrapolated values W_{ij} and W_{ji} of W at the interface ∂C_{ij} to evaluate the flux. The following approximation is performed:

$$\Phi_{ij} = \Phi_{ij}(W_{ij}, W_{ji}, \mathbf{n}_{ij}) \,,$$

with W_{ij} and W_{ji} which are linearly interpolated as:

$$W_{ij} \;=\; W_i + \frac{1}{2}\,(\nabla W)_{ij} \cdot \overrightarrow{P_i P_j} \qquad \text{and} \qquad W_{ji} \;=\; W_j + \frac{1}{2}\,(\nabla W)_{ji} \cdot \overrightarrow{P_j P_i} \,,$$

where, in contrast to the original MUSCL approach, the approximate "slopes" $(\nabla W)_{ij}$ and $(\nabla W)_{ji}$ are defined for any edge and obtained using a combination of centered, upwind and nodal gradients.

The centered gradient, which is related to edge P_iP_j, is defined as:

$$(\nabla W)^C_{ij} \cdot \overrightarrow{P_i P_j} = W_j - W_i \,.$$

Upwind and downwind gradients, which are also related to edge P_iP_j, are computed according to the definition of upstream and downstream tetrahedra of edge P_iP_j. These tetrahedra are respectively denoted K_{ij} and K_{ji}. K_{ij} (resp. K_{ji}) is the unique tetrahedron of the ball of P_i (resp. P_j) the opposite face of which is crossed by the line defined by the edge P_iP_j. Upwind and downwind gradients are then defined for vertices P_i and P_j as:

$$(\nabla W)^U_{ij} = (\nabla W)|_{K_{ij}} \quad \text{and} \quad (\nabla W)^D_{ij} = (\nabla W)|_{K_{ji}} \,.$$

where $(\nabla W)|_K = \sum_{P \in K} W_P \nabla \phi_P|_K$ is the P_1-Galerkin gradient on tetrahedron K. Parametrized nodal gradients are built by introducing the β-scheme:

$$\begin{aligned} (\nabla W)_{ij} &= (1-\beta)(\nabla W)^C_{ij} + \beta\,(\nabla W)^U_{ij} \\ (\nabla W)_{ji} &= (1-\beta)(\nabla W)^C_{ij} + \beta\,(\nabla W)^D_{ij} \,, \end{aligned}$$

where $\beta \in [0,1]$ is a parameter controlling the amount of upwinding. For instance, the scheme is centered for $\beta = 0$ and fully upwind for $\beta = 1$.

NUMERICAL DISSIPATION OF FOURTH-ORDER: V4-SCHEME. The most accurate β-scheme is obtained for $\beta = 1/3$. Indeed, it can be demonstrated that this scheme is third-order for the two-dimensional linear advection on structured triangular meshes. On unstructured meshes, a second-order scheme with a fourth-order numerical dissipation is obtained. These high-order gradients are given by:

$$\begin{aligned} (\nabla W)^{V4}_{ij} &= \frac{2}{3}\,(\nabla W)^C_{ij} + \frac{1}{3}\,(\nabla W)^U_{ij} \\ (\nabla W)^{V4}_{ji} &= \frac{2}{3}\,(\nabla W)^C_{ji} + \frac{1}{3}\,(\nabla W)^D_{ij} \,. \end{aligned}$$

NUMERICAL DISSIPATION OF SIXTH-ORDER: V6-SCHEME. An even less dissipative scheme has been proposed.[19] It is a more complex linear combination of gradients using centered, upwind and nodal P_1-Galerkin gradients. The nodal P_1-Galerkin gradient of P_i is related to cell C_i and is computed by averaging the gradients of all the tetrahedra containing vertex P_i:

$$(\nabla W)_{P_i} = \frac{1}{4\,|C_i|} \sum_{K \in C_i} |K| (\nabla W)|_K \,.$$

A sixth-order dissipation scheme is then obtained by considering the following high-order gradient:

$$\begin{aligned} (\nabla W)^{V6}_{ij} &= (\nabla W)^{V4}_{ij} - \frac{1}{30}\left((\nabla W)^U_{ij} - 2\,(\nabla W)^C_{ij} + (\nabla W)^D_{ij}\right) - \frac{2}{15}\left((\nabla W)_{M_i} - 2\,(\nabla W)_{P_i} + (\nabla W)_{P_j}\right) \\ (\nabla W)^{V6}_{ji} &= (\nabla W)^{V4}_{ji} - \frac{1}{30}\left((\nabla W)^D_{ij} - 2\,(\nabla W)^C_{ij} + (\nabla W)^U_{ij}\right) - \frac{2}{15}\left((\nabla W)_{M_j} - 2\,(\nabla W)_{P_j} + (\nabla W)_{P_i}\right) \,, \end{aligned}$$

where $(\nabla W)_{M_{i,j}}$ is the gradient at the points $M_{i,j}$ intersection of the line defined by P_iP_j and upwind-downwind tetrahedra. These gradients are computed by linear interpolation of the nodal gradients of faces containing $M_{i,j}$.

DERVIEUX LIMITER. The previous MUSCL schemes are not monotone. Therefore, limiting functions must be coupled with the previous high-order gradient evaluations to guarantee the TVD property of the scheme. The gradient is substituted by a limited gradient denoted $(\nabla W)_{ij}^{lim}$. Here, we consider the three-entries limiter introduced by Dervieux which is a generalization of the Superbee limiter:[18]

$$\begin{aligned}&\text{if } uv \leq 0 \text{ then}\\ &\quad Lim(u,v,w) = 0\\ &\text{else}\\ &\quad Lim(u,v,w) = Sign(u)\, min(2\,|u|, 2\,|v|, |w|)\,,\end{aligned}$$

and we use: $Lim((\nabla W)_{ij}^{C}, (\nabla W)_{ij}^{D}, (\nabla W)_{ij}^{HO})$ where $(\nabla W)_{ij}^{HO}$ is even $(\nabla W)_{ij}^{V4}$ or $(\nabla W)_{ij}^{V6}$.

V.C. Boundary conditions

For presented flow simulations, three boundary conditions are required. Slip boundary conditions are imposed for bodies when the flow is considered inviscid or for symmetry. For viscous flow, no slip boundary conditions are considered for bodies. And finally, we used Steger-Warming flux to set up free-stream (external flow) conditions.

SLIP CONDITION For this boundary condition we impose weakly $\mathbf{u}\cdot\mathbf{n} = 0$, which is done by imposing the following boundary flux:

$$\Phi_{slip}(W_i, T) = \sum_{T \ni P_i} \int_{\partial C_i \cap T} \mathbf{F}_{slip}(W_i)\cdot\mathbf{n}_T \,\mathrm{d}\gamma \quad \text{with} \quad \mathbf{F}_{slip}(W_i)\cdot\mathbf{n}_T = (0,\ p_i\,\mathbf{n}_T,\ 0)^t\,.$$

where T are boundary faces with normals $\mathbf{n}_T$.

NO SLIP CONDITION. Adiabatic conditions are considered, therefore only a null velocity is imposed strongly for this boundary condition: $\mathbf{u} = 0$.

FREE-STREAM CONDITION. This condition imposes a free-stream uniform flow from the infinite. It applies when we have a boundary Γ_∞ for which the infinite constant state W_∞ is prescribed:

$$W_\infty = (\rho_\infty,\ (\rho\mathbf{u})_\infty,\ (\rho E)_\infty)^t \quad \text{and} \quad \tilde{\nu}_{farfield} \in [3\nu_\infty\,,\, 5\nu_\infty]\,.$$

This state enables upwind fluxes at the infinite to be computed. The considered boundary fluxes are built from a decomposition following the characteristics values. We consider the Steger-Warming flux which is completely upwind on solution W_i:

$$\Phi_\infty(W_i, T) = A^+(W_i, \mathbf{n_T})W_i + A^-(W_i, \mathbf{n_T})W_\infty \quad \text{where} \quad A^+ = \frac{|A| + A}{2} \ \text{ and } \ A^- = \frac{|A| - A}{2}\,.$$

V.D. Time integration

V.D.1. Implicit scheme

For an explicit time discretization, the semi-discretized system reads:

$$\frac{|C_i|}{\delta t_i^n}\left(W_i^{n+1} - W_i^n\right) \ = \ R_i(W^n)$$

where $R_i(W^n) = -\sum_{j\in\mathcal{V}(i)} \Phi_{hllc}(W_i^n, W_j^n, \mathbf{n}_{ij}) - \sum_{T\ni P_i} \Phi_{bc,T}(W_i^n, \mathbf{n}_T)$. For implicit time discretization, we have :

$$\frac{|C_i|}{\delta t_i^n}\left(W_i^{n+1} - W_i^n\right) \ = \ R_i(W^{n+1})$$

which is linearized as:

$$\left(\frac{|C_i|}{\delta t_i^n} I_d - \frac{\partial R_i}{\partial W}(W^n)\right)\left(W_i^{n+1} - W_i^n\right) \ = \ R_i(W^n)$$

where $\dfrac{\partial R_i}{\partial W}(W^n)$ contributes the i^{th} line of the matrix and the following linearization have been done (with the notation $\delta W = W^{n+1} - W^n$):

$$\begin{aligned}
\Phi_{hllc}(W_i^{n+1}, W_j^{n+1}, \mathbf{n}_{ij}) &= \Phi_{hllc}(W_i^n, W_j^n, \mathbf{n}_{ij}) + \frac{\Phi_{hllc}}{\partial W_i}(W_i^n, W_j^n, \mathbf{n}_{ij})\delta W_i + \frac{\Phi_{hllc}}{\partial W_j}(W_i^n, W_j^n, \mathbf{n}_{ij})\delta W_j \\
\Phi_{bc,T}(W_i^{n+1}, \mathbf{n}_T) &= \Phi_{bc,T}(W_i^n, \mathbf{n}_T) + \frac{\Phi_{bc,T}}{\partial W_i}(W_i^n, \mathbf{n}_T)\delta W_i \,.
\end{aligned}$$

Terms $\dfrac{\partial \Phi}{\partial W_i}$ and $\dfrac{\partial \Phi}{\partial W_i}$ contribute to the matrix diagonal block and extra-diagonal block, respectively, and terms Φ contributes to the right-hand side.

We then rewrite the linearized system in compact form:

$$\mathbf{A}^n \, \delta \mathbf{W}^n = \mathbf{R}^n$$

$$\text{where} \quad \mathbf{A}^n = \frac{|C|}{\delta t^n}\mathbf{I} - \frac{\partial \mathbf{R}^n}{\partial \mathbf{W}} \quad \text{and} \quad \delta \mathbf{W}^n = \mathbf{W}^{n+1} - \mathbf{W}^n \,.$$

V.D.2. Newton's method

To solve the non-linear system, we follow the approach based on Lower-Upper Symmetric Gauss-Seidel (LU-SGS) implicit solver initially introduced by Jameson[32] and fully developed by Sharov et al. and Luo et al.[43,44,52,53] The Newton's method can be either the LU-SGS approximate factorization or the SGS relaxation or the GMRES method with LUSGS or SGS as preconditioner. The LU-SGS and SGS are very attractive because they use an edge-based data structure which can be efficiently parallelized with p-threads.[4,52] From our experience, we have made the following - crucial - choices to solve the compressible Euler equations.

Converging the Newton's method is important for the global convergence of the Euler non-linear problem. Hence, an iterative method is required such as SGS or GMRES+LUSGS or GMRES+SGS[a]. Usually, the Newton's method iterates until the residual of the linear system is reduced by two orders of magnitude: 0.01.

The choice of the renumbering also impacts strongly the convergence of the non-linear system. While Hilbert-type (space filling curve) renumbering is very efficient for cache misses and memory contention,[4,52] Breadth-first search renumbering proves to be more effective for the convergence of the implicit method and the overall efficiency.

Luo et al.[43,44,52] proposed to use a simplified flux function - a Rusanov approximate Riemann solver for the convective terms and the operator spectral radius for the viscous terms - to compute Jacobians while keeping the complex flux function for the right-hand side term. But, we observed that this modification slow down the convergence of the whole process. We found very advantageous to fully differentiate the HLLC approximate Riemann solver,[8] the FEM viscous terms and the Spalart-Allmaras source terms.[3]

To achieve high efficiency, automation and robustness in the resolution of the non-linear system of algebraic equations to steady-state, it is mandatory to have a clever strategy to specify the time step. This is done by coupling local under-relaxation coefficient and local CFL, see Section D 4.

In this work, we have considered the symmetric Gauss-Seidel (SGS) relaxation. This linear system can be re-written:

$$(\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})\,\delta \mathbf{W}^n = \mathbf{R}^n + (\mathbf{L}\mathbf{D}^{-1}\mathbf{U})\,\delta \mathbf{W}^n$$

The following approximate system is used:

$$(\mathbf{D} + \mathbf{L})\mathbf{D}^{-1}(\mathbf{D} + \mathbf{U})\,\delta \mathbf{W}^n = \mathbf{R}^n \,.$$

Matrix $(\mathbf{D}+\mathbf{L})\mathbf{D}^{-1}(\mathbf{D}+\mathbf{U})$ can inverted in two sweeps which correspond to the LU-SGS approximate factorization:

$$\begin{aligned}
\text{Forward sweep:} \quad (\mathbf{D} + \mathbf{L})\,\delta \mathbf{W}^* &= \mathbf{R} \\
\text{Backward sweep:} \quad (\mathbf{D} + \mathbf{U})\,\delta \mathbf{W} &= \mathbf{D}\delta \mathbf{W}^* \,.
\end{aligned}$$

This sweeps can be written point wise:

$$\begin{aligned}
\delta W_i^* &= D_{ii}^{-1}\Big(R_i - \sum_{j \in \mathcal{L}(i)} L_{ij}\delta W_j^*\Big) \\
\delta W_i &= \delta W_i^* - D_{ii}^{-1} \sum_{j \in \mathcal{U}(i)} U_{ij}\delta W_j^* \,.
\end{aligned}$$

[a] In comparison, the LU-SGS method works well for the compressible Euler equation

where $\mathcal{L}(i)$ (resp. $\mathcal{U}(i)$) is the set of vertices with an index lower (resp. upper) that i. The lower and upper parts can be stored or not (i.e., matrix-free) as choice between efficiency or memory requirements.

In the SGS relaxation, we first zero the unknown: $\delta W^0 = 0$. Then, k_{max} sub-iterations are made using forward and backward sweeps:

$$\begin{aligned} (\mathbf{D}+\mathbf{L})\,\delta\mathbf{W}^{k+1/2} &= \mathbf{R}-\mathbf{U}\,\mathbf{W}^{k} \\ (\mathbf{D}+\mathbf{U})\,\delta\mathbf{W}^{k+1} &= \mathbf{R}-\mathbf{L}\,\mathbf{W}^{k+1/2}\,. \end{aligned}$$

or rewritten point wise:

$$\begin{aligned} \delta W_i^{k+1/2} &= D_{ii}^{-1}\Big(R_i - \sum_{j\in\mathcal{L}(i)} L_{ij}\delta W_j^{k+1/2} - \sum_{j\in\mathcal{U}(i)} U_{ij}\delta W_j^{k}\Big) \\ \delta W_i^{k+1} &= D_{ii}^{-1}\Big(R_i - \sum_{j\in\mathcal{U}(i)} U_{ij}\delta W_j^{k+1} - \sum_{j\in\mathcal{L}(i)} L_{ij}\delta W_j^{k+1/2}\Big)\,. \end{aligned}$$

For one sub-iteration, the SGS method is equivalent to the LU-SGS method.

V.D.3. Time step computation

The local time step is computed at each vertex:

$$\delta t = \mathrm{CFL}\,\frac{h^2}{h\,(c+\|\mathbf{u}\|)},$$

where h represents the minimal altitude of the cell of the vertex.

V.D.4. CFL Law

Many CFL laws exist in the literature - linear, geometric, residual based, ... - but these laws generally require parameters that are difficult to establish optimally because they depend on the considered flow, the geometry and the size of the mesh. In other words, they dependent too much user's data. But, they are mandatory to achieve fast convergence in solving non-linear equations.

To avoid this issue, Luke et al. proposed a new approach[42] based on bounding the primitive variables, ρ, p and T, variations at each time step. More precisely, initially we allowed the maximal time step at each vertex, then this local time step is truncated such that the change in ρ, p and T are below a user given percentage η. But, the change in primitive variables during a given interval of time has to be estimated. A way to accomplish this is to solve an explicit time-integration step to describe a functional relationship between time and primitive variable. Notice that it is done before assembling the matrix and the truncated local time step is used to compute the mass matrix.

This method achieves a maximal efficiency as each vertex is progressing at its own optimal time step. But, that choice is made from an estimation before the linear system resolution, thus there is no guarantee on the convergence of the Newton's method.

Another attractive approach has been proposed by Burgess and Glasby[12] which couples under-relaxation coefficient and dynamic CFL. Here, the solution is analyzed at each step of the Newton's method (after solving the linear system) and before updating the solution. First, the change in primitive variables, ρ and p, is again controlled by a user given percentage η and defines a under-relaxation coefficient ω^n at each step of the process. This global coefficient is then applied to the solution evolution: $W^{n+1} = W^n + \omega^n \delta W$. Then, the CFL value is updated depending on that under-relaxation coefficient:

$$CFL^{n+1} = \begin{cases} 0.1\,CFL^n & \text{if } \omega^n < 0.1 \\ CFL^n & \text{if } 0.1 \le \omega^n < 1 \\ \alpha CFL^n + \beta & \text{if } \omega^n = 1 \end{cases}$$

where we choose $\alpha = 1$ and $\beta = 1$ for a linear increase or $\alpha = 2$ and $\beta = 0$ for a geometric increase. This adaptive CFL, thus time step, is attractive because it is based on the behavior of the Newton's method. To improve even more the robustness of the method, they propose to set the solution update to zero when the value of ω^n is less than 0.1, *i.e.,* $\omega^n = 0$.

This approach is extremely robust because if the Newton's method diverges, the current step is cancelled and the time step, via the CFL, is automatically reduced. But the considered criterium is global and hence one bad vertex in the mesh can kill the overall efficiency by not allowing the CFL to grow.

In `Wolf`, we consider an hybrid method having the efficiency of the first method and the robustness of the second one. We proceed exactly like the second approach but the under-relaxation coefficient is set locally, *i.e.,* vertex-wise, and each vertex is supplied with its own CFL coefficient which evolves with respect to its own under-relaxation coefficient. Thus, we have a local time step and a local CFL for each vertex.

VI. Correctors for the Euler equations

In this section, we first derive the linear error equation corresponding to (38) in the case of the Euler equations. We then turn this error model into a non linear error equation.

CORRECTOR IN A LINEAR SETTING. In order to apply the previous analysis to the set of Euler equations, we use the finite-volume/Galerkin equivalence.[48] We do not mention boundary conditions that are expressed by boundary integrals ("BI"). The numerical and continuous solutions of the Euler equations are given by the following equations:

$$\iint \phi \,\mathrm{div}\mathcal{F}(W) \;+\; BI = \;0 \text{ and } \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h) \;+\; BI_h = \;0,$$

where the discrete fluxes are simply:

$$\mathcal{F}_h(.) = \Pi_h \mathcal{F}(\Pi_h .).$$

To build the *a posteriori* estimate, we combine previous equations, the added last term is null:

$$\iint \phi \,\mathrm{div}(\mathcal{F}(W) - \mathcal{F}(W_h)) \;+\; BI \;=\; -\iint \phi \,\mathrm{div}\mathcal{F}(W_h) + \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h) \;+\; BI$$

By using a Taylor expansion of $\mathcal{F}$, we get:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}}{\partial W}\delta w \;=\; -\iint \phi \,\mathrm{div}\mathcal{F}(W_h) + \iint \phi_h \,\mathrm{div}\mathcal{F}_h(W_h),$$

where δw is the implicit error $\delta w = \Pi_h W - W_h$ By using the Green formula, we obtain:

$$\iint \phi \,\mathrm{div}\frac{\partial \mathcal{F}}{\partial W}\delta w \;=\; +\iint \mathcal{F}(W_h)\nabla\phi - \iint \mathcal{F}_h(W_h)\nabla\phi_h. \tag{40}$$

In practice we take $\phi_h = 0$, then we discretise. Note that BI terms remain. In contrast to the elliptic case, the RHS will not be zero when ϕ is replaced by ϕ_h. To use this estimate, we assume we can recover from W_h a smooth solution $R_h(W_h)$ enjoying the following properties

1. $R_h(W_h)$ is smooth,
2. $R_h(W_h)$ interpolates W_h in the sense that $\Pi_h R_h(W_h) = W_h$.

We can build a quadratic representation of $R_h(W_h)$ from W_h. The proposed quadratic scheme uses $\mathbf{P}^2$ Lagrange test functions in triangle K to reconstruct a quadratic representation of the solution on K. This interpolation requires the solution nodal value at triangle vertices P_0, P_1 and P_2 , and the solution at the triangle mid-edges. We denote by P_3, P_4 and P_5 the middle of edges $\vec{e}_2$, $\vec{e}_0$ and $\vec{e}_1$, respectively. We denote by $(W_i)_{i=0,2}$ the nodal value of W_h and ∇W_i the nodal gradient of W_h. The quadratic scheme is given by:

$$R_h(W_h)(P) = \sum_{i=0}^{2} \psi_i(P)\, W_h(P_i) + \sum_{i=3}^{5} \psi_i(P)\, R_h(W_h)(P_i),$$

with:

$$\begin{cases} \psi_i(P) = \beta_i(P)\,(2\,\beta_i(P) - 1) & \text{for } i = 0, ..., 2\,, \\ \psi_i(P) = 4\,\beta_{[i]}(P)\,\beta_{[i+1]}(P) & \text{for } i = 3, ..., 5. \end{cases}$$

The mid-node values $(R_h(W_h)(P_i))_{i=3,5}$ are recovered from the nodal gradients of vertices P_0, P_1, P_2:

$$\begin{aligned}
R_h(W_h)(P_3) &= \tfrac{1}{2}(W_1 + W_2) + \tfrac{1}{4}(\nabla W_1 + \nabla W_2) \cdot P_1P_2, \\
R_h(W_h)(P_4) &= \tfrac{1}{2}(W_0 + W_2) + \tfrac{1}{4}(\nabla W_2 + \nabla W_0) \cdot P_2P_0, \\
R_h(W_h)(P_5) &= \tfrac{1}{2}(W_0 + W_1) + \tfrac{1}{4}(\nabla W_0 + \nabla W_1) \cdot P_0P_1.
\end{aligned}$$

With this scheme, we have $R_h(W_h)(P_i) = W_i$ for the nodal values, and R_h is quadratic, such that properties 1. and 2. hold. Then the last term of (40) is still zero but can be used for building an kind of interpolation error:

$$\iint \phi \,\mathrm{div} \frac{\partial \mathcal{F}}{\partial W} \delta w \;\approx\; + \iint \nabla\phi_h \,(\mathcal{F}(W_h) - \mathcal{F}_h(W_h)). \tag{41}$$

To get the *a priori* estimate, we combine the two states equations, where the added last term is null:

$$\int\int \phi_h \,\mathrm{div}(\mathcal{F}_h(W_h) - \mathcal{F}_h(W)) \;+\; BI_h = \;-\int\int \phi_h \mathrm{div}\mathcal{F}_h(W) + \int\int \phi_h \mathrm{div}\mathcal{F}(W).$$

In a similar way, by using a Taylor expansion and Green formula, we have:

$$\iint \phi \,\mathrm{div} \frac{\partial \mathcal{F}_h}{\partial W}(-\delta w) \;=\; + \iint \mathcal{F}_h(W)\nabla\phi_h - \iint \mathcal{F}(W)\nabla\phi_h. \tag{42}$$

To derive the corrector equation, we just have to replace W by W_h:

$$\iint \phi \,\mathrm{div} \frac{\partial \mathcal{F}_h}{\partial W}(-\delta w) \;=\; + \iint \nabla\phi_h \,(\mathcal{F}_h(W_h) - \mathcal{F}(W)). \tag{43}$$

We observe that (41) and (43) are close to each other.

Previous correctors are straightforward to implement when an adjoint solver is already integrated in the flow solver. However, they have several weaknesses:

- There is no guarantee that the corrected solution $W_h + \delta w$ will be physical,
- the approximation of the discrete fluxes $\mathcal{F}_h$ can be far the real numerical fluxes $\mathbf{\Phi_{ij}}$.

In addition, it is also very complex to apply the numerical fluxes $\mathbf{\Phi_{ij}}$ to the recovered solution $R_h(W_h)$. Indeed, as the nodal values are the same, it is necessary to take into account the gradient variation (during the extrapolation) in the different flux choices (fourth or sixth order dissipation). Another simpler strategy is used non linear correctors.

CORRECTOR IN A NON LINEAR SETTING. Instead of solving a linear equation to find the corrector, we build an error equation that consists in adding a source term to reduce a defect. This residual is close to the second member found in Equations (41) and (43). In the *a priori* setting, the defect is the residual of numerical flux applied to the continuous solution. Indeed, optimally, we want

$$\mathrm{div}(\mathcal{F}_h(W)) = 0, \quad \textit{(a priori)} \tag{44}$$

when W is the exact solution and $\mathcal{F}_h$ represents the numerical scheme. Consequently, the corrected solution W_c of the current discrete solution W_h is found by solving the non linear equation:

$$\mathrm{div}(\mathcal{F}_h(W_c)) = -S_h(W),$$

where $S_h(W)$ is the source term computed from the divergence of $\mathcal{F}_h(W)$. In our approach W is replaced by the smooth recovery $R_h(W_h)$ and we use locally a finer mesh to take into account the nodal values at mid-edges. $S_h(W)$ is then approximated as:

$$S_h(W) \approx \mathrm{div}(\mathcal{F}_{h/2}(R_h(W_h))).$$

A transfer procedure is used to accumulate $\mathcal{F}_{h/2}(R_h(W_h))$ on the coarser mesh h. In contrast with the linear approach, the only modification in the flow solver is to take into account a residual source term. The corrected solution is found (instead of the implicit error) and the corrected solution is guarantee to be physical.

In the *a posteriori* setting, the defect is the gap between the continuous PDE with respect to the numerical solution. Indeed, ideally, W_h should verify

$$\mathrm{div}(\mathcal{F}(W_h)) = 0, \quad \textit{(a posteriori)} \tag{45}$$

meaning that the numerical solution solves exactly the continuous set of Euler equations. In this case, the (exact) source term is the divergence of $\mathcal{F}(W_h)$. The continuous fluxes $\mathcal{F}$ are approximated on locally refined grids, so that the source term becomes the divergence of $\mathcal{F}_{h/2}(W_h)$. The solution on the finer mesh is interpolation from the coarser mesh and the source term computed on $h/2$ is accumulated back from the finer to the coarser mesh.

From a practical point of view, the finer grids are never generated as we can solve local problem to compute the source terms. The flow solver is then used to inverse the error equation directly. The procedure to derive the corrected solution is then:

Step 1.: Solve the flow problem to get the numerical solution W_h

Step 2.: Compute the source term on finer mesh $-S_h(W)$ or $S(W_h)$.

Step 3.: From W_h, converge again the solution with the residual source term added.

To perform mesh adaptation, then the second member of the adjoint equation is $W_c - W_h$. Then a standard metric-based goal-oriented metric is used. The norm of the implicit error is then controlled.

Fin de partie morte

VII. Numerical experiments

In this section, we focus on the validation of the correctors. We only consider the *a posteriori* corrector. In indeed, the *a priori* corrector is more delicate to used practically, especially with flows with shocks. If a limiter is used to recover $R_h(W_h)$, the recover solution may equal the the initial solution. On the contrary, with the *a posteriori* corrector, only the continuous flux is approximated on a locally finer mesh while the solution is considered linearly interpolated.

SUBSONIC BUMP EXAMPLE. We consider the extruded bump geometry[b], see Figure 1 with an inflow at Mach 0.3. We consider a sequence of uniform meshes and we compare the lift and drag computed from the initial solution and the corrected flow field. As the corrector is on the flow field, all the functional of interest can be corrected simultaneously. For the lift, we also observe that the corrected lift converges at a higher rates that the uncorrected version, see Figure 2. For each size of mesh, the corrected lift is 2 to 4 times smaller with the corrected flow field.

Conclusion

A first step in combining into a single formalism mesh adaptation and solution correction strategies is given. It is based on *a priori* or *a posteriori* analysis of the different component of the error arising when discretizing a PDE: interpolation error, approximation error and implicit error. The interpolation is completely controlled by the the derivatives of the order of the scheme. For implicit error for the Euler equations, an interpolation of the Euler fluxes is weighted by the gradient of the adjoint state with respect to the observation functional. The optimal metric is deduced

[b] from `http://turbmodels.larc.nasa.gov/bump.html`

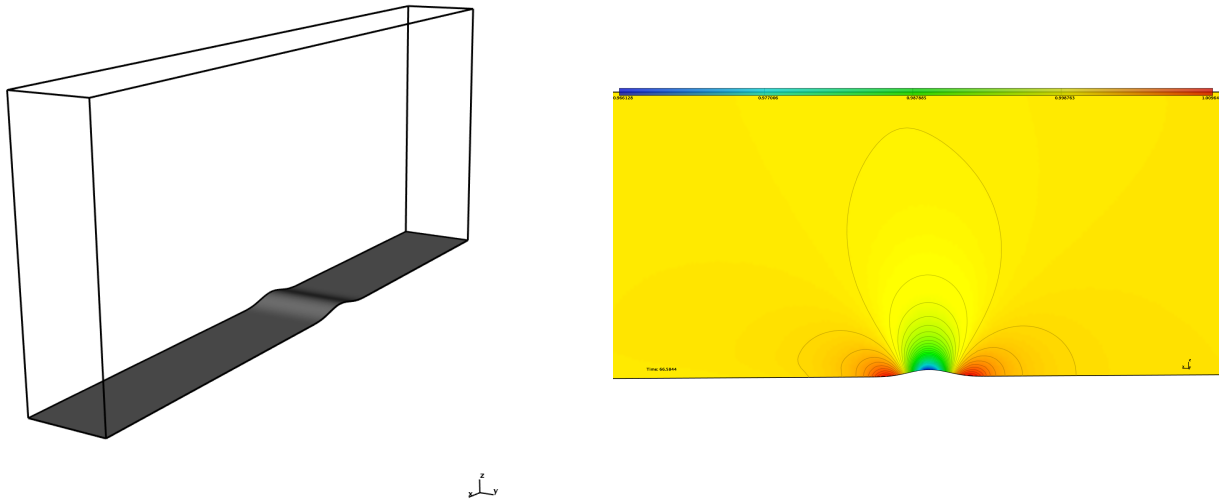
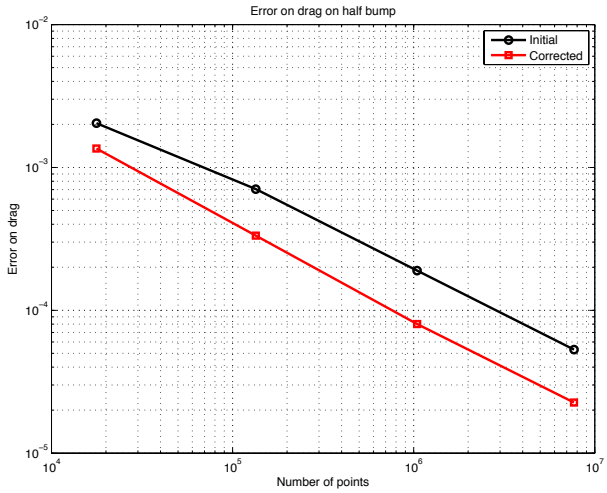
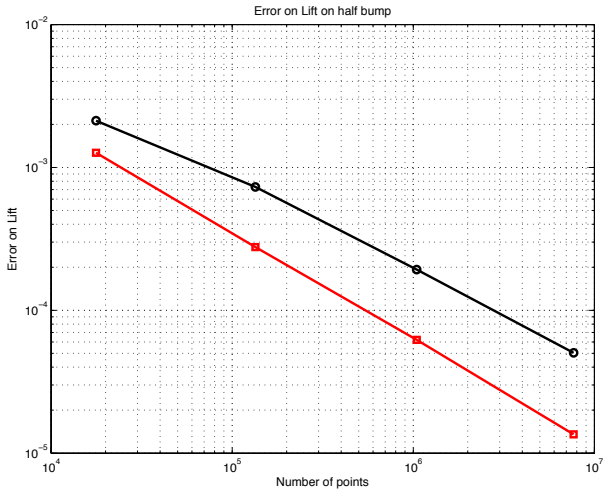
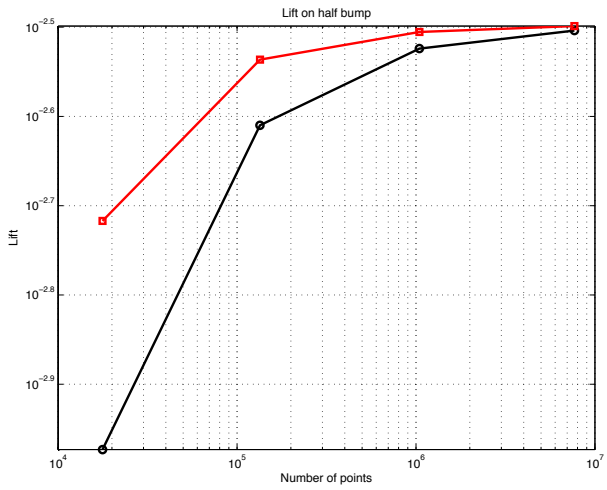


Figure 1. 3D bump geometry (left) and density flow field (right).







# vertices	Error on lift	Error on corrected lift
17723	66%	39%
134381	22 %	8.6%
1044943	6%	1.9%
7682230	1.5%	0.4%

Figure 2. 3D bump error on drag (top left), error on lift (top right), computed lift (bottom lift), error to the lift target without and with corrections (bottom right). Plain black line are uncorrected values while plain red lines are corrected values.

from the interpolation kernel with a sum of weighted Hessian. Finally, for the norm-oriented functional, a bi-adjoint strategy is used in order to derive a corrector term. It is then used for any sub-sequent functionals and by using the traditional goal-oriented strategy.

References

[1]

[2]F. Alauzet. `Wolf`, Euler and Navier-Stokes flow solver. RT-, INRIA, 2001.

[3]F. Alauzet. `Wolf` user guide. An edge-based Navier-Stokes flow solver based on the MEV numerical scheme. Internal report, INRIA, 2010.

[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. *Journal of Computational Physics*, 229:561–593, 2010.

[6]T.J. Barth. Aspects of unstructured grids and finite-volume solver for the Euler and Navier-Stokes equations. In *Von Karman Institute Lecture Series*, 1994.

[7]P. Batten, N. Clarke, C. Lambert, and D. M. Causon. On the choice of wavespeeds for the HLLC riemann solver. *SIAM J. Sci. Comput.*, 18(6):1553–1570, 1997.

[8]P. Batten, M.A. Leschziner, and U.C. Goldberge. Average-state Jacobians and implicit methods for compressible viscous and turbulent flows. *J. Comp. Phys.*, 137:38–78, 1997.

[9]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.

[10]A. Belme, A. Dervieux, and F. Alauzet. A fully anisotropic goal-oriented mesh adaptation for unsteady flows. In *Proceedings of the V ECCOMAS CFD Conf.*, 2010.

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

[12]N.K. Burgess and R.S. Glasby. Advances in numerical methods for CREATE-AV analysis tools. AIAA Paper2014-0417, National Harbor, MD, USA, Jan 2014.

[13]M. J. Castro-Díaz, F. Hecht, B. Mohammadi, and O. Pironneau. Anisotropic unstructured mesh adaptation for flow simulations. *Int. J. Numer. Meth. Fluids*, 25:475–491, 1997.

[14]L. Chen, P. Sun, and J. Xu. Optimal anisotropic meshes for minimizing interpolation errors in L^p-norm. *Math. Comp.*, 76(257):179–204, 2007.

[15]G. Compère, J.-F. Remacle, J. Jansson, and J. Hoffman. A mesh adaptation framework for dealing with large deforming meshes. *Int. J. Numer. Meth. Engng*, 82:843–867, 2010.

[16]T. Coupez. Génération de maillages et adaptation de maillage par optimisation locale. *Revue Européenne des Éléments Finis*, 9:403–423, 2000.

[17]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.

[18]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.

[19]C. Debiez and A. Dervieux. Mixed-Element-Volume MUSCL methods with weak viscosity for steady and unsteady flow calculations. *Comput. & Fluids*, 29:89–118, 2000.

[20]A. Dervieux, A. Loseille, and F. Alauzet. High-order adaptive method applied to high speed flows. In *Proceedings of WEHSFF2007*, 2007.

[21]C. Dobrzynski and P. J. Frey. Anisotropic delaunay mesh adaptation for unsteady simulations. In *Proc. of 17th Int. Meshing Rountable*, pages 177–194. Springer, 2008.

[22]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.

[23]L. Formaggia and S. Perotto. New anisotropic a prioiri error estimate. *Numer. Math.*, 89:641–667, 2001.

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

[25]P.J. Frey and P.-L. George. *Mesh generation. Application to finite elements*. ISTE Ltd and John Wiley & Sons, 2nd edition, 2008.

[26]P.L. George. `Gamanic3d`, adaptive anisotropic tetrahedral mesh generator. Technical Note, INRIA, 2003.

[27]M. B. Giles. On adjoint equations for error analysis and optimal grid adaptation in CFD. Technical Report NA-97/11, Oxford, 1997.

[28]M. B. Giles and N. Pierce. Improved lift and drag estimates using adjoint euler equations. *AIAA Paper*, 1999-3293, 1999.

[29]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.

[30]N. Gourvitch, G. Rogé, I. Abalakin, A. Dervieux, and T. Kozubskaya. A tetrahedral-based super convergent scheme for aeroacoustics. RR-5212, INRIA, May 2004.

[31]W. Huang. Metric tensors for anisotropic mesh generation. *J. Comp. Phys.*, 204:633–665, 2005.

[32]A. Jameson and S. Yoon. Lower-Upper implicit schemes with multiple grids for the Euler equations. *AIAA Journal*, 25(7):929–935, 1987.

[33]W. T. Jones, E. J. Nielsen, and M. A. Park. Validation of 3D adjoint based error estimation and mesh adaptation for sonic boom prediction. *AIAA Paper*, 2006-1150, 2006.

[34]B. Van Leer. Towards the ultimate conservative difference scheme i. The quest of monotonicity. *Lecture notes in physics*, 18:163–168, 1973.

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

[36]A. Loseille. *Adaptation de maillage 3D anisotrope multi-échelles et ciblé à une fonctionnelle. Application à la prédiction haute-fidélité du bang sonique.* PhD thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008.

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

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

[39]A. Loseille, F. Alauzet, A. Dervieux, and P. J. Frey. Achievement of second order mesh convergence for discontinuous flows with adapted unstructured mesh adaptation. *AIAA Paper*, 07-4186, 2007.

[40]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.

[41]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.

[42]E. Luke, X.-L. Tong, J. Wu, L. Tang, and P. Cinnella. A step towards shape shifting algorithms: Reacting flow simulations using generalized grids. AIAA Paper2001-0897, Reno, NV, USA, Jan 2001.

[43]H. Luo, J.D. Baum, and R. Löhner. A fast, matrix-free implicit method for compressible flows on unstructured grids. *J. Comp. Phys.*, 146:664–690, 1998.

[44]H. Luo, J.D. Baum, and R. Löhner. An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids. *Comput. & Fluids*, 30:137–159, 2001.

[45]F. Magoules. *Computational Fluid Dynamics*. CRC Press, Boca Raton, London, New York, Washington D.C., 2011.

[46]Victorien Menier, A. Loseille, and F. Alauzet. *Multi-scale mesh adaptation for viscous flows*. American Institute of Aeronautics and Astronautics, 2014/12/01 2014.

[47]K. Mer. Variational analysis of a mixed element/volume scheme with fourth-order viscosity on general triangulations. *Comput. Methods Appl. Mech. Engrg.*, 153:45–62, 1998.

[48]Katherine Mer. Variational analysis of a mixed finite element : finite volume scheme on general triangulation. Technical Report RR-2213, March 1994.

[49]C. C Pain, A. P. Umpleby, 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.

[50]M.A. Park. Adjoint-based, three-dimensional error prediction and grid adaptation. *AIAA Paper*, 2002-3286, 2002.

[51]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. *Computers & Mathematics with Applications*, 52:1213–1242, 2006.

[52]D. Sharov, H. Luo, J.D. Baum, and R. Löhner. Implementation of unstructured grid GMRES+LU-SGS method on shared-memory, cache-based parallel computers. *AIAA Paper*, 2000-0927, 2000.

[53]D. Sharov and K. Nakahashi. Reordering of hybrid unstructured grids for Lower-Upper Symmetric Gauss-Seidel computations. *AIAA Journal*, 36(1):484–486, 1997.

[54]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.

[55]Y.V. Vasilevski and K.N. Lipnikov. An adaptive algorithm for quasioptimal mesh generation. *Comput. Math. Math. Phys.*, 39(9):1468 1486, 1999.

[56]Y.V. Vasilevski and K.N. Lipnikov. Error bounds for controllable adaptive algorithms based on a hessian recovery. *Computational Mathematics and Mathematical Physics*, 45(8):1374–1384, 2005.

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