

Comparing anisotropic adaptive strategies on the 2nd AIAA sonic boom workshop geometry

Adrien Loseille, Loic Frazza, Frédéric Alauzet

► **To cite this version:**

Adrien Loseille, Loic Frazza, Frédéric Alauzet. Comparing anisotropic adaptive strategies on the 2nd AIAA sonic boom workshop geometry. *Journal of Aircraft*, American Institute of Aeronautics and Astronautics, 2018, pp.1-15. hal-01962175

HAL Id: hal-01962175

<https://hal.inria.fr/hal-01962175>

Submitted on 20 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparing anisotropic adaptive strategies on the 2nd AIAA sonic boom workshop geometry

Adrien Loseille*, Loïc Frazza[†] and Frédéric Alauzet[‡]
INRIA Saclay-Ile de France, 1 Rue Honoré d'Estienne d'Orves, 91 120 Palaiseau, France

The recent release of the 2nd AIAA sonic boom geometry offers the opportunity to review the classical anisotropic adaptive strategies for complex geometries. The design of Mach-aligned tailored grids is also a great challenge to see how adaptivity can compete with user-defined tailored grids. Two classical adaptive strategies, multi-scale and goal-oriented, are compared with the results obtained on tailored grids. For the flow solver, we discuss several low-dissipation numerical schemes of order 4th and 6th with respect to regular 2nd order scheme both on inviscid and RANS flow conditions. We finally perform a non-linear error analysis to assess the convergence of the sequence of adaptive meshes with respect to tailored grids. All results and discussions are based on the C25D baseline geometry.

I. Introduction

Anisotropic mesh adaptation has been designed to automatically take into account the anisotropic features of the physical phenomena under study. In this respect, the computation of near field signatures of supersonic aircraft have been one of the primary field of applications (and success) of anisotropic mesh adaptation [1].

Over the past decade, anisotropic mesh adaptation has gained in maturity thanks to the improvements of the meshing algorithms, error estimates, and numerical schemes. Most of typical second-order flow solvers are now naturally compatible with highly stretched anisotropic elements. The use of anisotropic mesh adaptation, especially for supersonic steady flows, leads to the following observations : (i) an early capturing of the physical phenomena (not only shocks but also shear layers, contact discontinuities, vortices, . . .), (ii) a second-order convergence for flows with shocks, (iii) an optimized ratio CPU time over degrees of freedom, . . . However, in the mean time, the complexity of the geometries has increased similarly, so that the aforementioned features are only observed when a special care is used to design and combine all the components of the adaptive loop: error estimates, flow solver, adaptive meshing algorithms,

Complex low-boom geometries are particularly challenging as they tend to produce soft and small shocks that are difficult to predict numerically. One example is the Quiet Spike concept from Gulfstream Aerospace. These phenomena are thus important to predict accurately the ground signature. Assessing the near field signatures in this context for both

*Research scientist, AIAA member

[†]Phd Student, AIAA student member

[‡]Research scientist, AIAA member.

inviscid and viscous flows is still a challenge. The 2nd AIAA sonic boom workshop, is a unique occasion to assess and compare the performance of different solvers [2] [3] [4], approaches and meshing techniques to solve this case.

The main process used during the workshop is inherited from Gulfstream procedure [5] to produce tailored meshes aligned with the Mach cone. It can be summarized as it. First, the complex geometry is embedded in a cylinder up to $R/L = 0.3$ (or below). In this domain (aircraft-cylinder), a fully 3D body-fitted unstructured mesh is generated. Then, starting from the cylinder to the far-field, a prismatic mesh is extruded aligned with the Mach cone. This procedure is really fast, only a small 3D domain needs to be meshes with standard mesh techniques as constrained-Delaunay or advancing front.

The primary intent of these meshes is to drastically reduce solver diffusion when predicting the shock waves propagation. The performance of this approach in near-field (aircraft-cylinder) has been studied in [6] and the resulting propagation in [7]. In addition, the level of anisotropy along the Mach cone and the density of the mesh in the azimuthal direction is controlled. Details on the procedure are included in [8].

Comparing adaptive results with these meshes is then also particularly challenging. In this paper, we only consider the C25D geometry, see Figure 1 (left), with flow through and powered nacelle. The geometry is described in [9]. For the comparisons, we keep the same far field domain definition for the adaptive case, see Figure 1 (right).

For the adaptive computations, we only consider Hessian-based and goal-oriented approaches based on fully unstructured meshes. Both of them naturally yields an anisotropic prescription of the optimal meshes and are independent to the level of complexity of the geometry. For the Hessian-based approach, we control the interpolation on the Mach number in L^2 norm while for the goal-oriented, the pressure functional is observed in the symmetry plane for $z < 0$. For each case, sequences of meshes with an increasing complexity are generated. Best effort meshes for each strategies are then compared with signature obtained on tailored meshes.

Finally, we perform an non-linear error analysis to provide point-wise error estimates of the implicit error on the signature for goal-oriented and tailored meshes. This analysis described in [10] is a non-linear extension of the correction analysis described in [11].

The paper is organized as follows. In Section II, we describe the flow solver numerical schemes used for inviscid and RANS flow conditions. In Section III, we briefly describe the adaptive mesh procedure based on a metric field. Section IV is devoted to the description of the feature-based and goal-oriented error estimates. We finally discuss the numerical results on the sonic boom geometry and we provide an uncertainty analysis in Section V.

II. Finite Volume - Finite Element Flow solver

We describe in this section our in-house flow solver `Wolf`. We also discuss convergence history and signature convergence analysis on the sequence of provided tailored meshes. This allows us to compare the accuracy of the

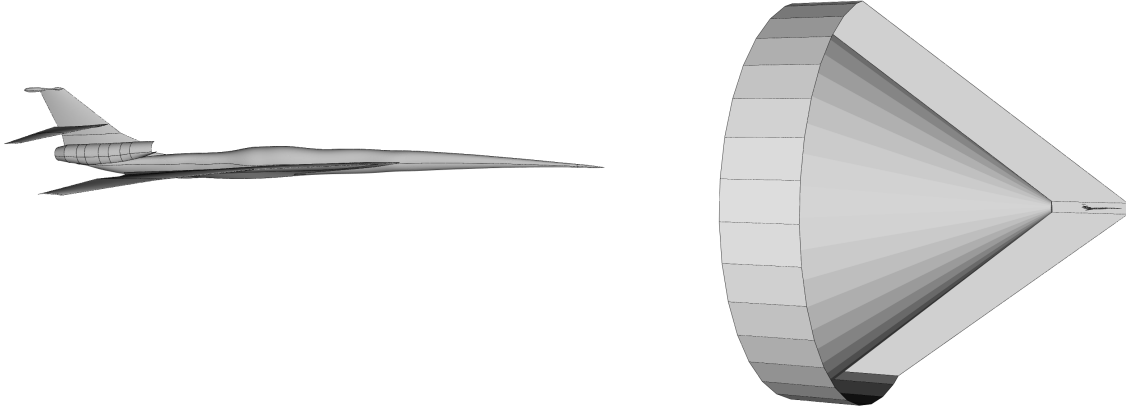


Fig. 1 C25 Geometry (left) and typical computational domain aligned with Mach cone (right).

schemes when different dissipation orders are used.

The compressible Navier-Stokes equations for mass, momentum and energy conservation read:

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \\ \frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = \nabla \cdot (\mu \mathcal{T}), \\ \frac{\partial (\rho e)}{\partial t} + \nabla \cdot ((\rho e + p) \mathbf{u}) = \nabla \cdot (\mu \mathcal{T} \mathbf{u}) + \nabla \cdot (\lambda \nabla T), \end{array} \right. \quad (1)$$

where ρ denotes the density (kg/m^3), \mathbf{u} the velocity (m/s), e the total energy per mass ($m^2 \cdot s^{-2}$), p the pressure (N/m^2), T the temperature (K), μ the laminar dynamic viscosity ($kg/(m \cdot s)$) and λ the laminar conductivity. \mathcal{T} the laminar stress tensor:

$$\mathcal{T} = (\nabla \otimes \mathbf{u} + {}^t \nabla \otimes \mathbf{u}) - \frac{2}{3} \nabla \cdot \mathbf{u} \mathbb{I},$$

where (in 3D) $\mathbf{u} = (u, v, w)$ and

$$\nabla \cdot \mathbf{u} \mathbb{I} = \begin{pmatrix} u_x + v_y + w_z & 0 & 0 \\ 0 & u_x + v_y + w_z & 0 \\ 0 & 0 & u_x + v_y + w_z \end{pmatrix},$$

where $u_x = \frac{\partial u}{\partial x}$, $u_y = \frac{\partial u}{\partial y}$, $u_z = \frac{\partial u}{\partial z}$ (idem for v and w).

The variation of nondimensionalized laminar dynamic viscosity and conductivity coefficients μ and λ as a function

of a dimensional temperature T is defined by Sutherland's law:

$$\mu = \mu_\infty \left(\frac{T}{T_\infty} \right)^{\frac{3}{2}} \left(\frac{T_\infty + \text{Su}}{T + \text{Su}} \right) \quad \text{and} \quad \lambda = \lambda_\infty \left(\frac{T}{T_\infty} \right)^{\frac{3}{2}} \left(\frac{T_\infty + \text{Su}}{T + \text{Su}} \right),$$

where $\text{Su} = 110$ is the Sutherland constant and the index ∞ denotes reference quantities. The relation linking μ and λ is expressed from the Prandtl laminar number:

$$\text{Pr} = \frac{\mu C_p}{\lambda} \quad \text{with} \quad \text{Pr} = 0.72 \quad \text{for (dry) air,}$$

where C_p is the specific heat at constant pressure.

A. Turbulence modeling

According to the standard approach to turbulence modeling based upon the Boussinesq hypothesis, the turbulence is modeled with an eddy viscosity μ_t , which is added to the laminar (or dynamic) viscosity, μ . The dynamic viscosity is usually taken to be a function of the temperature, whereas μ_t is obtained using a turbulence model. Here we chose the Spalart-Allmaras one equation turbulence model [12] given by the following equation:

$$\frac{\partial \tilde{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\nu} = c_{b1} [1 - f_{t2}] \tilde{S} \tilde{\nu} - \left[c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2 + \frac{1}{\sigma} \left[\nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu}) + c_{b2} \|\nabla \tilde{\nu}\|^2 \right] + f_{t1} \Delta \mathbf{u}^2, \quad (2)$$

where $\tilde{\nu}$ is the turbulent kinematic viscosity and all the constants are defined below. In the standard model the trip term is being left out, *i.e.*, $f_{t1} = 0$. Moreover, some implementations also ignore the f_{t2} term as it is argued that if the trip is not included, then f_{t2} is not necessary [13]. In `WOLF`, this simplified version has been considered and we prefer to write it under the following form, which is more appropriate for its discretization with the finite element/finite volume method. Indeed, Equation (2) can be decomposed into the following terms:

$$\frac{\partial \rho \tilde{\nu}}{\partial t} + \underbrace{\mathbf{u} \cdot \nabla \rho \tilde{\nu}}_{\text{convection}} = \underbrace{c_{b1} \tilde{S} \rho \tilde{\nu}}_{\text{production}} - \underbrace{c_{w1} f_w \rho \left(\frac{\tilde{\nu}}{d} \right)^2}_{\text{destruction}} + \underbrace{\frac{\rho}{\sigma} \nabla \cdot ((\nu + \tilde{\nu}) \nabla \tilde{\nu})}_{\text{dissipation}} + \underbrace{\frac{c_{b2} \rho}{\sigma} \|\nabla \tilde{\nu}\|^2}_{\text{diffusion}}.$$

Note that this is not a conservative model. If a conservative form of the Spalart-Allmaras is foreseen, we have to consider the variation proposed by Catris and Aupoix [14]. The turbulent eddy viscosity is computed from:

$$\mu_t = \rho \tilde{\nu} f_{v1},$$

where

$$f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3} \quad \text{and} \quad \chi = \frac{\tilde{\nu}}{\nu} \quad \text{with} \quad \nu = \frac{\mu}{\rho}.$$

Additional definitions are given by the following equations:

$$f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}} \quad \text{and} \quad \tilde{S} = \Omega + \frac{\tilde{v}}{\kappa^2 d^2} f_{v2} \quad \text{where} \quad \Omega = \|\nabla \times \mathbf{u}\|.$$

d is the distance to nearest wall which is computed for each vertex at the beginning of the simulation. The set of closure constants for the model is given by

$$\begin{aligned} \sigma &= \frac{2}{3}, & c_{b1} &= 0.1355, & c_{b2} &= 0.622, & \kappa &= 0.41, \\ c_{w1} &= \frac{c_{b1}}{\kappa} + \frac{1 + c_{b2}}{\sigma}, & c_{w2} &= 0.3, & c_{w3} &= 2, & c_{v1} &= 7.1. \end{aligned}$$

Finally, the function f_w is computed as:

$$f_w = g \left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6} \right)^{1/6} \quad \text{with} \quad g = r + c_{w2} (r^6 - r) \quad \text{and} \quad r = \min \left(\frac{\tilde{v}}{\tilde{S} \kappa^2 d^2}, 10 \right).$$

B. Vector form of the RANS system

We write the RANS system in the following (more compact) vector form:

$$W_t + F_1(W)_x + F_2(W)_y + F_3(W)_z = S_1(W)_x + S_2(W)_y + S_3(W)_z + Q(W),$$

where $S_i(W)_a = \frac{\partial S_i(W)}{\partial a}$ ($i = 1, 2, 3, a = x, y, z$) (idem for F). W is the nondimensionalized conservative variables vector:

$$W = (\rho, \rho u, \rho v, \rho w, \rho E, \rho \tilde{v})^T.$$

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

$$\begin{aligned} F_1(W) &= (\rho u, \rho u^2 + p, \rho uv, \rho uw, u(\rho E + p), \rho u \tilde{v})^T, \\ F_2(W) &= (\rho v, \rho uv, \rho v^2 + p, \rho vw, v(\rho E + p), \rho v \tilde{v})^T, \\ F_3(W) &= (\rho w, \rho uw, \rho vw, \rho w^2 + p, w(\rho E + p), \rho w \tilde{v})^T. \end{aligned} \tag{3}$$

$S(W) = (S_1(W), S_2(W), S_3(W))$ are the laminar viscous fluxes:

$$\begin{aligned} S_1(W) &= \left(0, \tau_{xx}, \tau_{xy}, \tau_{xz}, u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \lambda\mathcal{T}_x, \frac{\rho}{\sigma}(v + \tilde{v})\tilde{v}_x \right)^T, \\ S_2(W) &= \left(0, \tau_{xy}, \tau_{yy}, \tau_{yz}, u\tau_{xy} + v\tau_{yy} + w\tau_{yz} + \lambda\mathcal{T}_y, \frac{\rho}{\sigma}(v + \tilde{v})\tilde{v}_y \right)^T, \\ S_3(W) &= \left(0, \tau_{xz}, \tau_{yz}, \tau_{zz}, u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + \lambda\mathcal{T}_z, \frac{\rho}{\sigma}(v + \tilde{v})\tilde{v}_z \right)^T, \end{aligned} \quad (4)$$

where τ_{ij} are the components of laminar stress tensor defined by:

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial v_k}{\partial x_k} \delta_{ij}.$$

where (v_i, v_j, v_k) are the three components of the velocity and δ_{ij} the Kroneker symbol.

$Q(W)$ are the source terms, i.e. the diffusion, production and destruction terms from the Spalart-Allmaras turbulence model:

$$Q(W) = (0, 0, 0, 0, 0, \frac{c_{b2}\rho}{\sigma} \|\nabla \tilde{v}\|^2 + \rho c_{b1} \tilde{S} \tilde{v} + c_{w1} f_w \rho \left(\frac{\tilde{v}}{d} \right)^2)^T. \quad (5)$$

Note that $Q = 0$ in the case of the laminar Navier-Stokes equations, unless additional source terms are added (to take into account gravity, for instance).

C. Spatial discretization

The spatial discretization of the fluid equations (1) and (2) is based on a vertex-centered finite element/finite volume formulation on unstructured meshes. The equations are integrated on each finite volume cell C_i (using the Green formula):

$$|C_i| \frac{dW_i}{dt} + \mathbf{F}_i = \mathbf{S}_i + \mathbf{Q}_i, \quad (6)$$

where W_i is the mean value of the solution W on cell C_i , \mathbf{F}_i , \mathbf{S}_i and \mathbf{Q}_i are respectively the numerical convective, viscous and source flux terms:

$$\mathbf{F}_i = \int_{\partial C_i} F(W_i) \cdot \mathbf{n}_i d\gamma, \quad \mathbf{S}_i = \int_{\partial C_i} S(W_i) \cdot \mathbf{n}_i d\gamma, \quad \mathbf{Q}_i = \int_{C_i} Q(W_i) dx,$$

where \mathbf{n}_i is the outer normal to the finite volume cell surface ∂C_i , and F , S and Q are respectively the convective, viscous and source terms flux functions as defined previously in Relations (3), (4) and (5). It combines a HLLC approximate Riemann solver [15] to compute the convective fluxes and the Galerkin centered method to evaluate the viscous terms. Second order space accuracy is achieved through a piecewise linear extrapolation based on the Monotonic Upwind

Scheme for Conservation Law (MUSCL) procedure [16] which uses a particular edge-based formulation with upwind elements.

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_I and S_J , respectively) of all the waves present in the exact solution. If $S_I > 0$ then the flow is supersonic from left to right and the upwind flux is simply defined from $F(W_i)$ where W_i is the state to the left of the discontinuity. Similarly, if $S_J < 0$ then the flow is supersonic from right to left and the flux is defined from $F(W_j)$ where W_j is the state to the right of the discontinuity. In the more difficult subsonic case when $S_I < 0 < S_J$ we have to calculate $F(W_i^*)$ or $F(W_j^*)$. Consequently, the HLLC flux is given by:

$$\Phi_{ij}^{hllc}(W_i, W_j, \mathbf{n}_{ij}) = \begin{cases} F(W_i) \cdot \mathbf{n}_{ij} & \text{if } S_I > 0 \\ F(W_i^*) \cdot \mathbf{n}_{ij} & \text{if } S_I \leq 0 < S_M \\ F(W_j^*) \cdot \mathbf{n}_{ij} & \text{if } S_M \leq 0 \leq S_J \\ F(W_j) \cdot \mathbf{n}_{ij} & \text{if } S_J < 0 \end{cases} .$$

W_i^* and W_j^* are evaluated as follows. We denote by $\eta = \mathbf{u} \cdot \mathbf{n}$. Assuming that $\eta^* = \eta_i^* = \eta_j^* = S_M$, the following evaluations are proposed [15] (the subscripts i and j are omitted for clarity):

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

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

$$S_M = \frac{\rho_j \eta_j (S_J - \eta_j) - \rho_i \eta_i (S_I - \eta_i) + p_i - p_j}{\rho_j (S_J - \eta_j) - \rho_i (S_I - \eta_i)} ,$$

and the acoustic wave speeds based on the Roe average:

$$S_I = \min(\eta_i - c_i, \tilde{\eta} - \tilde{c}) \quad \text{and} \quad S_J = \max(\eta_j + c_j, \tilde{\eta} + \tilde{c}) .$$

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

2nd-order accurate version. The MUSCL type reconstruction method has been designed to increase the order of accuracy of the scheme [16]. The idea is to use extrapolated values W_{ij} and W_{ji} instead of W_i and W_j at the interface ∂C_{ij} to evaluate the flux. Note that, in the implementation, the primitive variables are extrapolated to guarantee the positivity of the density and the pressure, then the conservative variables are reconstructed from these values. Thus, the gradients of the primitive variables are evaluated. However, in the following, we still denote by W the primitive variables vector. The numerical flux becomes:

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

where W_{ij} and W_{ji} are linearly extrapolated as:

$$W_{ij} = W_i + \frac{1}{2} (\nabla W)_i \cdot \overrightarrow{P_i P_j} \quad \text{and} \quad W_{ji} = W_j + \frac{1}{2} (\nabla W)_j \cdot \overrightarrow{P_j P_i}.$$

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 $P_i P_j$, is implicitly defined along edge $P_i P_j$ by the relation:

$$(\nabla W)_{ij}^C \cdot \overrightarrow{P_i P_j} = W_j - W_i \quad \text{and} \quad (\nabla W)_{ij}^C \cdot \overrightarrow{P_j P_i} = W_i - W_j.$$

Upwind and downwind gradients, which are also related to edge $P_i P_j$, are computed according to the definition of upwind and downwind tetrahedra of edge $P_i P_j$. 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 $P_i P_j$, see Figure 2.

Upwind and downwind gradients are then defined for vertices P_i and P_j as:

$$(\nabla W)_{ij}^U = (\nabla W)|_{K_{ij}} \quad \text{and} \quad (\nabla W)_{ij}^D = (\nabla W)|_{K_{ji}}.$$

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

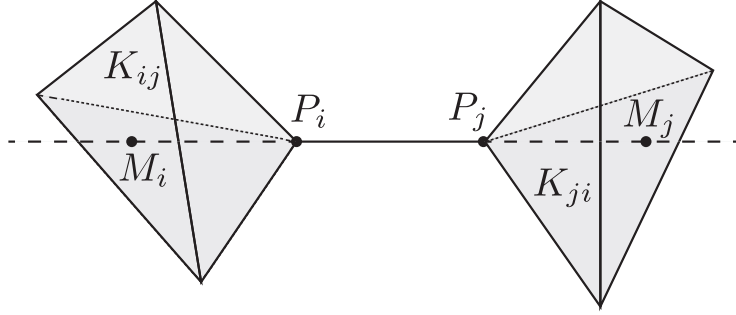


Fig. 2 Downwind K_{ij} and upwind K_{ji} tetrahedra associated with edge $P_i P_j$.

introducing the β -scheme:

$$\begin{aligned} (\nabla W)_i \cdot \overrightarrow{P_i P_j} &= (1 - \beta)(\nabla W)_{ij}^C \cdot \overrightarrow{P_i P_j} + \beta(\nabla W)_{ij}^U \cdot \overrightarrow{P_i P_j} \\ (\nabla W)_j \cdot \overrightarrow{P_j P_i} &= (1 - \beta)(\nabla W)_{ij}^C \cdot \overrightarrow{P_j P_i} + \beta(\nabla W)_{ij}^D \cdot \overrightarrow{P_j P_i}, \end{aligned}$$

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 $\beta = 1$.

Fourth-order numerical dissipation: 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:

$$\begin{aligned} (\nabla W)_i^{V4} \cdot \overrightarrow{P_i P_j} &= \frac{2}{3}(\nabla W)_{ij}^C \cdot \overrightarrow{P_i P_j} + \frac{1}{3}(\nabla W)_{ij}^U \cdot \overrightarrow{P_i P_j} \\ (\nabla W)_j^{V4} \cdot \overrightarrow{P_j P_i} &= \frac{2}{3}(\nabla W)_{ij}^C \cdot \overrightarrow{P_j P_i} + \frac{1}{3}(\nabla W)_{ij}^D \cdot \overrightarrow{P_j P_i}. \end{aligned}$$

Sixth-order numerical dissipation: V6-scheme. The V4-scheme is inspired from a one dimensional four points stencil and it can theoretically be extended to higher precision. The centered, downwind and upwind gradients used by the V4-scheme are completed with the nodal gradients of each nodes computed as the sum of the gradients of the neighboring elements. This yields a second-order scheme with a sixth-order numerical dissipation on unstructured grids.

We compare in the following the different levels of accuracy that are obtained when using low-dissipative schemes as V3-V6 on the sequence of Tailored meshes. tailored meshes are the ideal case for V4 and V6 schemes as each edge points toward an other point, so that there is no loss in interpolation and configurations are ideals to compute nodal gradients. We can see how less dissipative the V6-scheme is compared to the V3-scheme. This is especially visible on the shocks in the signature on the inviscid case in Figure 3, where both schemes are used on a grid with 27M nodes.

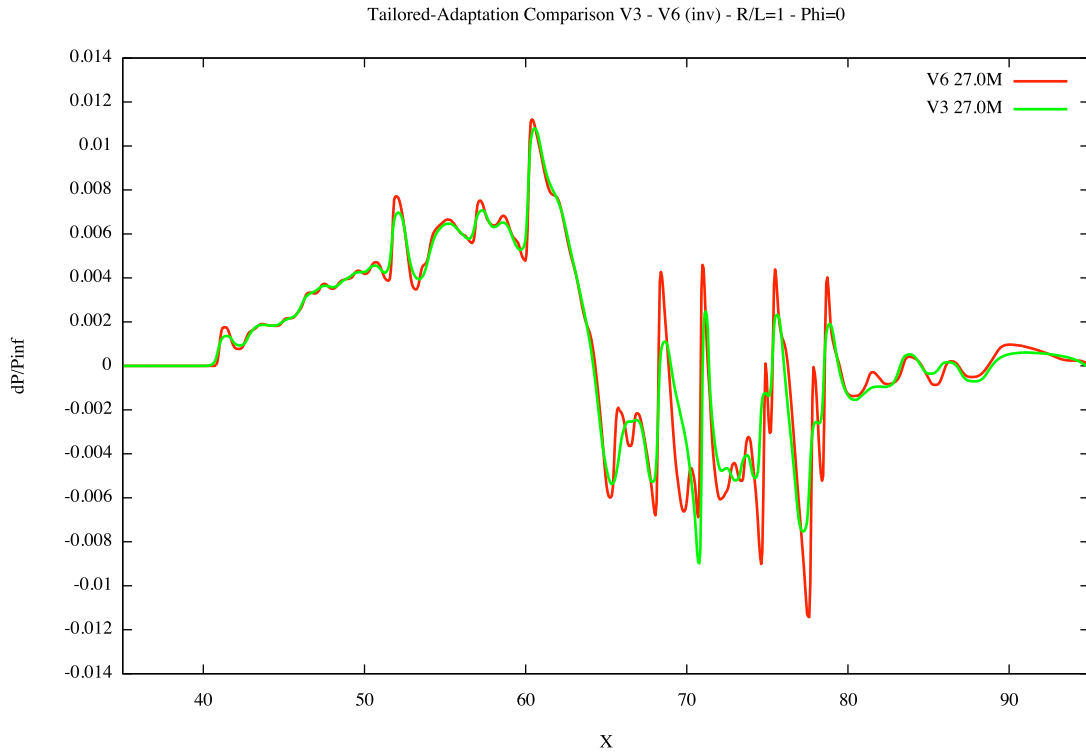


Fig. 3 Pressure signatures provided on the inviscid case by the V3-scheme and the V6-scheme on the same 27M nodes grid.

This low dissipation leads to an improvement in the propagation and capture of the shocks. We can see on Figures 4 that the V6-scheme already provides on a 6.3M nodes grid the same result as the V3-scheme on a 27M nodes grid. The viscous case shows a similar trend in Figures 5 and 6.

Figure 7 shows the same comparison on adapted grids. We can see on the fore part of the signature that the V6-scheme is slightly less dissipative. However, adapted grids are less favorable for V4 and V6-scheme as the intersection with the upwind/downwind elements generally happens on a face and thus require an interpolation. Similarly, the nodal gradients reconstructions are noisier on unstructured meshes reducing V6-scheme precision. Finally, mesh adaptation already drastically damps numerical dissipation in shocks and wakes, so that the influence of low dissipation schemes is less visible.

Limiters. The MUSCL schemes are not monotone and can be a source of spurious oscillations especially in the vicinity of discontinuities [17]. These oscillations can affect the accuracy of the final solution or simply end the computation because (for instance) of negative pressures. A widely used technique for addressing this issue is to guarantee the TVD property in 1D [18] or the LED property in 2D/3D of the scheme, which ensures that the extrapolated values W_{ij} and W_{ji} are not invalid. To guarantee the TVD or the LED properties, limiting functions are coupled with

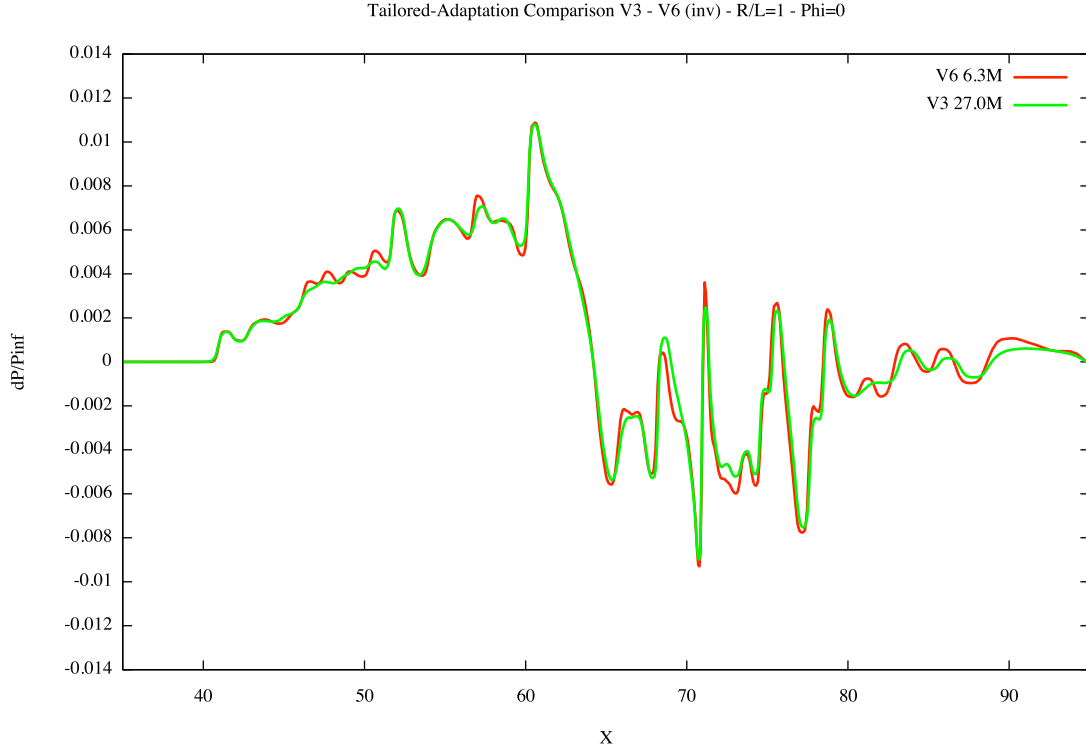


Fig. 4 Pressure signatures provided on the inviscid case by the V6-scheme on a 6.3M nodes grid and the V3-scheme on a 27M nodes grid.

the previous high-order gradient evaluations. The gradient is substituted by a limited gradient denoted $(\nabla W)_{ij}^{lim}$. The choice of the limiting function is crucial as it directly affects the convergence of the simulation. In this work, we use the Piperno's limiter which is expressed in a factorized form,

$$(\nabla W)^{Lim} = \nabla W^C \psi_{PI} \left(\frac{\nabla W^C}{\nabla W^{V4}} \right),$$

with

$$\psi_{PI}(R) = \left(\frac{1}{3} + \frac{2}{3}R \right) \begin{cases} \frac{3\frac{1}{R^2} - 6\frac{1}{R} + 19}{\frac{1}{R^3} - 3\frac{1}{R} + 18} & \text{if } R < 1 \\ 1 + \left(\frac{3}{2}\frac{1}{R} + 1 \right) \left(\frac{1}{R} - 1 \right)^3 & \text{if } R \geq 1 \end{cases}$$

where

$$R = \frac{\nabla W^C}{\nabla W^{V4}}.$$

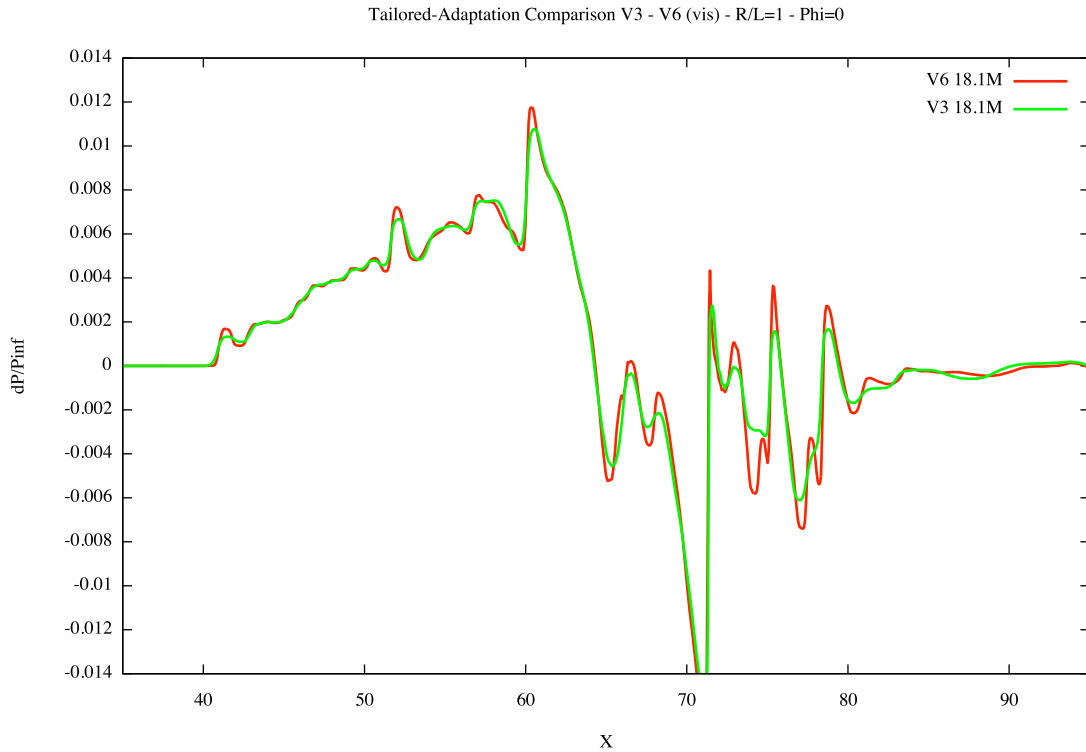


Fig. 5 Pressure signatures provided on the viscous case by the V3-scheme and the V6-scheme on the same 27M nodes grid.

D. Residuals and Limiter Freezing

In a mesh adaptation context, it is crucial to guarantee the convergence of the solver so that no time is wasted on intermediate grids and a reasonably good solution is used to derive the next mesh. A major drawback of limiters is that they can prevent the solver from converging by creating limit cycles so that the solution shows very little temporal evolution but the numerical residual does not converge. Such a situation is shown in Figure 8, we can see that the residual does not converge, although the solution is actually converged.

Freezing limiters. This behavior is mainly due to the interaction between the approximate Jacobian used in the implicit solver and small variations in the gradients which induce larger variations in the fluxes through the limiter used. An option consists in "freezing" the limiters either by keeping the gradients to a given value (its value at the moment of the freezing) or by updating the value of the gradient only if it is smaller than the previous value. In this paper, we use the second option. Note that the usual limiters tend to a first order method if gradients are set to zero so that no extrapolation is done. In Figure 8, we show the convergence history of residual with different maximal CFLs, growing rates of CFLs and different times of limiter freezing. With no surprise, a higher CFL leads to a faster "convergence" of the solution. We can see how effective limiter freezing is in terms of residual convergence.

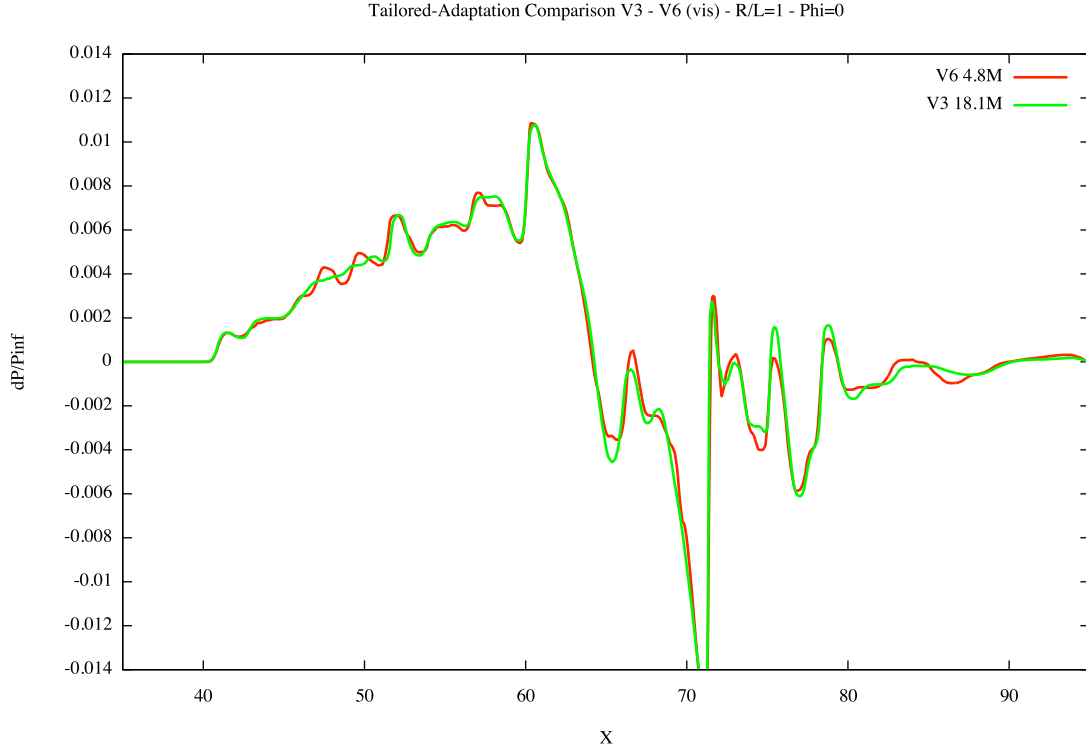


Fig. 6 Pressure signatures provided on the viscous case by the V6-scheme on a 6.3M nodes grid and the V3-scheme on a 27M nodes grid.

Statistical residual. It is here obvious that freezing limiter must be done at a proper moment, especially not too soon if we don't want to deteriorate the solution. In particular we can doubt of the pertinence of freezing limiter at the 250th iteration for the fourth case (purple curve) in Figure 8. It turns out this noise due to limiters is often concentrated in a few elements which dominates the residual. Representing the nodal residual field with a single scalar

$$Res = \sum res_i,$$

leads to a huge loss of information. In particular, we can see that a single point in the far field with a high nodal residual can block the whole residual even though other more relevant points have a lower residual. This is why we replaced the scalar residual by a statistical distribution by counting the number of nodes n_j with a residual in between 10^{-j} and $10^{-(j+1)}$ and computing the volumes v_j as the sum of the dual volumes of nodes with a residual in between 10^{-j} and $10^{-(j+1)}$. The classical residual can still be approximately computed as

$$Res = \sum_j n_j \times 10^{-j},$$

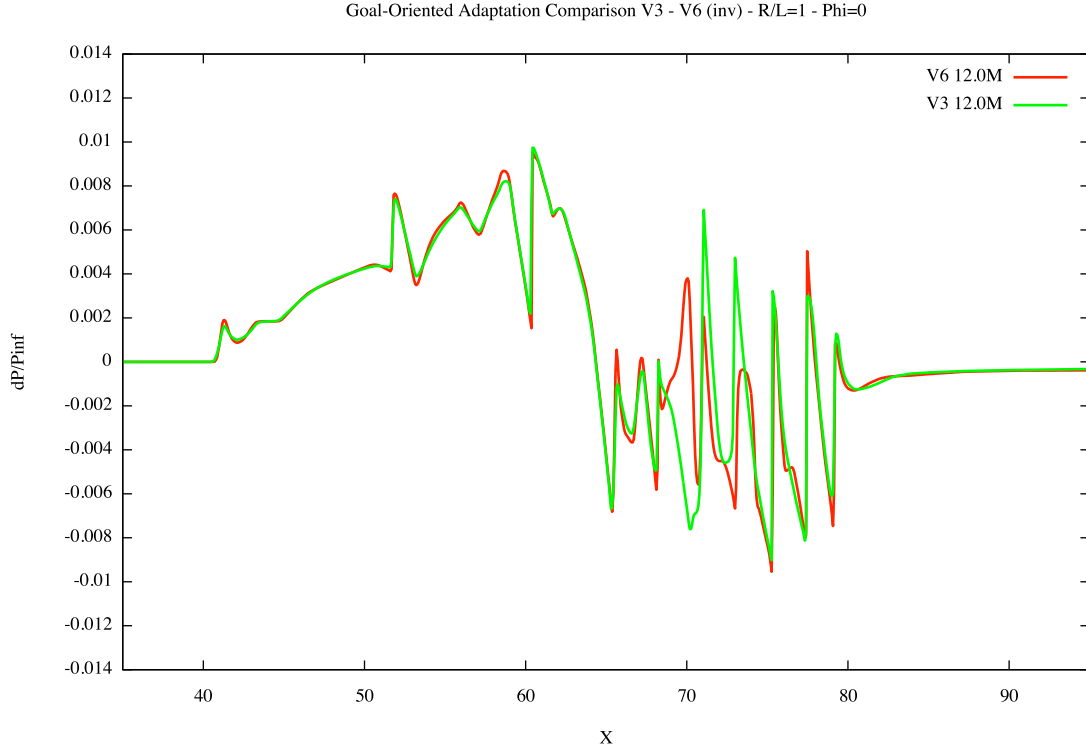


Fig. 7 Comparison of pressure signatures provided on the inviscid case by the V3-scheme and the V6-scheme on the same adapted 12M nodes grid

but this simple distribution also gives useful indications on the convergence. We can thus determine whether all the nodes are still not converged or a few points are not converging. We can also discriminate the nodes associated with large dual volumes and thus of lower interest for the solution. This allows to derive a statistical residual indicator as

$$Res^{log} = 10^{-(\sum j \times n_j) / \sum n_j}.$$

Used together, these two residuals give a better representation of the convergence of the solution. We can see in Figure 9 that the statistical residual initially converges below 10^{-2} and stagnates while the classical residual increases and stagnates indicating that most of the nodes are converging while a few are blocking the residual. Hence, this can be used as a proper sensor to freeze the limiter.

E. Boundary Conditions

1. Supersonic/Subsonic Inlet

Subsonic and supersonic inflow boundary conditions are used to prescribe a consistent physical incoming flow with a given total pressure, total temperature and flow direction [19].

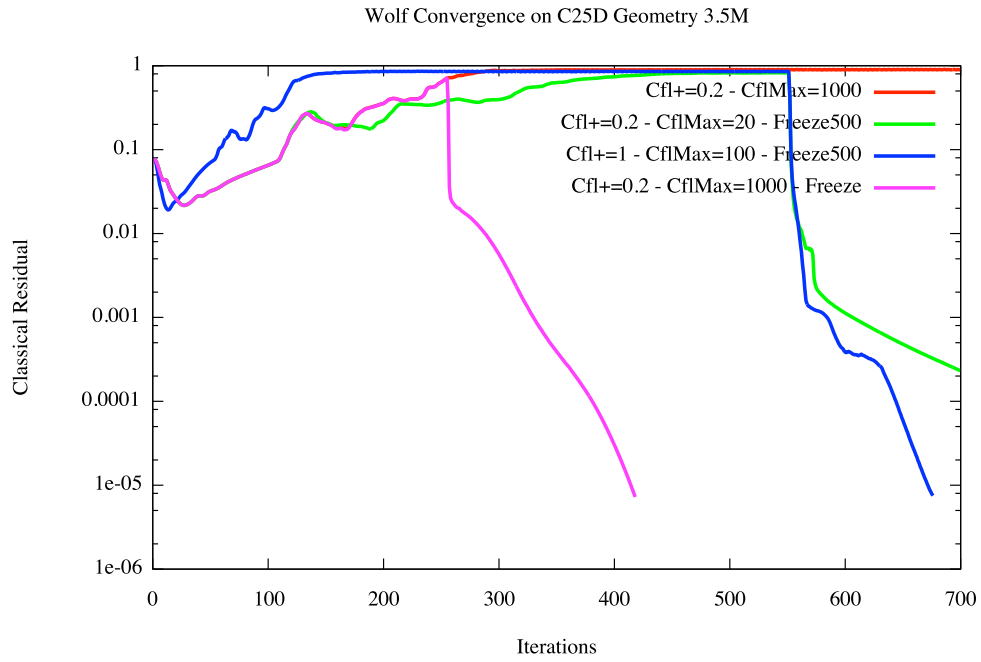


Fig. 8 Convergence history when a standard residual is used, with and without freezing the limiters.

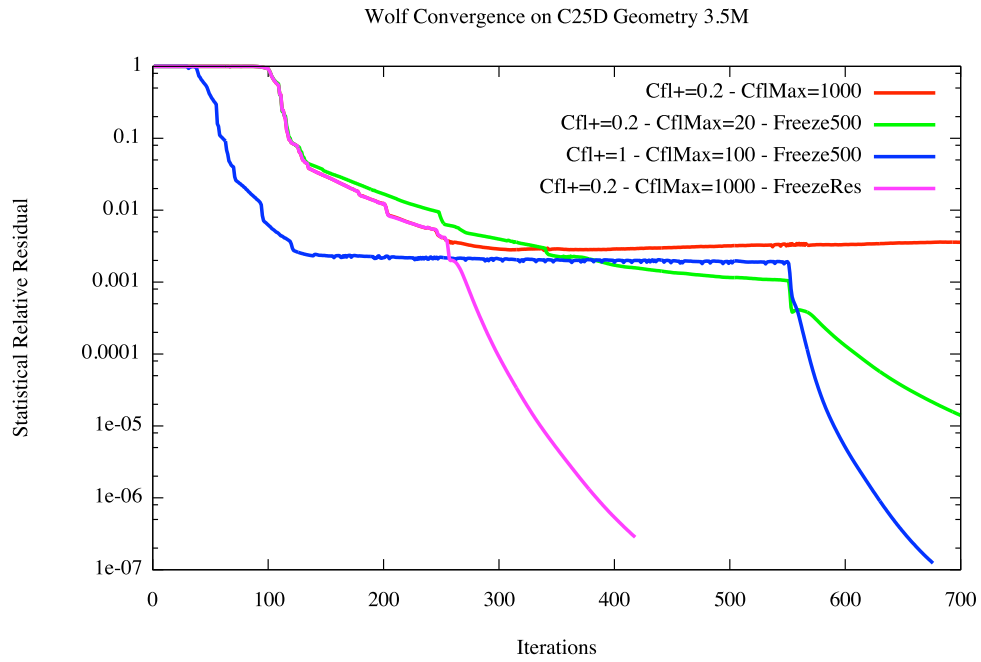


Fig. 9 Convergence history when a statistical residual is used to recording convergence history.

Flux prescription. In order to provide a numerically and physically consistent boundary condition, we rely on Riemann invariants across the boundary surface to compute an appropriate external state. The boundary flux is then computed with the same approximate Riemann solver (HLLC, Roe, ...) as inside the domain, using the external and the inner state.

In the supersonic case, all eigen values of the Riemann problem are positive, meaning that no information comes from the inside domain (sound waves cannot go upstream). A numerical consistent boundary condition can be provided by prescribing the full desired state outside which does not depend on the inner state

$$W_{ext} = (\rho_{ext}, u_{ext}, p_{ext}).$$

In the subsonic case, sound wave travels upstream and imposing a given state as in supersonic case leads to wave reflexions and instabilities that slow the convergence. The negative Riemann invariant is thus used to compute a consistent external state. Given the inner state $(\rho, \rho u, \rho v, \rho w, \rho e)$ and the normal of the face, we can compute the normal velocity

$$u_n = \mathbf{u} \cdot \mathbf{n},$$

the boundary sound velocity

$$c = \sqrt{\gamma p / \rho},$$

and the outgoing characteristic that must be conserved

$$R^- = u_n - \frac{2c}{\gamma - 1},$$

and the ingoing enthalpy, defined by our desired state

$$H_{tot} = \frac{\gamma}{\gamma - 1} RT_{tot}.$$

These quantities are related to external state by:

$$H_{tot} = \frac{c_{ext}}{\gamma - 1} + \frac{u_{ext}^2}{2} \quad \text{and} \quad R^- = u_{ext} - \frac{2c_{ext}}{\gamma - 1},$$

which allows us to compute either u_{ext} or c_{ext} as solution of a quadratic equation and deduce the other one from R^- value. The physical root is the largest one. External pressure and temperature are then deduced from

$$p_{ext} = P_{tot} \left(1 + \frac{\gamma - 1}{2} \frac{u_{ext}^2}{c_{ext}^2} \right), \quad T_{ext} = T_{tot} \left(\frac{p_{ext}}{P_{tot}} \right)^{\frac{\gamma - 1}{\gamma}},$$

and finally, external state reads

$$W_{ext} = \left(\frac{p_{ext}}{RT_{ext}}, u_{ext} \mathbf{n}, p_{ext} \right).$$

2. Supersonic/Subsonic Outlet

Subsonic and supersonic outflow boundary conditions are used to prescribe a consistent physical outgoing flow with a given pressure. As for inflow boundary conditions, we rely on Riemann invariants.

Flux prescription. In the supersonic case, no information should come from outside the domain, so the external state is chosen equal to the inner state. Thus if the outgoing flow is supersonic, no outflow pressure is effectively imposed. In the subsonic case, outflow pressure is imposed by upwind going sound waves. Downstream traveling Riemann invariant

$$R^+ = \frac{2c}{\gamma - 1} + \mathbf{u} \cdot \mathbf{n},$$

and entropy

$$s = \frac{p}{\rho^\gamma},$$

are computed from inside the domain and assumed to be constant. As we want to impose a given pressure, we can deduce

$$\rho_{ext} = \left(\frac{p_{ext}}{s} \right)^{\frac{1}{\gamma}}, \quad c_{ext} = \sqrt{\gamma p_{ext} / \rho_{ext}},$$

and the normal velocity from the Riemann invariant

$$u_{ext} = R^+ - \frac{2c_{ext}}{\gamma - 1}.$$

As tangential velocity is advected by the flow and assumed constant, external state is defined as

$$W_{ext} = (\rho_{ext}, u_{ext} \mathbf{n} + \mathbf{u}_t, p_{ext}),$$

with

$$\mathbf{u}_t = \mathbf{u} - (\mathbf{u} \cdot \mathbf{n}) \mathbf{n}.$$

3. No-slip Boundary Condition

For no-slip boundary conditions, $\mathbf{u} = 0$ and $v_t = 0$ are strongly enforced at each iteration. Consistently, we impose $\Phi_\rho = \Phi_{\mathbf{u}} = 0$ at the boundary. The energy flux is fixed according to the desired temperature behavior: for an adiabatic wall it is null and for an isothermal wall the energy variable is enforced similarly to the velocity.

III. Mesh adaptation algorithms and metric-based error estimates

We describe in this section, the local mesh modification software `Feflo.a`. It is based on a unique cavity-based operator to adapt the mesh with respect to a provided metric-field. We then describe how to derive metric-based error estimates to drive the adaptive process. We consider two error estimates : a multi-scale (feature-based) one that focuses on a particular field of the flow and a goal-oriented one where the approximation error of a scalar-output functional is reduced. Finally, we describe a non-linear corrector for CFD solutions that is used to estimate point-wise error estimates on the flow. This allows us to assess the convergence of the adaptive process.

Metric-based mesh adaptation. `Feflo.a` is a generic purpose adaptive mesh generator dealing with 2D, 3D and surface mesh generation. It belongs to the class of metric-based mesh generator [20–24] which aims at generating a unit mesh with respect to a prescribed metric field \mathcal{M} . A mesh is said to be unit when composed of almost unit-length edges and unit-volume element. The length of an edge AB in \mathcal{M} is evaluated with:

$$\ell_{\mathcal{M}}(AB) = \int_0^1 \sqrt{tAB \mathcal{M}((1-t)A + tB) AB} dt,$$

while the volume is given by $|K|_{\mathcal{M}} = \sqrt{\det \mathcal{M}}|K|$, where $|K|$ is the Euclidean volume of K . From a practical point of view, the volume and length requirements are combined into a quality function defined by :

$$Q_{\mathcal{M}}(K) = \frac{36 \sum_{i=1}^6 \ell_{\mathcal{M}}^2(\mathbf{e}_i)}{3^{\frac{1}{3}} |K|_{\mathcal{M}}^{\frac{2}{3}}} \in [1, \infty],$$

where $\{\mathbf{e}_i\}_{i=1,6}$ are the edges of element K . A perfect element has a quality of 1.

Cavity-based operators. A complete mesh generation or mesh adaptation process usually requires a large number of operators: Delaunay insertion, edge-face-element point insertion, edge collapse, point smoothing, face/edge swaps, etc. Independently of the complexity of the geometry, the more operators are involved in a remeshing process, the less robust the process may become. Consequently, the multiplication of operators implies additional difficulties in maintaining, improving and parallelizing a code. In [25], a unique cavity-based operator has been introduced which embeds all the aforementioned operators. This unique operator is used at each step of the process for surface and volume remeshing.

The cavity-based operator is inspired from incremental Delaunay methods [26–28] where the current mesh \mathcal{H}_k is modified iteratively through sequences of point insertion. The insertion of a point P can be written:

$$\mathcal{H}_{k+1} = \mathcal{H}_k - C_P + \mathcal{B}_P, \tag{7}$$

where, for the Delaunay insertion, the cavity C_P is the set of elements of \mathcal{H}_k such that P is contained in their circumsphere

and \mathcal{B}_P is the ball of P , i.e., the set of new elements having P as vertex. These elements are created by connecting P to the set of the boundary faces of C_P .

In [25], each meshing operator is equivalent to a node (re)insertion inside a cavity. For each operator, we just have to define judiciously which node P to (re)insert and which set of volume and surface elements will form the cavity C where point P will be reconnected:

$$\mathcal{H}_{k+1} = \mathcal{H}_k - C + \mathcal{R}_P. \quad (8)$$

Note that if \mathcal{H}_k is a valid mesh (only composed of elements of positive volume) then \mathcal{H}_{k+1} will be valid if and only if C is connected (through internal faces of tetrahedron) and \mathcal{R}_P generates only valid elements.

The use of the previous cavity-based operators allows us to design a remeshing algorithm that has a linear complexity in time with respect to the required work (sum of the number of collapses and insertions). On a typical laptop computer Intel Core I7 at 2.7 GHz, the speed for the (cavity-based) collapse is around 20 000 points removed per second and the speed for the insertion is also around 20 000 points or equivalently 120 000 elements inserted per second. Both estimates hold in an anisotropic context [29].

The local mesh adaptation strategy is then used on several metric-based estimates.

Hessian-based mesh adaptation. A first set of error estimate is based on the minimization of the interpolation error of one or several sensors depending on the CFD solution [20, 30–35]. Given a numerical solution W_h , a solution of higher regularity $R_h(W_h)$ is recovered, so that the following interpolation error estimate [36, 37] holds:

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

where $H_{R_h(W_h)}$ is the Hessian of the recovered solution and N an estimate of the desired number of nodes. We have also the optimal metric field leading to the previous error estimate:

$$\mathcal{M}_{L^p} = \mathcal{K}_{L^1}(|H_{R_h(W_h)}|) = D_{L^p}(\det|H_{R_h(W_h)}|)^{\frac{-1}{2p+3}} |H_{R_h(W_h)}|.$$

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 [38]. Indeed, if $R_h(W_h)$ is a better approximate of W in the following meaning:

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

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, L^2 -projection, green formula or the Zienkiewicz-Zhu recovery operator. For all the numerical examples, the L^2 norm is used with the L^2 -projection to recover derivatives.

Goal-oriented mesh adaptation. A second set of error estimate tends to couple adaptivity with the assessment of the numerical prediction of the flow. Goal-oriented optimal methods [1, 39–42] aim at minimizing the error committed on the evaluation of a scalar functional. A standard functional for sonic boom study is the observation of the pressure field on an observation surface γ :

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

where W and W_h are the solution and the numerical solution of the compressible Euler equations, respectively. They do take into account the features of the PDE, through the use of an adjoint state that gives the sensitivity of W to the observed functional j . In order to solve the goal-oriented mesh optimization problem, an *a priori* analysis has been introduced [43, 44] which restricts to the main asymptotic term of the local error. In the case of inviscid flows, an optimal goal-oriented metric is defined as:

$$\mathcal{M}_{go}^{opt}(W_h) = \mathcal{K}_{L^1}(|\nabla W_{go,h}^*| \cdot |H_{R_h(F_h(W_h))}|),$$

where $W_{go,h}^*$ is the adjoint solution with second member being $g_{go,h} = \frac{\partial j}{\partial W}(W_h)$, and $R_h(F_h(W_h))$ the recovered Hessian of the continuous Euler fluxes. If a super-convergence of $|j(W) - j_h(W_h)|$ may be observed in some cases [45, 46], 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 (by means of multiple adjoint states), the optimality of the mesh and the convergence properties of the

approximation error may be lost.

Non-linear corrections. In each case, the aforementioned adaptive strategies address specifically one goal. In particular the implicit error $W_h - \Pi_h W$ is neither controlled nor estimated. This error can be interpreted as a point-wise error that estimates the point-wise error between the exact solution and the solution provided by the numerical scheme. We intend to estimate it in this paper, in order to see that this error is also reduced during the refinement process. To do so, we estimate the defect, δ , of the continuous solution when applied to the numerical solution. In the framework of the Euler equations, we have

$$\operatorname{div}(F(W_h)) = \delta \approx 0.$$

To estimate δ , the continuous flux function F is evaluated on a finer mesh. Then we compute $F_{h/2}(\Pi_{h/2}W_h)$ which is equivalent to one solver flux evaluation. This flux is accumulated back on the current mesh, defining a source term:

$$S_{post} = A_{h/2 \rightarrow h}(F_{h/2}(\Pi_{h/2}W_h)),$$

The corrected solution is then solution of :

$$\operatorname{div}(F_h(W_c)) = S_{post},$$

starting with $W_c = W_h$. Note that the finer is never generated from a practical point of view, only a local subdivision is used. In addition, the computation of S_{post} is naturally highly parallel.

IV. numerical results and discussions

In this section, we focus on adaptive computations with the multi-scale and goal-oriented error estimates for the inviscid cases. We compare the results obtained with these strategies with the sequence of tailored meshes aligned on the Mach cones. For all the simulations the same computational domain is used and is similar to the one provided on fixed grids, see Figure 1.

Multi-scale. For the multi-scale approach, we control the interpolation error on the Mach number in \mathbf{L}^2 norm. A sequence of 16 meshes are generated. 4 meshes at fixed complexity N are generated, and the prescribed complexity is multiplied by 2 for the 4 next iterations. The sizes of the meshes, ratios and anisotropic quotients are reported in Table 1. The final mesh is composed of more than 27 million vertices. The total number of iterations for the flow solver is 10 176. Note that the adaptation is performed on the full computational domain. Consequently, much less accurate pressure signatures are computed at different R/L , see Figures 10, 11 and 11. However, the accuracy is independent of the incident angle. It appears also that reflections of the shocks waves appear at the bottom of the far field. This prevents us from adding additional step for the multi-scale approach.

# vertices	# tetrahedra	mean ratio	mean quotient
1 946 918	11 093 526	10	97
5 529 916	32 726 736	18	297
12 953 271	77 497 813	18	325
27 644 454	166 148 147	19	400

Table 1 Statistics for the sequence of multi-scale meshes.

# vertices	# tetrahedra	mean ratio	mean quotient
1 555 516	8 680 173	8	61
3 112 225	17 894 815	14	197
5 915 408	34 553 239	17	2213
11 736 061	68 985 939	46	5713

Table 2 Statistics for the sequence of goal-oriented meshes.

Goal-oriented. For the adjoint case, the function of interest is the pressure integral on the symmetry plane with $z < 0$. 16 meshes are generated in this case, with the same strategy of performing 4 iterations at fixed complexity. The statistics of the generated meshes are reported in Table 2. The convergence of the pressure signatures is reported in Figures 10, 11 and 12. The convergence of the functional is given in Figure 13. In comparison with the multi-scale approach, the mean anisotropic ratio and quotient are increasing at each level of refinement. For this case, a total of 13 076 solver iterations are necessary.

Comparisons. To compare the level of accuracy of the three approaches : tailored (provided), multi-scale and adjoint meshes, we compare the pressure signatures at iso-degrees of freedom for $R/L = 1, 2, 3$ on Figure 14. The multi-scale is clearly the less accurate as the mesh is adapted in the whole computational domain. The adjoint and tailored provide a similar pressure signature for the front signal while very large discrepancies arise on the second part of the signature. Similar observations hold when comparing the best effort meshes in Figure 15. The complexity of the flow pattern is depicted in Figures 16 and 17. To understand the differences on the back of the signals, we compare the pressure signature on the cylinder of radius 5 which corresponds to the limit of the uniform mesh for tailored meshes. We observe that a lot of shock waves are already dissipated before being propagated in the quasi-structured mesh, see Figure 18. Also, it appears that the surface mesh is highly under-resolved as a complex pattern of shocks and reflections are present in the vicinity of the aircraft, see Figure 19.

Uncertainty estimates. In order to show the impact of adaptivity, we perform an uncertainty analysis on the pressure distribution at $R/L = 1, 3, 5$. Corrected solutions are computed on the tailored baseline meshes and on the sequence of goal-oriented adapted meshes. We show on Figure 20 that the estimated point-wise error is reduced at each step

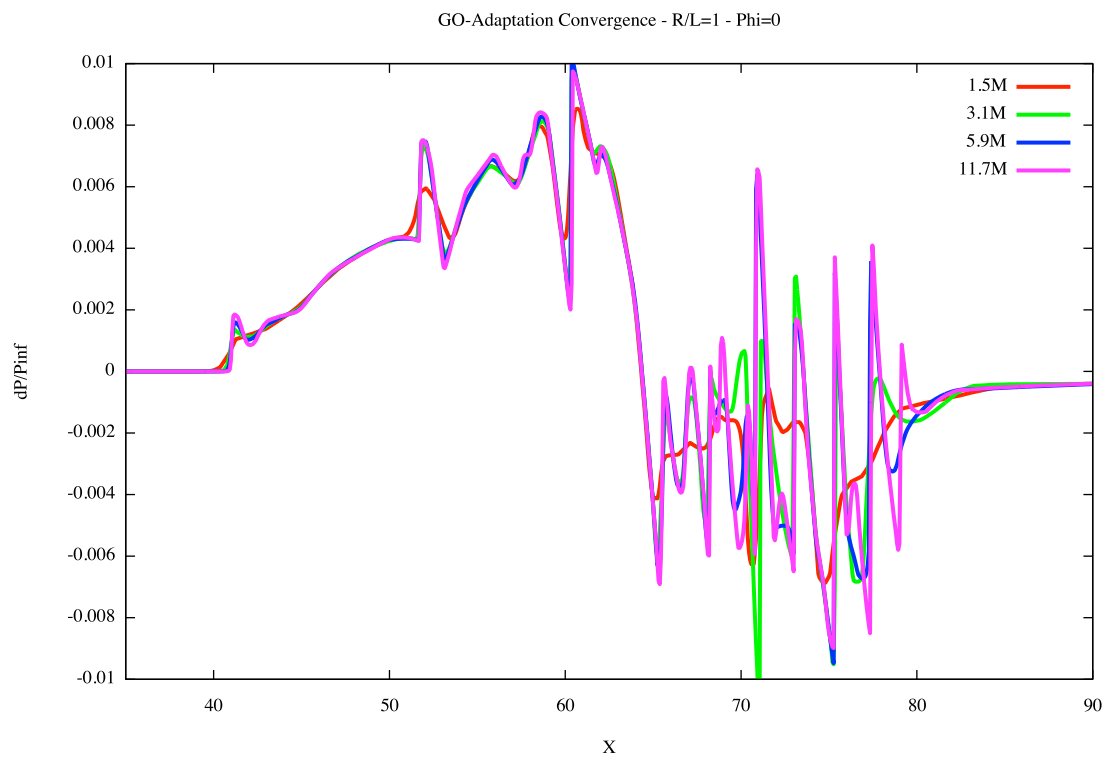
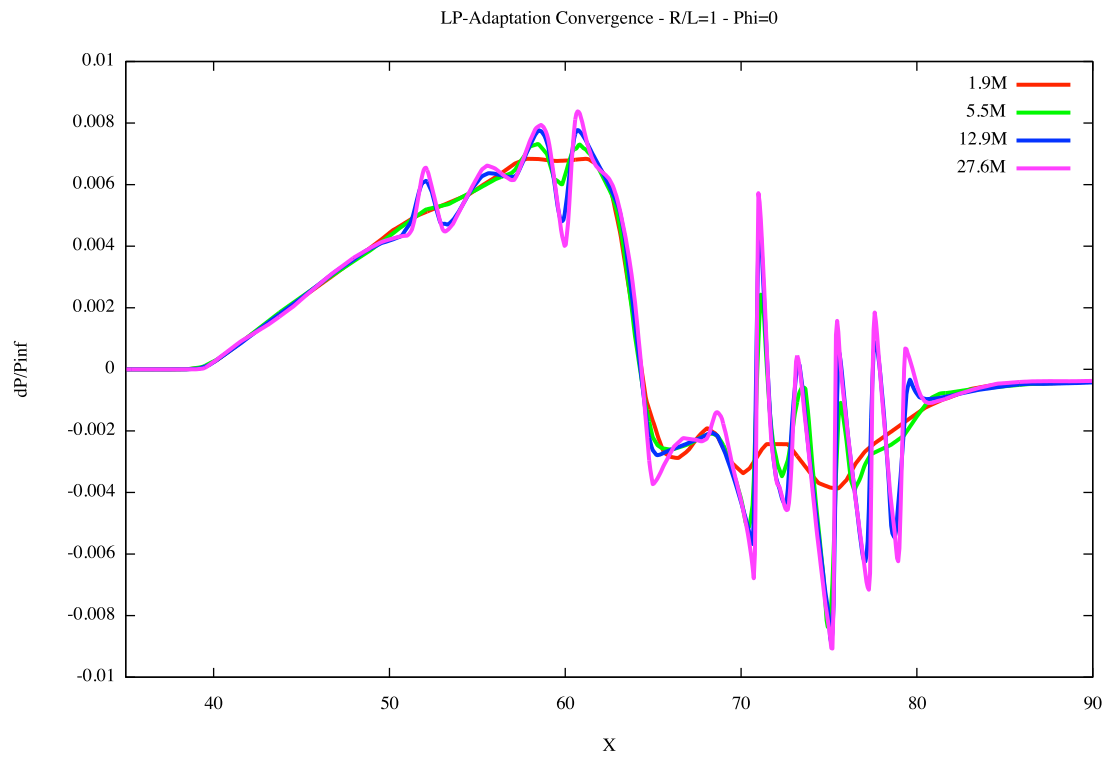


Fig. 10 Pressure signature convergence at $R/L = 1$ with multi-scale mesh adaptation (top) and goal-oriented mesh adaptation (bottom).

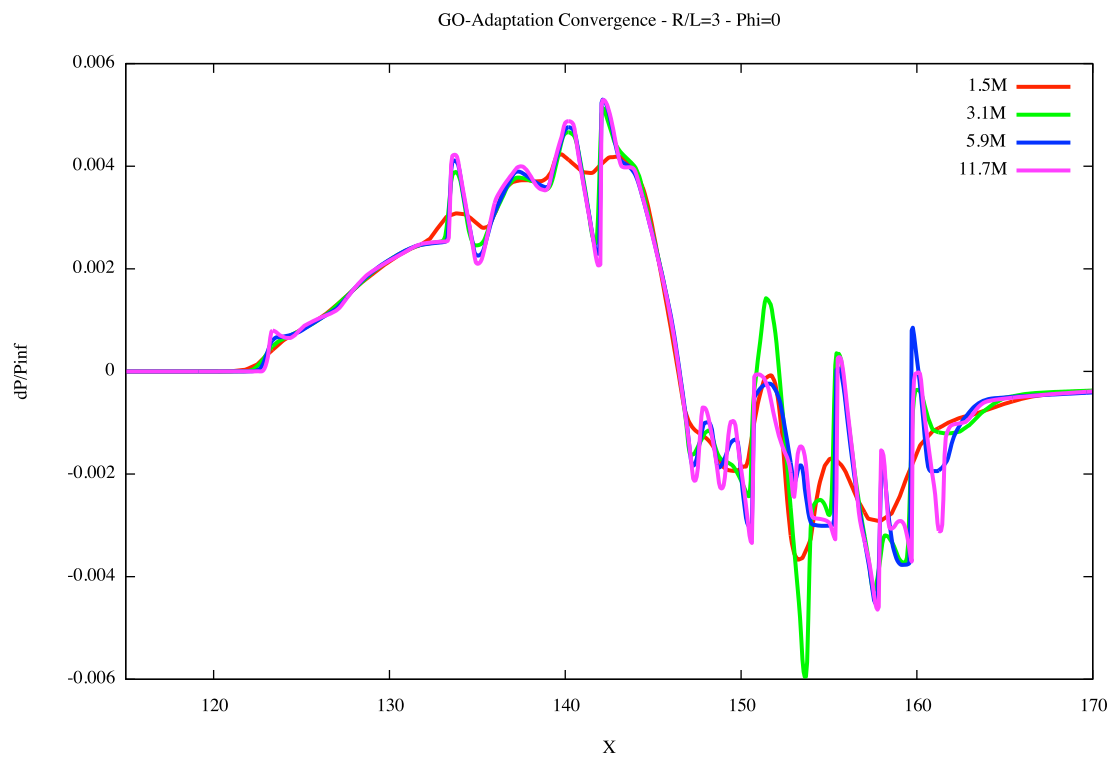
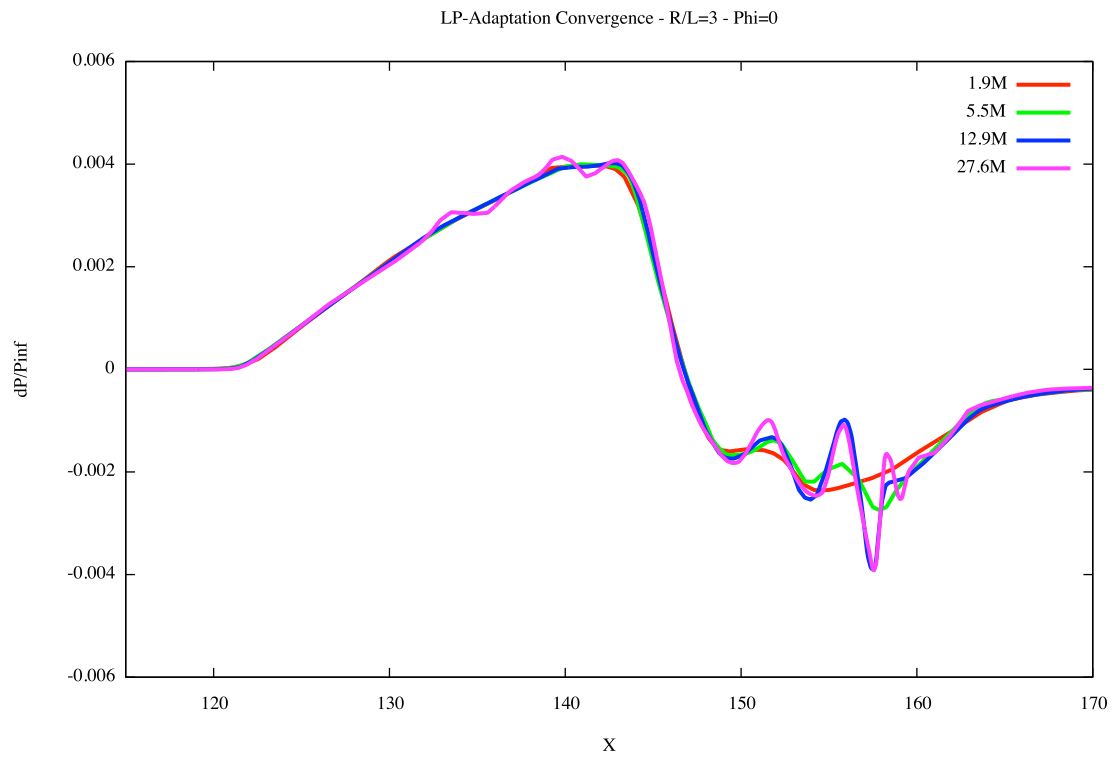


Fig. 11 Pressure signature convergence at $R/L = 3$ with multi-scale mesh adaptation (top) and goal-oriented mesh adaptation (bottom).

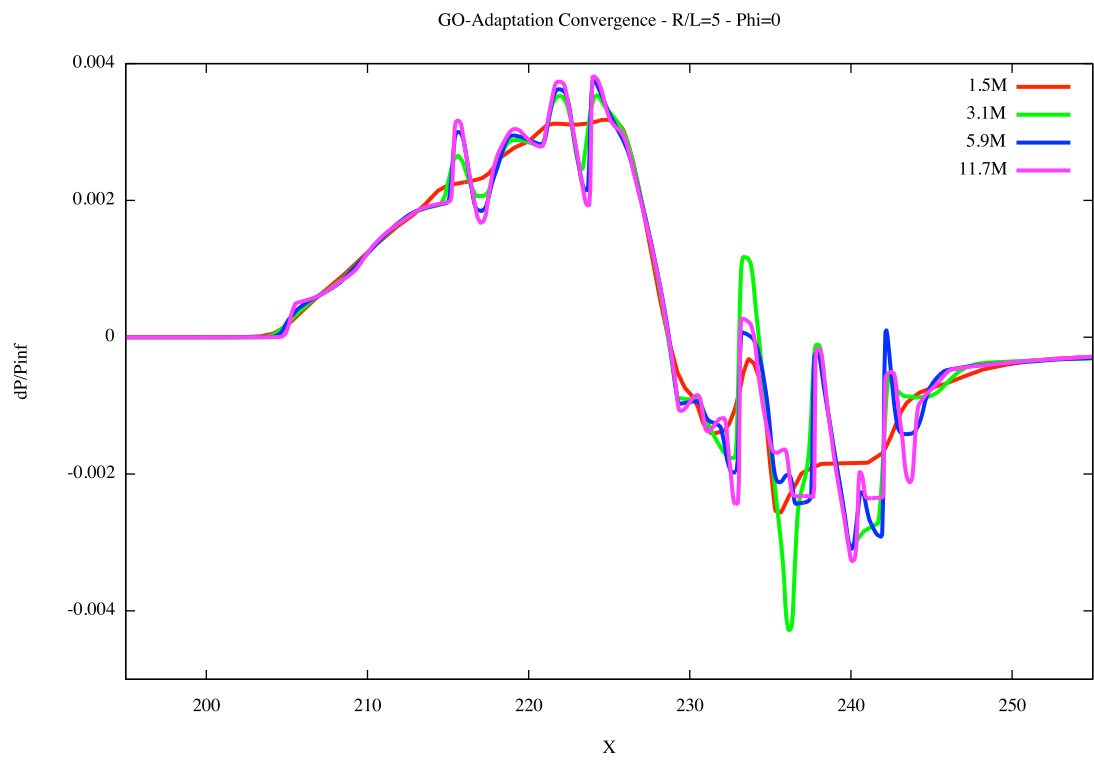
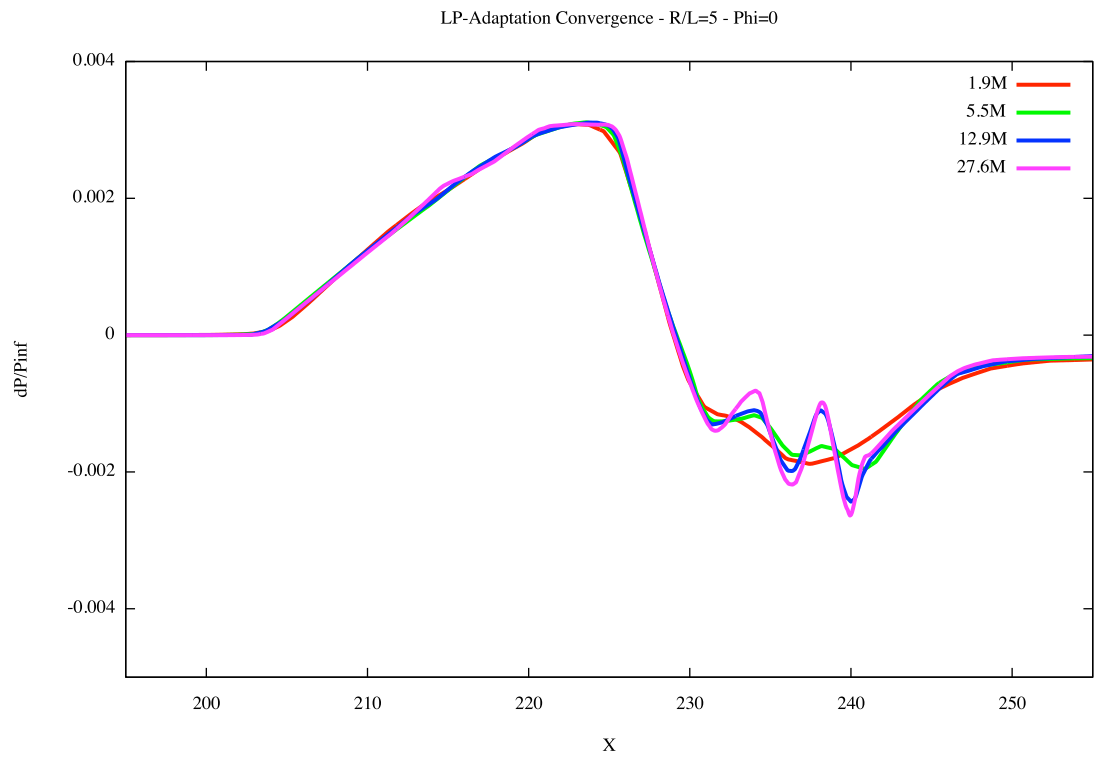


Fig. 12 Pressure signature convergence at $R/L = 5$ with multi-scale mesh adaptation (top) and goal-oriented mesh adaptation (bottom).

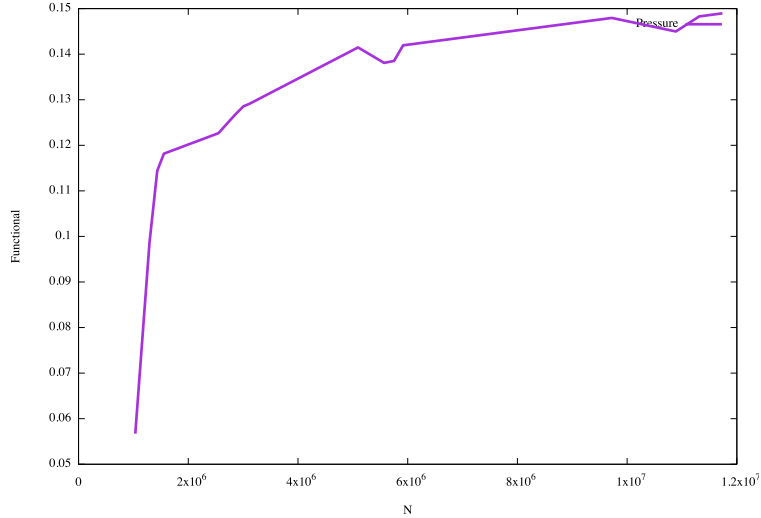


Fig. 13 Convergence of the functional of interest.

for the adaptive computations while this is not true when using tailored meshes. In the later case, the implicit error is not reduced at each level of pre-adapted grids. This tends to prove that automatic mesh adaptation mesh provides a more consistent sequence of *embedded* grids that grids build on *a priori* knowledge. Indeed, for the tailored grids, the dissipation is controlled only in one direction (across the Mach cone) whereas the dissipation is controlled in every direction for adaptive methods.

Conclusion

In this analysis, we have studied the spatial mesh convergence for the multi-scale L^2 and the goal-oriented approaches on the C25D geometry of the 2nd AIAA sonic boom workshop. The CFD computations are based on a low dissipation second order flow solver. Especially for low boom geometry, reaching asymptotic level of convergence is mandatory to assess the predicted pressure signature below the aircraft. When the full domain is adapted (as in the tailored approach or multi-scale), this requires large meshes with degrees of freedom of the order of $10^7 - 10^9$. In this respect, goal-oriented methods can substantially reduced the size of the mesh to reach the asymptotic rates. The difficulty in this case is to design proper functional to observe. In the case of integrated functional, the integration may smooth the phenomena so that small variation in the pressure may have very small impact on the observed outputs.

The non linear corrector used in this paper allows us to provide additional insight on the obtained numerical solutions by providing upper and lower bounds of every quantity of interest. We show that the corrected solutions are minimized along the adaptive iterations which is not the case when applied to tailored meshes.

For this particular geometry, providing highly resolved signatures is still a challenge due to the strong shocks patterns and their interactions with the wake of the engine. It results that large variations in the pressure signatures even at small

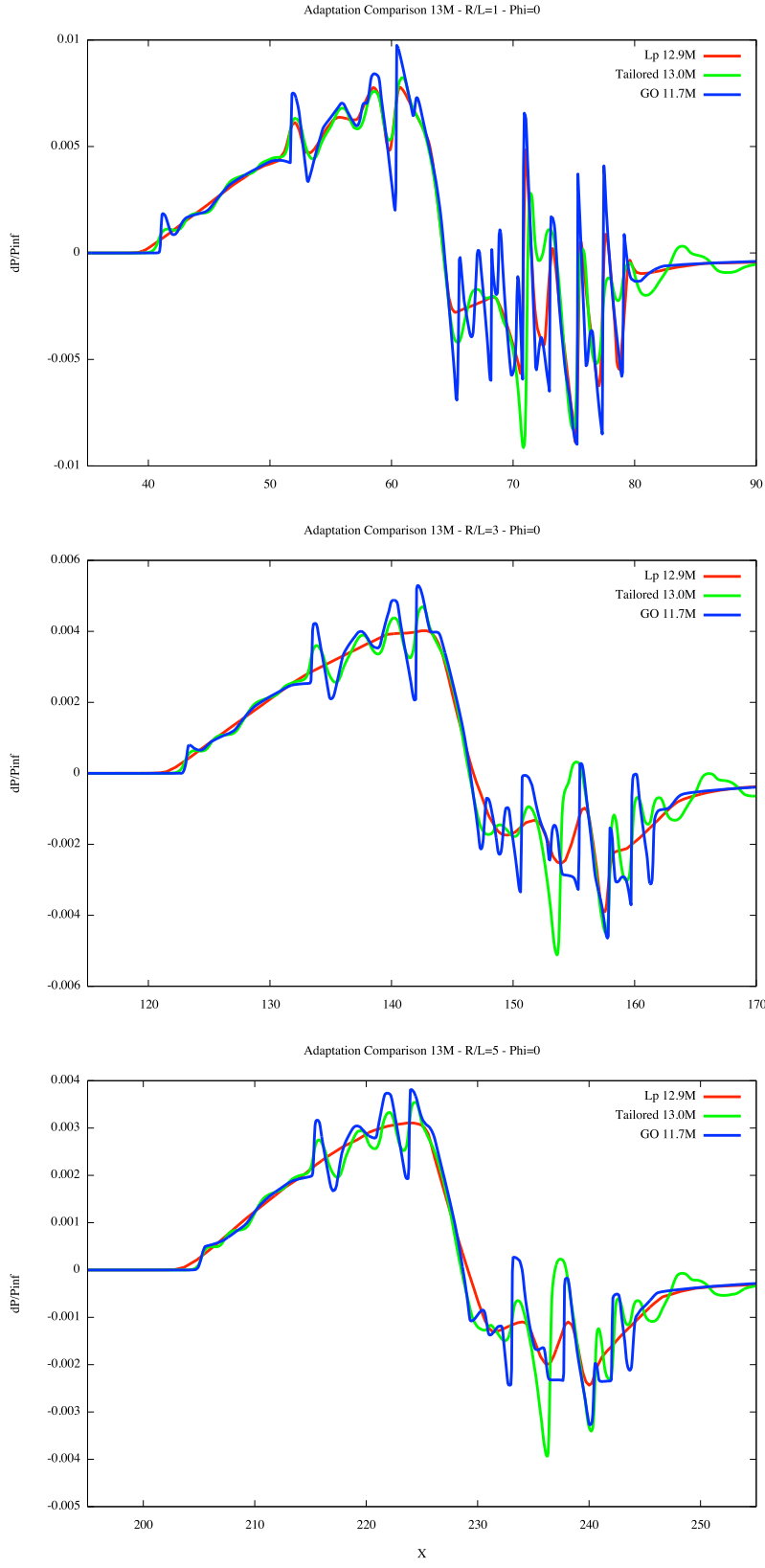


Fig. 14 Iso-dof pressure signatures for $R/L = 1, 2, 3$ comparing tailored, L^2 and adjoint-based meshes.

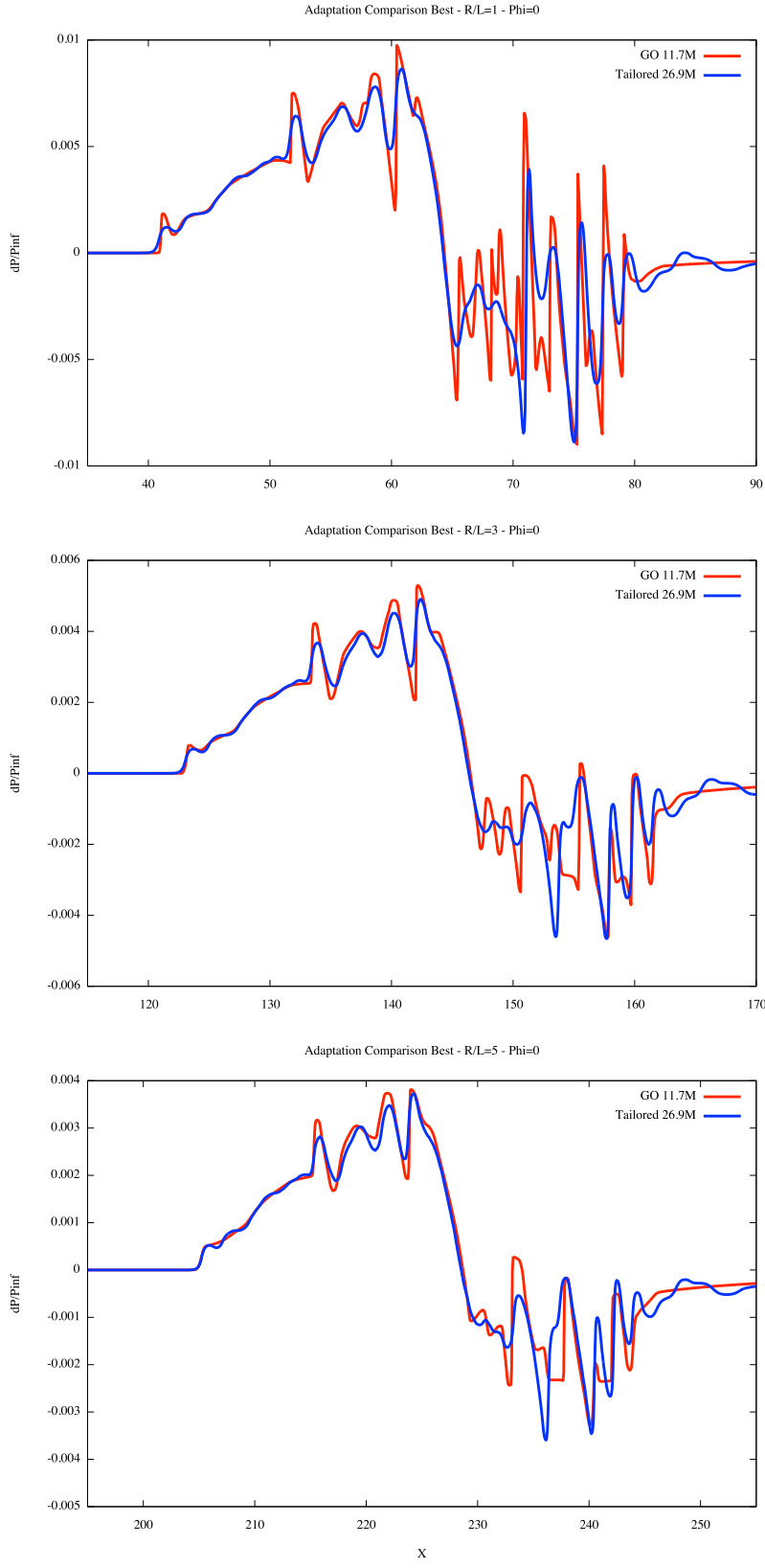


Fig. 15 Best effort pressure signatures for $R/L = 1, 2, 3$ comparing tailored, L^2 and adjoint-based meshes.

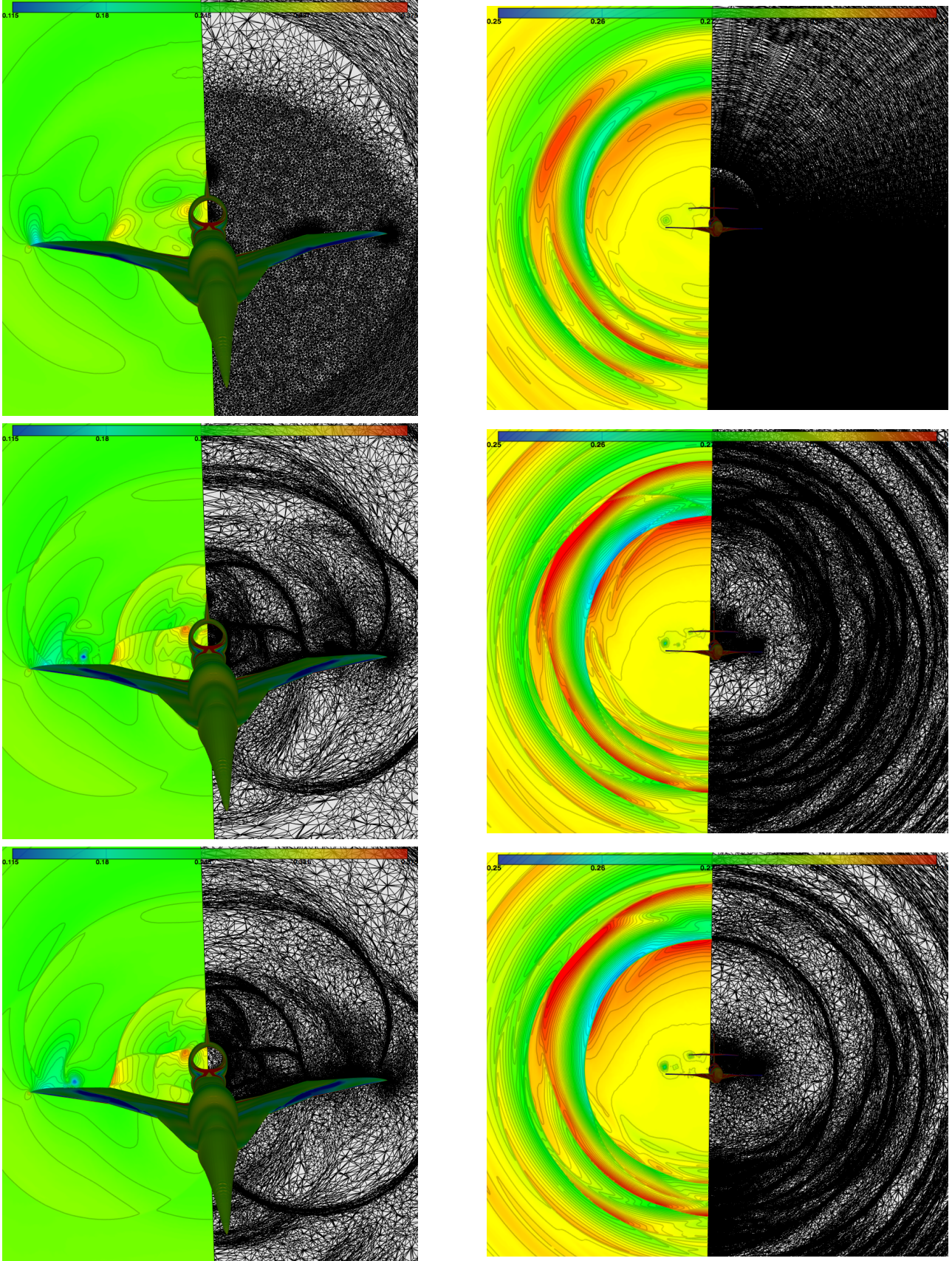


Fig. 16 Comparisons of the best effort meshes at cut plane $x = 30$ (left) and $x = 50$ (right) for the tailored mesh (top), multi-scale (middle) and goal-oriented (bottom).

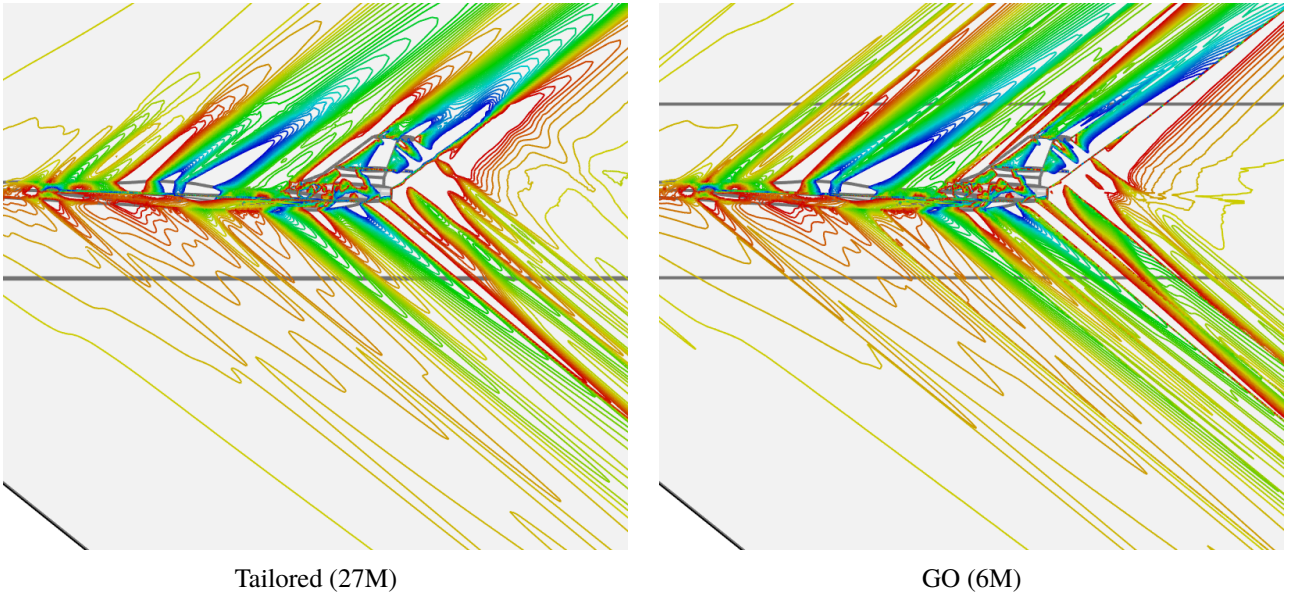


Fig. 17 Pressure signature on the symmetry plane.

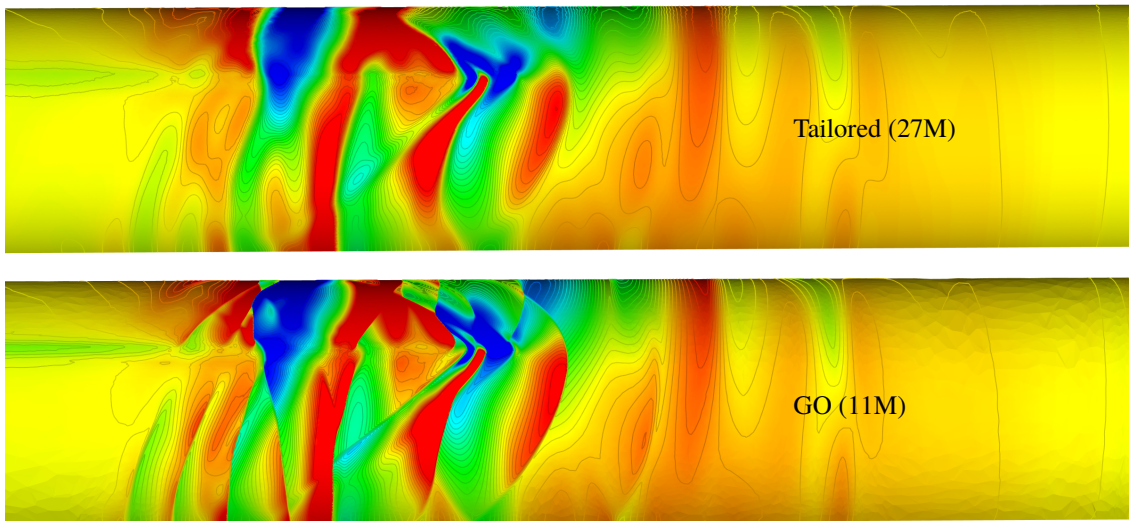


Fig. 18 Pressure imprint on a cylinder at 5m from the aircraft.

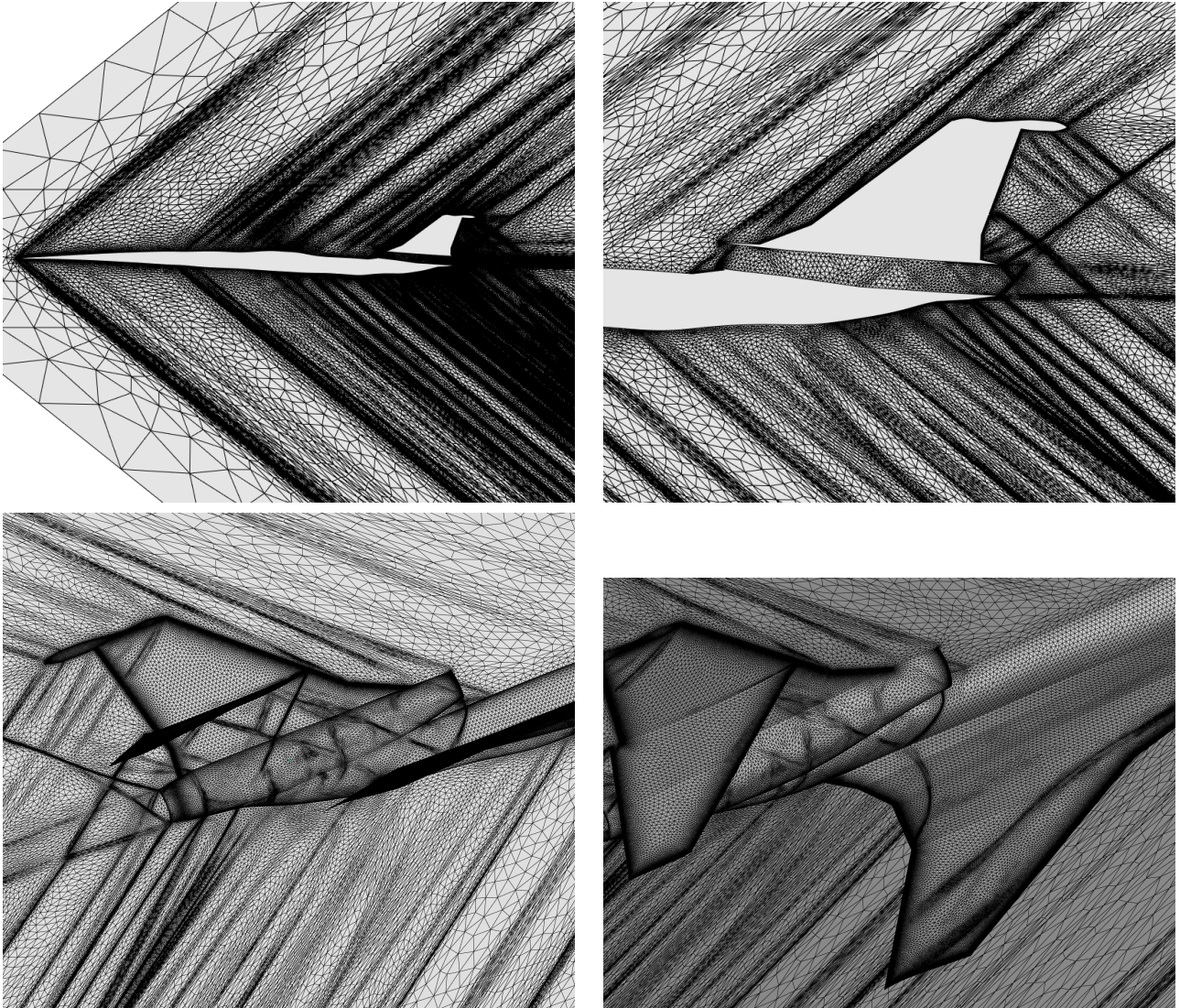


Fig. 19 Surface meshes for the goal-oriented cases on the symmetry plane (top) and on the surface of the aircraft (bottom).

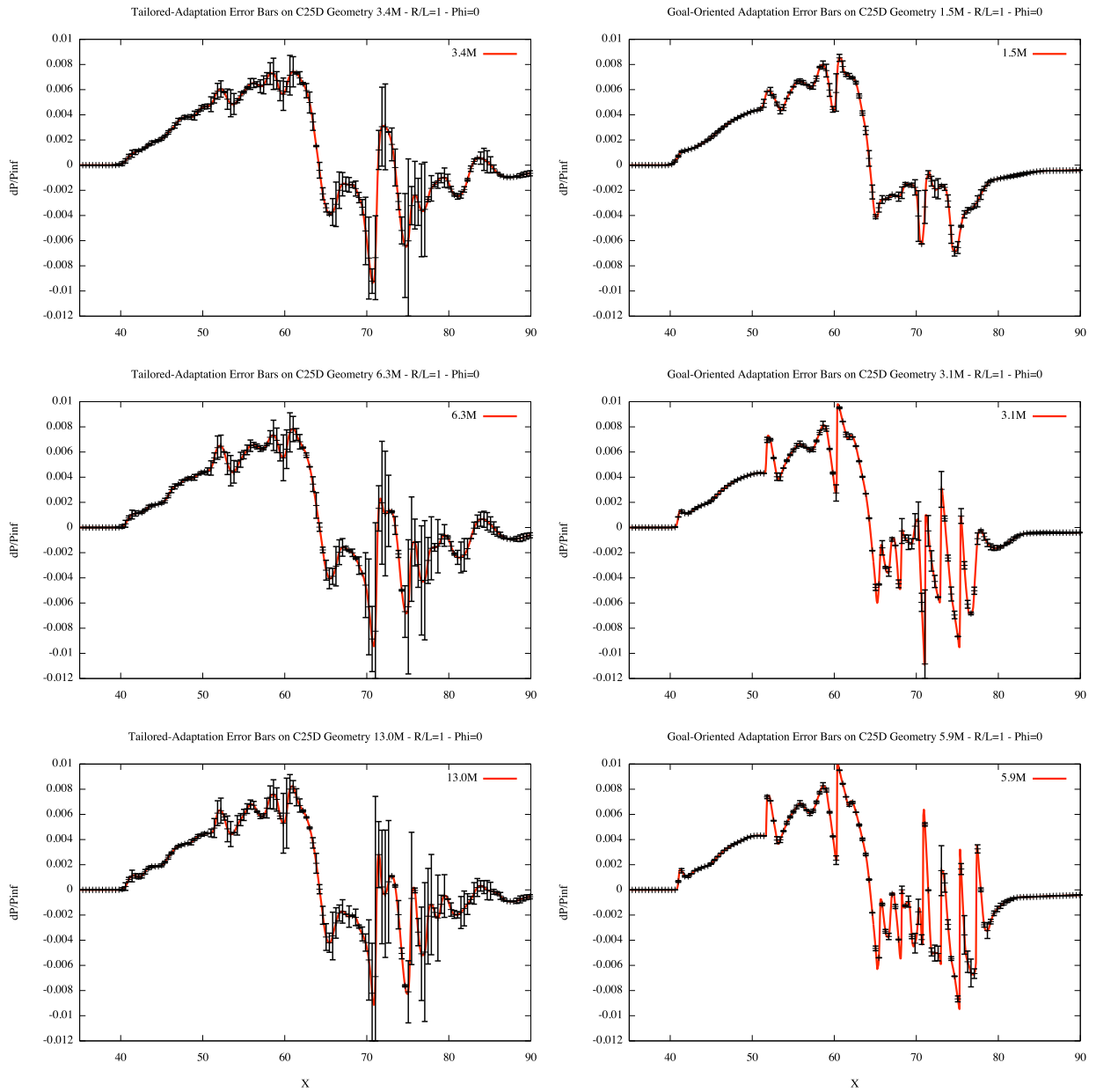


Fig. 20 Non-linear corrector predictions of the pressure on tailored meshes (left) and adapted goal-oriented meshes (right).

R/L are observed. In particular, it may appear appropriate to couple adaptivity in the unstructured part of the domain when designing tailored meshes.

References

- [1] Loseille, A., Dervieux, A., and Alauzet, F., “Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations,” *Journal of Computational Physics*, Vol. 229, 2010, pp. 2866–2897.
- [2] Kirz, J., and Rudnik, R., “DLR TAU Simulations for the Second AIAA Sonic Boom Prediction Workshop,” *Journal of Aircraft*. (Not yet published).
- [3] Elmilgui, A. A., Carter, M., Cliff, S. E., Nayani, S. N., and Pearl, J. M., “USM3D Simulations for 2nd Sonic Boom Workshop,” *Journal of Aircraft*. (Not yet published).
- [4] Anderson, G., Aftosmis, M. J., and Nemec, M., “Cart3D Simulations for the Second AIAA Sonic Boom Prediction Workshop,” *Journal of Aircraft*. (Not yet published).
- [5] Howe, D., “Improved Sonic Boom Minimization with Extendable Nose Spike,” *43rd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, 2005. doi:doi:10.2514/6.2005-1014.
- [6] Park, M. A., and Nemec, M., “Near Field Summary and Statistical Analysis of the Second AIAA Sonic Boom Prediction Workshop,” *Journal of Aircraft*. (Not yet published).
- [7] Rallabhandi, S. K., and Loubeau, A., “Propagation Summary of the Second AIAA Sonic Boom Prediction Workshop,” *Journal of Aircraft*. (Not yet published).
- [8] Park, M. A., Campbell, R. L., Elmilgui, A. A., Cliff, S. E., and Nayani, S., “Specialized CFD Grid Generation Methods for Near-Field Sonic Boom Prediction,” *52nd Aerospace Sciences Meeting*, AIAA, 2014. doi:doi:10.2514/6.2014-0115.
- [9] Wintzer, M., Ordaz, I., and Fenbert, J. W., “Under-Track CFD-Based Shape Optimization for a Low-Boom Demonstrator Concept,” *33rd AIAA Applied Aerodynamics Conference*, AIAA, 2015. doi:doi:10.2514/6.2015-2260.
- [10] Loseille, A., Dervieux, A., and Alauzet, F., “Anisotropic Norm-Oriented Mesh Adaptation for Compressible Flows,” *53rd AIAA Aerospace Sciences Meeting*, AIAA, 2015. doi:doi:10.2514/6.2015-2037.
- [11] Derlaga, J. M., and Park, M. A., “Application of Exact Error Transport Equations and Adjoint Error Estimation to AIAA Workshops,” *55th AIAA Aerospace Sciences Meeting*, AIAA, 2017. doi:doi:10.2514/6.2017-0076.
- [12] Spalart, P., and Allmaras, S., “A one-equation turbulence model for aerodynamic flows,” *30th Aerospace Sciences Meeting and Exhibit*, AIAA, 1992. doi:doi:10.2514/6.1992-439.
- [13] Eca, L., Hoekstra, M., Hay, A., and Pelletier, D., “Verification of RANS Solvers with Manufactured Solutions,” *Engineering with Computers*, Vol. 23, No. 4. doi:doi:10.1007/s00366-007-0067-9.

- [14] Catris, S., and Aupoix, B., "Density corrections for turbulence models," *Aerospace Science and Technology*, Vol. 4, 2000, pp. 1–11. doi:doi:10.1016/S1270-9638(00)00112-7.
- [15] Batten, P., Clarke, N., Lambert, C., and Causon, D. M., "On the choice of wavespeeds for the HLLC Riemann solver," *SIAM Journal on Scientific Computing*, Vol. 18, No. 6, 1997, pp. 1553–1570.
- [16] Leer, B. V., "Towards the ultimate conservative difference scheme I. The quest of monotonicity," *Lecture notes in physics*, Vol. 18, 1973, pp. 163–168.
- [17] Cournède, P.-H., Koobus, B., and Dervieux, A., "Positivity statements for a Mixed-Element-Volume scheme on fixed and moving grids," *European Journal of Computational Mechanics*, Vol. 15, No. 7-8, 2006, pp. 767–798.
- [18] Harten, A., "High Resolution Schemes for Hyperbolic Conservation Laws," *Journal of Computational Physics*, Vol. 49, 1983, pp. 357–393.
- [19] Park, M., and Carlson, J.-R., "Turbulent Output-Based Anisotropic Adaptation," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA, 2010. doi:doi:10.2514/6.2010-168.
- [20] Loseille, A., and Lohner, R., "Anisotropic Adaptive Simulations in Aerodynamics," *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, AIAA, 2010. doi:doi:10.2514/6.2010-169.
- [21] Coupez, T., "Génération de maillages et adaptation de maillage par optimisation locale," *Revue Européenne des Éléments Finis*, Vol. 9, 2000, pp. 403–423.
- [22] Li, X. L., Shephard, M. S., and Beall, M. W., "3D anisotropic mesh adaptation by mesh modification," *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 48-49, 2005, pp. 4915–4950.
- [23] Dobrzynski, C., and Frey, P. J., "Anisotropic Delaunay Mesh Adaptation for Unsteady Simulations," *Proc. of 17th Int. Meshing Roundtable*, Springer, 2008, pp. 177–194.
- [24] Michal, T., and Krakos, J., "Anisotropic mesh Adaptation through edge primitive operations," *AIAA Paper*, Vol. 2011-0159.
- [25] Loseille, A., and Menier, V., "Serial and Parallel Mesh Modification Through a Unique Cavity-Based Primitive," *Proceedings of the 22th International Meshing Roundtable*, Springer, 2013, pp. 541–558.
- [26] Bowyer, A., "Computing Dirichlet tessellations," *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 162–166.
- [27] Watson, D., "Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes," *The Computer Journal*, Vol. 24, No. 2, 1981, pp. 167–172.
- [28] Hermeline, F., "Triangulation automatique d'un polyèdre en dimension N ," *RAIRO - Analyse numérique*, Vol. 16, No. 3, 1982, pp. 211–242.
- [29] Loseille, A., "Metric-orthogonal anisotropic mesh generation," *Proceedings of the 23th International Meshing Roundtable, Procedia Engineering*, Vol. 82, 2014, pp. 403–415. doi:doi:10.1016/j.proeng.2014.10.400.

- [30] Alauzet, F., and Loseille, A., “High Order Sonic Boom Modeling by Adaptive Methods,” *Journal of Computational Physics*, Vol. 229, 2010, pp. 561–593.
- [31] Castro-Díaz, M. J., Hecht, F., Mohammadi, B., and Pironneau, O., “Anisotropic Unstructured Mesh Adaptation for Flow Simulations,” *International Journal for Numerical Methods in Fluids*, Vol. 25, 1997, pp. 475–491.
- [32] Dompierre, J., Vallet, M. G., Fortin, M., Bourgault, Y., Habashi, W., Dompierre, J., Vallet, M. G., Fortin, M., Bourgault, Y., and Habashi, W., “Anisotropic mesh adaptation - Towards a solver and user independent CFD,” *35th Aerospace Sciences Meeting and Exhibit*, AIAA, 1997. doi:doi:10.2514/6.1997-861.
- [33] Vasilevski, Y., and Lipnikov, K., “Error bounds for controllable adaptive algorithms based on a hessian recovery,” *Computational Mathematics and Mathematical Physics*, Vol. 45, No. 8, 2005, pp. 1374–1384.
- [34] Huang, W., “Metric tensors for anisotropic mesh generation,” *Journal of Computational Physics*, Vol. 204, 2005, pp. 633–665.
- [35] Frey, P. J., and Alauzet, F., “Anisotropic mesh adaptation for CFD computations,” *Computer Methods in Applied Mechanics and Engineering*, Vol. 194, No. 48-49, 2005, pp. 5068–5082.
- [36] Chen, L., Sun, P., and Xu, J., “Optimal anisotropic meshes for minimizing interpolation errors in L^p -norm,” *Mathematics of Computation*, Vol. 76, No. 257, 2007, pp. 179–204.
- [37] Loseille, A., and Alauzet, F., “Continuous mesh framework. Part II: validations and applications,” *SIAM J. Numer. Anal.*, Vol. 49, No. 1, 2011, pp. 61–86.
- [38] Loseille, A., Dervieux, A., Frey, P., and Alauzet, F., “Achievement of Global Second Order Mesh Convergence for Discontinuous Flows with Adapted Unstructured Meshes,” *18th AIAA Computational Fluid Dynamics Conference*, AIAA, 2007. doi:doi:10.2514/6.2007-4186.
- [39] Giles, M., and Suli, E., “Adjoint methods for PDEs: a posteriori error analysis and postprocessing by duality,” *Acta Numerica*, Cambridge University Press, 2002, pp. 145–236.
- [40] Jones, W., Nielsen, E., and Park, M., “Validation of 3D Adjoint Based Error Estimation and Mesh Adaptation for Sonic Boom Prediction,” *44th AIAA Aerospace Sciences Meeting and Exhibit*, American Institute of Aeronautics and Astronautics, 2006. doi:doi:10.2514/6.2006-1150.
- [41] Venditti, D. A., and Darmofal, D. L., “Grid adaptation for functional outputs: application to two-dimensional inviscid flows,” *Journal of Computational Physics*, Vol. 176, No. 1, 2002, pp. 40–69.
- [42] Power, P., Pain, C., Piggott, M., Fang, F., Gorman, G., Umpheby, A., and Goddard, A., “Adjoint a posteriori error measures for anisotropic mesh optimization,” *Computers & Mathematics with Applications*, Vol. 52, 2006, pp. 1213–1242.
- [43] Loseille, A., “Adaptation de maillage 3D anisotrope multi-échelles et ciblé à une fonctionnelle. Application à la prédiction haute-fidélité du bang sonique.” Ph.D. thesis, Université Pierre et Marie Curie, Paris VI, Paris, France, 2008.

- [44] Belme, A., Dervieux, A., and Alauzet, F., “Time accurate anisotropic goal-oriented mesh adaptation for unsteady flows,” *Journal of Computational Physics*, Vol. 231, 2012, pp. 6323–6348.
- [45] Giles, M. B., “On adjoint equations for error analysis and optimal grid adaptation in CFD,” Tech. Rep. NA-97/11, Oxford, 1997.
- [46] Giles, M., and Pierce, N., “Improved lift and drag estimates using adjoint Euler equations,” *14th Computational Fluid Dynamics Conference*, AIAA, 1999. doi:doi:10.2514/6.1999-3293.