



Mono-block and non-matching multi-block structured mesh adaptation based on aerodynamic functional total derivatives for RANS flow

A. Resmini, J. Peter, Didier Lucor

► To cite this version:

A. Resmini, J. Peter, Didier Lucor. Mono-block and non-matching multi-block structured mesh adaptation based on aerodynamic functional total derivatives for RANS flow. International Journal for Numerical Methods in Fluids, 2016, 10.1002/fld.4296 . hal-01426042

HAL Id: hal-01426042

<https://hal.science/hal-01426042>

Submitted on 22 Nov 2021

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.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Mono-block and non-matching multi-block structured mesh adaptation based on aerodynamic functional total derivatives for RANS flow

A. Resmini^{1,2,*,†}, J. Peter¹ and D. Lucor³

¹ONERA – The French Aerospace Lab, 29, av. de la Division Leclerc, 92322 Châtillon, France

²Sorbonne University, UPMC Univ Paris 06, UMR 7190, Inst Jean Le Rond d'Alembert, F-75005 Paris, France

³LIMSI, CNRS, Université Paris-Saclay, Campus Universitaire Bâtiment 508, Rue John von Neumann, F-91405 Orsay cedex, France

An enhanced goal-oriented mesh adaptation method is presented based on aerodynamic functional total derivatives with respect to mesh nodes in a Reynolds-Averaged Navier-Stokes (RANS) finite-volume mono-block and non-matching multi-block-structured grid framework. This method falls under the category of methods involving the adjoint vector of the function of interest. The contribution of a Spalart–Allmaras turbulence model is taken into account through its linearization. Meshes are adapted accordingly to the proposed indicator. Applications to 2D RANS flow about a RAE2822 airfoil in transonic, and detached subsonic conditions are presented for the drag coefficient estimation. The asset of the proposed method is patent. The obtained 2D anisotropic mono-block mesh well captures flow features as well as global aerodynamic functionals. Interestingly, the constraints imposed by structured grids may be relaxed by the use of non-matching multi-block approach that limits the outward propagation of local mesh refinement through all of the computational domain. The proposed method also leads to accurate results for these multi-block meshes but at a fraction of the cost. Finally, the method is also successfully applied to a more complex geometry, namely, a mono-block mesh in a 3D RANS transonic flow about an M6 wing.

KEY WORDS: mesh; adaptation; adjoint; RANS; airfoil; wing

1. INTRODUCTION

The extensive use of computational fluid dynamics (CFD) models and resources in the aeronautical community has required the need of accurate predictions. The verification and validation process is then of primary importance. The validation approach of a CFD code with respect to (w.r.t.) experimental data must take into account the estimation of the discretization error [1], modeling error, and the impact of parameters uncertainty [2, 3]. This study is focused on the discretization error and more precisely on the accuracy of the output functionals. Often, only some quantities of interest (QoI), for example, forces and moments, rather than the overall flow-field, are scrutinized in aerodynamic simulations. In this framework, the development of goal-oriented mesh adaptation techniques has seen a rapid growth since the mid 1990s. Most of the time, these methods involve the adjoint vector of the QoI.

Successful theory of *a posteriori* error and goal-oriented mesh adaptation has been developed in the finite-element framework. Important contributions have been given by Johnson *et al.* [4–6],

*Correspondence to: A. Resmini, ONERA – The French Aerospace Lab, 29, av. de la Division Leclerc, 92322 Châtillon, France.

†E-mail: andrea.resmini@onera.fr

Giles *et al.* [7], Prudhomme *et al.* [8], Larson and Barth [9], Machiels *et al.* [10], Hartman *et al.* [11–13], and Alauzet *et al.* [14]. Concerning finite volume, the major reference is the Venditti and Darmofal method [15–17]. A list of main applications involving the latter method can be found in [18].

The present paper presents an enhanced goal-oriented mesh adaptation method based on aerodynamic functional total derivatives w.r.t. mesh nodal coordinates in a finite-volume-structured grid framework for RANS flow. The method principles have been already introduced by Peter *et al.* [18] and Nguyen-Dinh *et al.* [19] with applications to Euler flows. In contrast to the Venditti and Darmofal method that requires two levels of meshes, the proposed method is based on a scalar indicator for one mesh level only. The use of only one mesh level is quite rare in literature; nevertheless, it is possible to notice the contribution of Dwight [20, 21] where only one mesh level is needed but limited to the classical Jameson *et al.* numerical scheme [22].

Using the proposed method, this paper’s objectives are twofold: to properly adapt the mesh taking into account the precious information coming from the complete linearization of the complete Navier–Stokes equations and to show the flexibility of the approach in engineering applications where non-traditional mesh configurations may be used. In this context, adaptations are carried out on standard mono-block-structured grid (2D and 3D RANS flows) as well as on non-matching[‡] multi-block-structured mesh (2D RANS flows). The latter kind of adaptations are uncommon in literature, while their use seems quite natural for complex geometry configuration in the presence of structured mesh. A structured grid may be disadvantageous w.r.t. an unstructured one in terms of the number of nodes (a local refinement spreads through the entire computational domain) and in terms of mesh design flexibility (a structured mesh with well defined quadrilateral elements is hardly compatible with complex configurations). Nevertheless, from a numerical point view, they turn out to be algorithmically efficient because of their simple addressing approach based on regular connectivity, that is, through only two (or three) indices for 2D case (or 3D case). The use of non-matching multi-block meshes alleviates the constraints given by structured mesh. An example of this is shown.

The paper is organized as follows. Section 2 recalls the proposed method and defines the refinement indicator. Section 3 describes the numerical setup used as well as the method verification and the used meshes, while in Section 4, applications to 2D RANS flow about a RAE2822 airfoil in transonic and detached subsonic conditions for the drag coefficient estimation are presented. 3D RANS transonic flow application is shown in Section 5. Conclusions and perspectives are given in Section 6.

2. PROJECTED TOTAL DERIVATIVE OF THE FUNCTIONAL OUTPUT WITH RESPECT TO MESH NODAL COORDINATES

In this section, the proposed method based on the projected total derivative of the functional output w.r.t. mesh nodes is detailed. The derivation follows the papers of Peter *et al.* [18] and Nguyen-Dinh *et al.* [19]. The refinement indicator is presented as well.

2.1. The discrete adjoint method for gradients computation

Given an aerodynamic object where the geometry is subject to N_I shape parameters α_l , $l \in [1, N_I]$, it is denoted by \mathcal{J}_m , $m \in [1, N_J]$ and the N_J objective/constraints functions that need to be differentiated in a gradient-based optimization framework. Concerning the numerical scheme, the discretization of the Navier–Stokes equations leads to a nonlinear equations system defined by the following:

$$\mathbf{R}(\mathbf{W}, \mathbf{X}) = 0, \quad (1)$$

[‡]In literature, the term *non-conforming* is also used to refer to non-conventional mesh types. In this paper, the term *non-matching* is preferred in order to strengthen the point on the type of non-conforming mesh used.

where \mathbf{R} represents the discrete residual vector on each cell, \mathbf{W} is the steady flow solution, and \mathbf{X} defines the mesh coordinates. In order to compute first-order derivatives, it is assumed that

1. \mathbf{X} is C^1 regular, function of α_l ;
2. \mathbf{R} is C^1 regular w.r.t. its two vector arguments; and
3. $\det[\partial \mathbf{R} / \partial \mathbf{W}](\mathbf{W}, \mathbf{X}) \neq 0$ where $\mathbf{R}(\mathbf{W}, \mathbf{X}) = 0$.

In this framework, the implicit function theorem ensures that $\mathbf{W}(\alpha_l)$ is a C^1 regular function of α_l . The objective/constraints functions read as follows:

$$\mathcal{J}_m(\alpha_l) = J_m(\mathbf{W}(\alpha_l), \mathbf{X}(\alpha_l)). \quad (2)$$

In the discrete adjoint method scheme, in order to compute the derivative of the function $\mathcal{J}_m(\alpha_l)$ w.r.t. α_l , a Lagrangian functional is introduced:

$$\mathcal{L}_m(\mathbf{W}, \mathbf{X}, \mathbf{\Lambda}_m) = J_m(\mathbf{W}, \mathbf{X}) + \mathbf{\Lambda}_m^T \mathbf{R}(\mathbf{W}, \mathbf{X}), \quad (3)$$

where $\mathbf{\Lambda}_m$ is the Lagrange multiplier (or adjoint vector) for the objective function \mathcal{J}_m . Given the aforementioned hypothesis, it is possible to define the aerodynamic function J_m as a function of the mesh only, that is, $J_m(\mathbf{X}(\alpha_l)) = J_m(\mathbf{W}(\mathbf{X}(\alpha_l)), \mathbf{X}(\alpha_l))$.

The Lagrange multiplier is then the solution of $d\mathcal{L}_m/d\mathbf{W} = 0$:

$$\mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{W}} = -\frac{\partial J_m}{\partial \mathbf{W}}. \quad (4)$$

The right-hand side of (4) is computed through analytical differentiation, while the Jacobian matrix in the left-hand side is built at steady solution. The derivative of objective functions is then given by the following:

$$\frac{d\mathcal{J}_m}{d\alpha_l} = \frac{d\mathcal{L}_m}{d\alpha_l} = \frac{\partial J_m}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\alpha_l} + \mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\alpha_l} = \left(\frac{\partial J_m}{\partial \mathbf{X}} + \mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \right) \frac{d\mathbf{X}}{d\alpha_l}. \quad (5)$$

Between the previous parenthesis, it is possible to identify the following:

$$\frac{dJ_m}{d\mathbf{X}} = \frac{\partial J_m}{\partial \mathbf{X}} + \mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}}, \quad (6)$$

where the first term ($\partial J_m / \partial \mathbf{X}$) represents the direct dependency of J_m to the mesh nodes location and the second term ($\mathbf{\Lambda}_m^T (\partial \mathbf{R} / \partial \mathbf{X})$) corresponds to the flow field modification on the support of J_m because of a change of mesh nodes location.

In order to avoid the storage of the mesh sensitivity w.r.t. design variables, Nielsen and Park [23] used the *implicit*, $\mathbf{D}(\mathbf{X}, \mathbf{S}) = 0$, or *explicit*, $\mathbf{X} = \mathbf{X}(\mathbf{S})$, dependence between \mathbf{X} and \mathbf{S} , where the latter is the surface mesh. In the first case, the equations read as follows:

$$\mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{W}} = -\frac{\partial J_m}{\partial \mathbf{W}}, \quad (7)$$

$$\mathbf{\Gamma}^T \frac{\partial \mathbf{D}}{\partial \mathbf{X}} = \left(\frac{\partial J_m}{\partial \mathbf{X}} + \mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \right) = \frac{dJ_m}{d\mathbf{X}}, \quad (8)$$

$$\frac{d\mathcal{J}_m}{d\alpha_l} = \left[\mathbf{\Gamma}^T \frac{\partial \mathbf{D}}{\partial \mathbf{S}} \right] \frac{d\mathbf{S}}{d\alpha_l}, \quad (9)$$

while for an explicit dependence, the following equations replace Equation (5):

$$\mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{W}} = -\frac{\partial J_m}{\partial \mathbf{W}}, \quad (10)$$

$$\frac{d\mathcal{J}_m}{d\alpha_l} = \left[\frac{dJ_m}{d\mathbf{X}} \frac{d\mathbf{X}}{d\mathbf{S}} \right] \frac{d\mathbf{S}}{d\alpha_l}. \quad (11)$$

Therefore, the $dJ_m/d\mathbf{X}$ terms are the standard outputs of a modular memory-efficient adjoint solver where the complex parametrization (i.e., $d\mathbf{X}/d\alpha_l$) is not addressed anymore. The proposed goal-oriented method is based on this quantity.

2.2. Derivation of the Spalart–Allmaras turbulence model

In this paper, the Spalart–Allmaras (SA) turbulence model [24] is considered. Namely, the sixth steady RANS equation already coupled with SA reads as follows:

$$\oint_{\partial V} \left(\rho \tilde{v} \mathbf{U}^T - \frac{1}{\sigma} (\mu + \rho \tilde{v}) \nabla \tilde{v}^T \right) \cdot \mathbf{n} dS - \int_V \mathcal{S} dV = 0, \quad (12)$$

where ρ is the density, $\mathbf{U} = (u, v, w)^T$ the velocity vector,[§] and \tilde{v} the kinematic turbulent viscosity that is defined by the turbulence model, in this case SA. V is the cell volume, and the unit normal vector to ∂V is denoted by \mathbf{n} . \mathcal{S} is the source term composed of a production \mathcal{P} , destruction \mathcal{D} , and cross-diffusion \mathcal{C} terms. As mentioned earlier, the analytical differentiation of \mathbf{R} w.r.t. flowfield (\mathbf{W}) and metric (\mathbf{X}) terms needs to be undergone. It is possible either to differentiate only the five mean flow equations without differentiating the turbulent dynamic viscosity μ_T , that is, (12) (frozen μ_T assumption, $\tilde{\mu}_T$), or to fully differentiate the equations system including the turbulence model (μ_T^{lin}).

The convection term is discretized using first-order Roe’s flux. The discrete diffusion term is based on cell-centered gradient of \tilde{v} . The gradients at two adjacent cells are averaged at the common interface and corrected in the direction linking the two cells’ centers, from \tilde{v} cell values in order to have a short stencil in the considered direction. Discrete \mathcal{P} , \mathcal{D} , and \mathcal{C} terms are calculated from cell-centered values and gradients. It is then needed to derive these schemes w.r.t. discrete flowfield (Equation (5)) and mesh nodes coordinates (Equation (6)).

2.3. The refinement indicator

The $dJ/d\mathbf{X}^{\mathbb{Q}}$ field provides valuable information about the sensitivity of J w.r.t. the mesh coordinates \mathbf{X} . Considering the following first-order Taylor expansion:

$$J(\mathbf{X} + d\mathbf{X}_{i,j}) \approx J(\mathbf{X}) + \frac{dJ}{d\mathbf{X}_{i,j}} d\mathbf{X}_{i,j}, \quad (13)$$

the first-order effect on J due to an isolated move of a node $\mathbf{X}_{i,j}^{\mathbb{H}}$ in the polygon defined by the neighboring nodes is linked to the sensitivity $dJ/d\mathbf{X}_{i,j}$. The latter is true for a displacement $d\mathbf{X}_{i,j}$ – for Euler flows, please refer to Section 5 in [25]. This relation suggests that the objective function J is highly sensitive to the $\mathbf{X}_{i,j}$ node if $dJ/d\mathbf{X}_{i,j}$ is large^{**} and the distance of node $\mathbf{X}_{i,j}$ to the closest neighboring node (denoted as $h_{i,j}$) is significant. A refinement in that node location is then needed.

Some cautions need to be made about the obtained $dJ/d\mathbf{X}$ field. In particular, it will be projected according to the mesh node location. Namely, $\text{P}(dJ/d\mathbf{X}) = dJ/d\mathbf{X} - (dJ/d\mathbf{X} \cdot \mathbf{n}) \cdot \mathbf{n}$ if the node is on the support^{††} of J , walls, farfield (or mesh block) borders; $\text{P}(dJ/d\mathbf{X}) = 0$ if the node is on a

[§]The superscript T defined the transpose operator while the subscript T the turbulent component.

[¶]The pedices m and l drop for the sake of brevity.

^{||}For structured grid, (i, j) indices uniquely identify a mesh node.

^{**}In this framework, there is no *quantitative* link between $\theta(i, j)$ and its effect on J . The term *large* is always w.r.t the ensemble of all $\theta(i, j)$ for a given \mathbf{W} and \mathbf{X} . This justifies the use of the term *indicator* instead of *estimator*.

^{††}In the current paper applications, the *support* of J , where J is a global aerodynamic function, is typically the airfoil surface.

corner of the support of J or on a corner of farfield (or mesh block) domain, while $P(dJ/dX) = dJ/dX$ elsewhere. In this paper, the θ -criterion introduced by Peter *et al.* [18] and Nguyen-Dinh *et al.* [19] will be exploited. This indicator follows directly from the second term of the right-hand side of relation (13), and it reads as follows:

$$\theta(i, j) = \left\| P \left(\frac{dJ}{dX} \right) \right\| \frac{h_{i,j}}{2}. \quad (14)$$

The presence of $h_{i,j}$ in the refinement indicator is important in situations where large vector of $P(dJ/dX)$ is encountered in a mesh that is already fine. In this case, it is not possible to perform a mesh refinement by either adding or moving the nodes. The comparison of the results obtained with the μ - (the θ -criterion without $h_{i,j}$) and the θ -criteria has been already performed for Euler flows by Peter *et al.* [18]. θ has been identified as the correct indicator for mesh adaptation.

2.3.1. Venditti and Darmofal method. The classical goal-oriented error-based approach for finite-volume schemes has been introduced by Venditti and Darmofal. The error expression given by Venditti *et al.* in equation (12) of their article [16] is reported here after:

$$J_h(W_h) \simeq J_h(W_h^H) + \underbrace{(\Lambda_h^H)^T R_h(W_h^H)}_{\text{Computable Correction}} + \underbrace{\left(R_h^\Lambda(\Lambda_h^H) \right)^T \left[\frac{\partial R_h}{\partial W_h} \Big|_{W_h^H} \right]^{-1} R_h(W_h^H)}_{\text{error in computable correction}}, \quad (15)$$

where H and h are the characteristic mesh size of the coarse and fine grids, respectively, and R_h^Λ is the adjoint residual operator. In this framework, the improved functional evaluation consists in the first two terms of the right-hand side of relation (15) and the refinement strategy is to reduce the error in computable correction (ECC).

Previous studies for Euler flows have shown good consistency between the refinement zones detected by the θ -criterion and the ECC of the classic error-based method; please refer to Section 2.8 in [18] and to Appendix B in [25]. This comparison is also given for a 2D transonic turbulent flow in Section 4.4.1.

3. NUMERICAL SETUP

3.1. The *elsA* code

The ONERA Navier–Stokes finite-volume compressible *elsA* [26] solver has been used. The numerical scheme is based on Roe flux using a second-order MUSCL scheme (van Albada limiter) for the convective term while first-order Roe flux for the turbulent flow. The viscous fluxes of the averaged and turbulent flows are discretized through a centered scheme with cell-centered gradients corrected at the cell faces. The turbulence closure is assured by the SA turbulence model [24].

3.1.1. Non-matching structured meshes. The *elsA* code mainly deals with structured meshes. This kind of grids clearly shows some limitations in the case of complex geometry and generally results in a high number of nodes needed w.r.t. unstructured meshes. In order to alleviate the latter drawback, *elsA* can handle domain decomposition with discrete meshes that do not have match at their common interface. Thanks to this capability, non-matching joint between the different structured mesh blocks limits the spread of the local mesh refinement through the entire computational domain. The interface between the different domains may be straight or curved. In the first case, the conservation property is ensured, while in the second case, this approach is quasi-conservative [27]. In the examples given in Section 4, the non-matching multi-block strategy limits the local refinement along the i -direction over all of the domain by splitting the mesh in three concentric O-meshes. Despite this flexibility, the optimality of the domain decomposition choice remains an open question.

Table I. Verification with an OAT15A.

Method		Rotation		Camber		Thickness	
		$[\cdot 10^{-3}]$	Err. rel. %	$[\cdot 10^{-1}]$	Err. rel. %	$[\cdot 10^{-2}]$	Err. rel. %
FD		9.6	—	3.5	—	3.4	—
AdjParam	$(\bar{\mu}_T)$	8.178	−14.8	2.568	−26.6	2.458	−27.7
AdjParam	(μ_T^{lin})	9.142	−4.8	2.981	−14.8	2.916	−14.2
AdjMesh	(μ_T^{lin})	9.142	−4.8	2.981	−14.8	2.932	−13.8

Comparison of the derivative of C_D w.r.t shape parameters obtained from FD and discrete adjoint method (AdjParam) with both frozen $(\bar{\mu}_T)$ and linearized μ_T (μ_T^{lin}). Comparison of the derivative obtained through the derivation of C_D w.r.t mesh nodes is also given (AdjMesh).

FD, finite differences.

3.2. Verification

A test case on a OAT15A airfoil has been recently performed [28] in order to assess the contribution of the turbulence model derivation. The following results are representative of the actual accuracy of the *elsA* adjoint [29]. The flow is at $Ma = 0.73$, $Re = 13 \cdot 10^6$ with $AoA = 1.5^\circ$. Three shape parameters have been considered: angle of incidence, thickness through an affinity operator in the thickness coordinates, and camber through a local modification of the camber line.

Recalling Equation (5), in the *elsA* implementation, the $d\mathcal{J}_m/d\alpha_l$ may be computed in two fashions:

$$\left(\frac{d\mathcal{J}_m}{d\alpha_l}\right)_{\text{AdjParam}} = \frac{\partial \mathcal{J}_m}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\alpha_l} + \mathbf{\Lambda}_m^T \frac{\partial \mathbf{R}}{\partial \mathbf{X}} \frac{d\mathbf{X}}{d\alpha_l}, \quad (16)$$

$$\left(\frac{d\mathcal{J}_m}{d\alpha_l}\right)_{\text{AdjMesh}} = \frac{d\mathcal{J}_m}{d\mathbf{X}} \frac{d\mathbf{X}}{d\alpha_l}, \quad (17)$$

where $d\mathbf{X}/d\alpha_l$ is computed through finite differences in both cases. Table I shows the results of $dC_D/d\alpha_l$ given by finite differences, AdjParam (for both $\bar{\mu}_T$ and μ_T^{lin}), and AdjMesh (only for μ_T^{lin}) for the three shape parameters. It is possible to identify three main patterns:

1. As expected, the results given by AdjParam and AdjMesh with μ_T^{lin} agree very well one to each other.
2. The linearization of the turbulence model leads to a significant improvement of the sensitivity estimation; namely, the error is two-third less for the rotation case and halved for the other two cases.
3. The adjoint module of *elsA* behaves very well for global shape parameters, for example, rotation. Some issues rise for local shape modifications.

The adjoint *elsA* has some approximations on farfield boundaries and walls. Matching and non-matching joins pose limit to the adjoint precision because of ghost cells construction in the direct computation ^{‡‡}. It is recalled that because a linear system needs to be solved, local errors in the adjoint derivation spread over the domain and impact the final solution. Nevertheless, the adjoint *elsA* is able to capture the correct QoI sensitivity trend in a wide range of applications, for example, Euler, RANS, 2D and 3D, full aircraft configuration, and matching and non-matching mesh.

4. APPLICATION TO 2D RANS FLOW ABOUT A RAE2822 AIRFOIL

The proposed mesh adaptation technique is applied to a 2D RANS flow about a RAE2822 airfoil at two different operational conditions: a transonic condition that corresponds to case 6 of

^{‡‡}Two raw of ghost cells in the direct computation would need 3 raw of ghost cells in the adjoint one for Roe-MUSCL flux.

Cook *et al.* [30] equally used by Venditti *et al.* [17] and a detached subsonic condition derived from the latter by keeping the same Reynolds to Mach number ratio as in helicopter rotor operational condition. The efficiency of the proposed method is tested over a mono-block mesh as well as on a non-matching multi-block configuration. Refinement based on the functional C_D is considered which estimation is in general harder than the C_L one. For the C_D -based adapted mesh, the C_L value is also given for the sake of completeness. The considered tolerance is equal to 0.0005, that is, five drag counts. This value is in line with Li *et al.* [31]. Higher [32, 33] and lower [17] tolerances are also considered in literature.

4.1. Mesh characteristics

For all applications in this section, structured O-meshes have been used where the index i runs clockwise along the airfoil from the trailing edge, while j goes from the airfoil wall to the farfield boundary. The circular farfield boundary is placed at 150 chords from the airfoil leading edge. The hierarchy of NACA0012 meshes from Vassberg *et al.* [34] has been modified for a RAE2822 airfoil. Moreover, they have been adapted for RANS flows by adding a boundary layer (BL) growth in j direction. The number of mesh nodes in the BL can be imposed where the BL height is computed by using the estimation of the turbulent BL over a flat plate through the one-seventh-power law. The first cell height h_0 is fixed but differs in the two operational conditions in order to have y^+ lower than 1 in both cases. The cell height h in j direction is given by $h_{j+1} = (h_j)p$ where p varies as $1 < p < 1.1$ based on the number of nodes in j , N_j .

Concerning the non-matching multi-block configuration, the original mono-block mesh has been split in three concentric O-meshes. In many engineering applications, the surface mesh refinement is constrained or needs special care. In this context, it has been decided to test the proposed method in the case where the first block mesh (which contains the airfoil) is frozen through all the adaptation procedure. This block is uniformly refined in i with 401 nodes with 40 nodes in the BL (Figure 1(a)). For the two operational conditions, the first block dimension (in j direction) varies. In the subsonic case, it has been enlarged in order to ensure that the recirculation bubble is fully included in the first block. The second and third blocks are adapted following the θ -criterion. The cell height growth is conserved through the fixed block interfaces.

4.1.1. Refinement procedure. The refinement is carried out by adding lines in both directions i and j . The new mesh is interpolated in the reference fine grid with a number of nodes $N_i = N_j = 2049$.

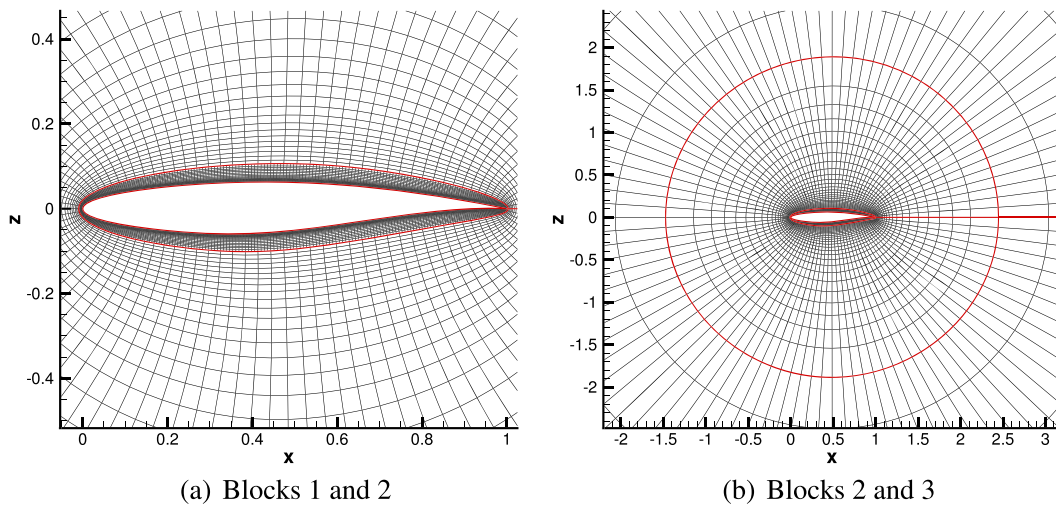


Figure 1. Starting non-matching multi-block mesh. In the special case of the starting mesh, the interface between blocks 2 and 3 is match. During the adaptation steps, mesh nodes are free to move in a non-matching framework; see, for instance, Figure 5(b) and (c). (a) Blocks 1 and 2 and (b) blocks 2 and 3.

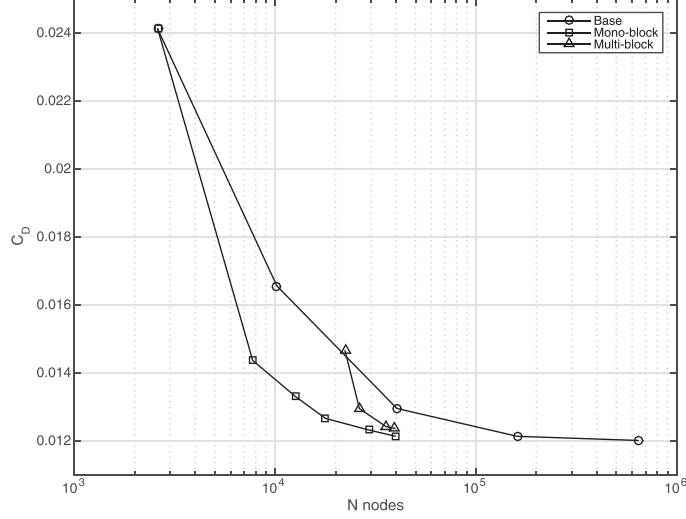


Figure 2. Convergence of C_D for the base solution, mono-block, and multi-block adaptation in the transonic flow condition.

The nodes position is defined through the following interpolation operators ϕ and \mathcal{B} :

$$\{1, N_i\}\{1, N_j\} \xrightarrow{\phi} [1, 2049][1, 2049] \xrightarrow{\mathcal{B}} \mathbb{R}^2. \quad (18)$$

In order to add lines in the current mesh, a criterion is introduced based on the θ -criterion (14). For the sake of brevity, only the i direction is considered. Introducing \bar{c} as

$$\bar{c}_{i+\frac{1}{2}} = \frac{c_i + c_{i+1}}{2} \quad \text{where} \quad c_i = \frac{1}{N_j} \sqrt{\sum_{j=1}^{N_j} \theta^2}, \quad (19)$$

the method consists in adding a number of lines at $i + \frac{1}{2}$, which is proportional to $\bar{c}_{i+\frac{1}{2}}$ value. A maximum number equals to 4 is retained for each $i + \frac{1}{2}$ location. In the particular case of i direction, the mesh lines (cell widths) are regularized in order to ensure a smooth nodes location^{§§}. An explicit third-order dissipation flux is used. The cell widths ϵ are obtained through $n + 1$ iterations through the relation $\epsilon_{i+\frac{1}{2}}^{n+1} = \epsilon_{i+\frac{1}{2}}^n + D_{i+1}^4 - D_i^4$ where $D_i^4 = -k^4(\epsilon_{i+\frac{3}{2}} - 3\epsilon_{i+\frac{1}{2}} + 3\epsilon_{i-\frac{1}{2}} - \epsilon_{i-\frac{3}{2}})$. These fluxes are set to 0 at i equals 1 and N_i , while $\epsilon_{\frac{1}{2}} = \epsilon_{\frac{3}{2}}$, and $\epsilon_{N_i+\frac{1}{2}} = \epsilon_{N_i-\frac{1}{2}}$.

4.2. Transonic flow

As mentioned earlier, a transonic condition is retained as in Section 6.2.1 of the paper of Venditti *et al.* [17]. The flow is at Reynolds number equals to $6.5 \cdot 10^6$ and Mach number at 0.725 with $AoA = 2.466^\circ$. A shock wave appears at the pressure side at $x/C \approx 0.55$ where C is the chord.

The starting mesh is the same for the standard base refinement (i.e., by doubling $N_{i,j}$ at each step) and for the C_D -based mono-block adaptation, namely, with $N_i = N_j = 51$ uniformly distributed in i direction and $N_{BL} = 10$. Concerning the multi-block adaptation, the first block is frozen and uniformly refined in i with 401 nodes with 40 nodes in the BL as mentioned in Section 4.1. This first block has been extracted from the second last base mesh of the black curve in Figure 2. The remaining blocks have $N_i = 101$. The C_D convergence is plotted in Figure 2 where the asset given by the proposed method is patent for the mono-block adaptation w.r.t. the standard refinement. The method equally managed to refine the non-matching multi-block case where the physical shock crosses the

^{§§}For j direction, there is no need for a smoothing operator because the cell height h follows the aforementioned rule from the wall to the farfield boundary.

non-matching interface between blocks 1 and 2. The limit values may be found in Table II. For both kind of meshes, the C_D estimation error is under four drag counts. A very good agreement also between numerical results and experiments is proved. The underestimation of C_D is visible also in figure 8 (right) of the paper of Venditti *et al.* [17] through the Allmaras's computation. The C_L values show excellent agreement one to each other. The nodes saving for the mono-block case are of the order of 75% w.r.t. the second last base mesh. While concerning the multi-block case, please note that almost 19,000 nodes out of 39,500 are imposed by the frozen first block. The precision scatter between the results of the mono-block and multi-block meshes may be explained by the quasi-conservative approach at high curvature interfaces (e.g., leading edge) between the two non-matching blocks, as mentioned in Section 3.1.1.

4.3. Subsonic detached flow

For the subsonic detached case, the operational condition has been derived from the transonic one by keeping the same Reynolds to Mach number ratio as in helicopter rotor operational condition. Namely, the flow is at Reynolds number equals to $2.7 \cdot 10^6$ and Mach number at 0.3 with $AoA = 11^\circ$. The converged flow exhibits a trailing edge separation bubble at $x/C > 0.9$.

Figure 3 shows the C_D convergence. The starting mesh is equal to the one used for the transonic case except for the first cell height h_0 and the physical size of the first block in the case of multi-block refinement as mentioned in Section 4.1. This justifies the different starting point in terms of

Table II. Transonic case.

	Block	N_i	N_j	N_{BL}	N_{TOT}	C_D	C_L
Experiment	—	—	—	—	—	0.0127	0.743
Base (finest)	—	801	801	80	641,601	0.01200	0.744
Mono-block	—	221	179	77	39,559	0.01213	0.738
Multi-block	1 (frozen)	401	47	40	39,437	0.01235	0.743
	2	210	71	—			
	3	142	40	—			

Comparison of C_D and C_L values obtained from the standard base refinement (finest grid) and from the C_D -based adapted mono-block and multi-block meshes. The number of nodes in i and j directions is shown as well as the number of nodes in the BL and the total mesh size, N_{TOT} . The experimental values from Cook *et al.* [30] are equally reported.

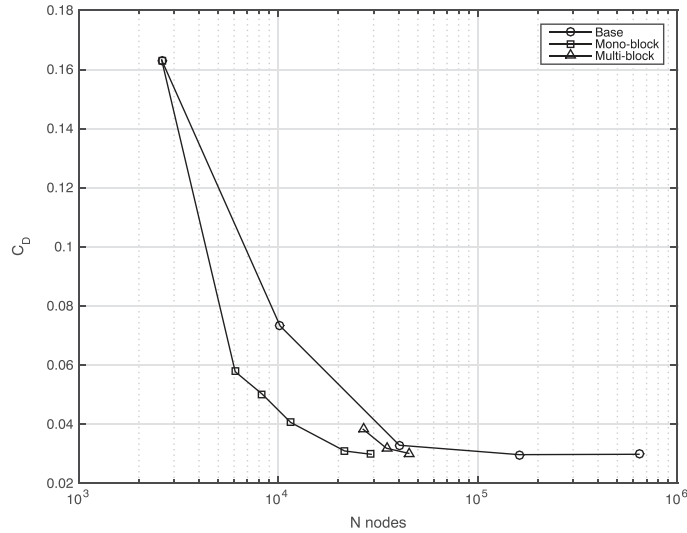


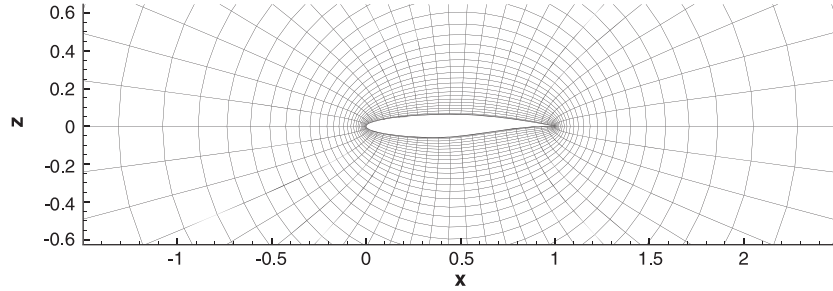
Figure 3. Convergence of C_D for the base solution, mono-block, and multi-block adaptation in the subsonic detached flow condition.

number of nodes of the blue curves in Figures 2 and 3. Also in this case, the proposed method well behaved w.r.t. the standard base refinement. In Table III, the limit values are reported. The C_D estimation error is also in this case of maximum three drag counts while of 0 drag count for the mono-block case. The C_L values are also all coherent. The nodes saving for the mono-block case

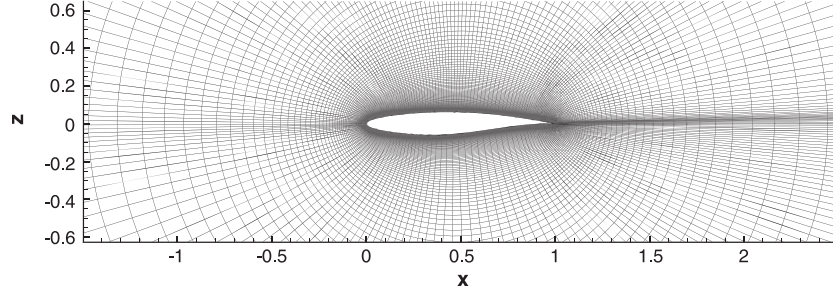
Table III. Subsonic detached case.

	Block	N_i	N_j	N_{BL}	N_{TOT}	C_D	C_L
Base (finest)	–	801	801	80	641,601	0.02987	1.295
Mono-block	–	163	179	71	29,177	0.02986	1.290
Multi-block	1 (<i>frozen</i>)	401	59	40			
	2	178	80	–	45,027	0.03018	1.293
	3	132	54	–			

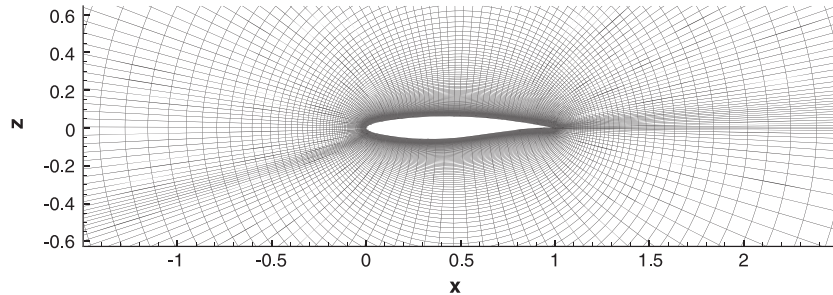
Comparison of C_D and C_L values obtained from the standard base refinement (finest grid) and from the C_D -based adapted mono-block and multi-block meshes. The number of nodes in i and j directions is shown as well as the number of nodes in the BL and the total mesh size, N_{TOT} .



(a) Starting mesh (2061 nodes)

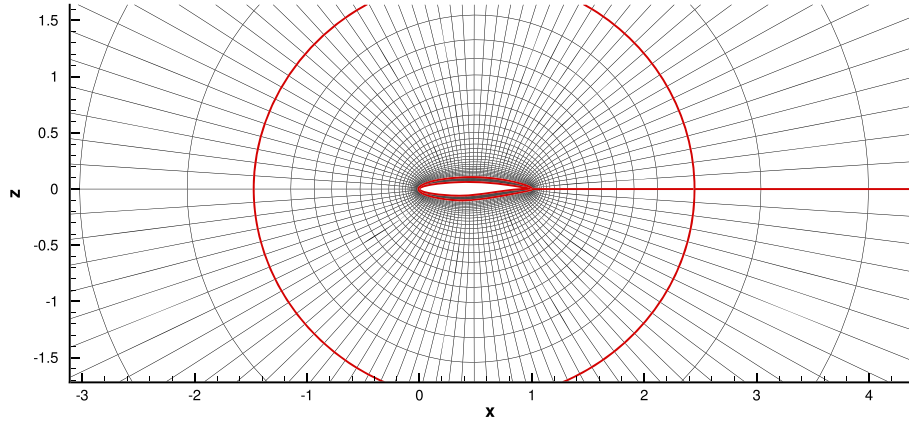


(b) C_D -based adapted mesh, transonic flow (39559 nodes)

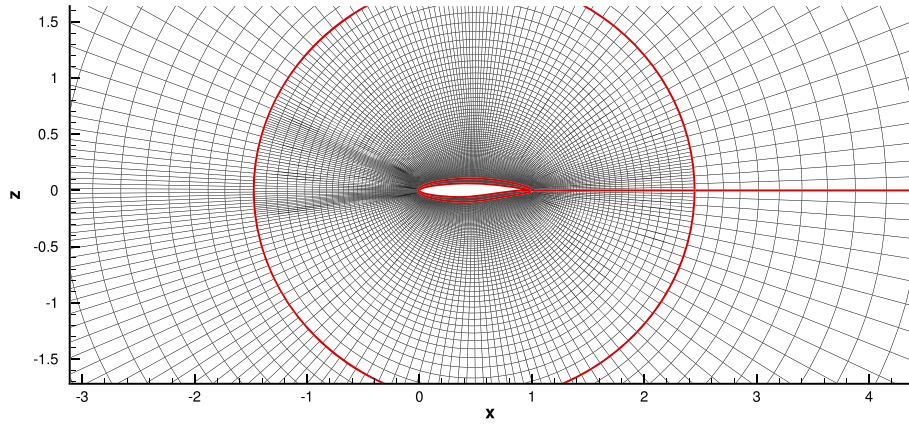


(c) C_D -based adapted mesh, subsonic flow (29117 nodes)

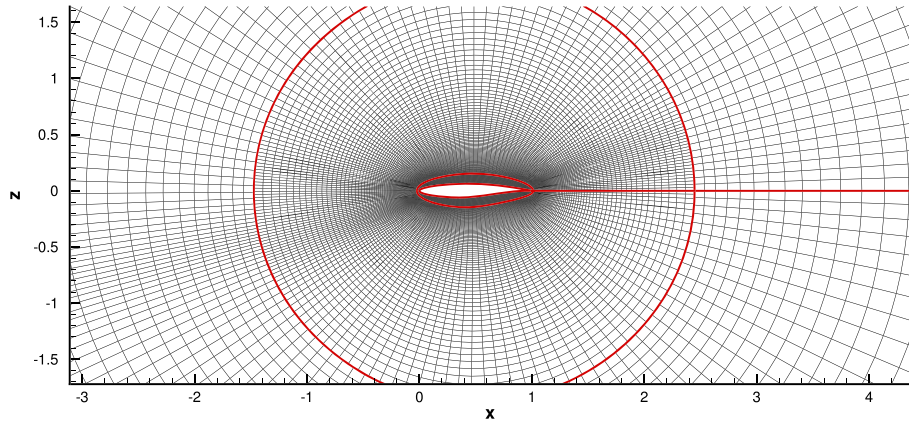
Figure 4. Comparison of mono-block meshes. The starting mesh is given in (a) starting mesh (2061 nodes) while final adapted meshes in (b) C_D -based adapted mesh, transonic flow (39,559 nodes) and (c) C_D -based adapted mesh, subsonic flow (29,117 nodes).



(a) Starting mesh - transonic (22284 nodes) - subsonic (26603 nodes)



(b) C_D -based adapted mesh, transonic flow (40005 nodes)



(c) C_D -based adapted mesh, subsonic flow (45027 nodes)

Figure 5. Comparison of non-matching multi-block meshes. The starting mesh is given in (a) while final adapted meshes in (b) and (c). (a) Starting mesh – transonic (22,284 nodes) and subsonic (26,603 nodes), (b) C_D -based adapted mesh, transonic flow (40,005), (c) C_D -based adapted mesh, subsonic flow (45,027 nodes).

are of the order of 82% w.r.t. the second last base mesh. For the multi-block mesh, almost 24,000 nodes out of 45,000 are imposed by the frozen first block.

4.4. Analysis of the obtained mesh pattern

In order to deepen the analysis of the proposed method and of the results, it is interesting to get a closer look to the obtained anisotropic C_D -based adapted meshes as well as to the values of the θ -criterion. This analysis is carried out only for the mono-block case for the sake of brevity. The same considerations apply to the multi-block case.

Figure 4 shows the starting mesh as well as the C_D -based adapted ones for both operational conditions. The image layout is similar to the one of figure 9 of the paper of Venditti *et al.* [17], which allows a straightforward comparison between the obtained meshes. Taking into account the intrinsic construction differences between structured and unstructured grids, in the transonic case (b), similar areas are highlighted, namely, the upstream flow region (typical for adjoint-based adaptation), the wake, the shock region at $x/C \approx 0.55$, and leading and trailing edge. In structured mono-block grid, it is clear that these local refinements are convected through all the computational domain. For the subsonic case (c), refinement is performed in the upstream flow region, separation region at $x/C > 0.8$, wake, and leading and trailing edge.

In Figure 5, the starting mesh and the C_D -based adapted ones for both operational conditions for the multi-block case are shown. The block interfaces are depicted in green. The same area highlighted in Figure 4 is visible here. In this case, the use of non-matching multi-block approach limits the convection of these refinements to the farfield boundaries. For the upstream flow region, the refinement crosses the interfaces and equally for the wake region but in a lightly fashion.

The Mach iso-contours are given in Figures 6–8 for the starting mesh as well as for the finest mesh of the standard base refinement and for the C_D -based adapted meshes for both operational conditions. Good agreement is proved between Figure 7(a) with Figure 8(a) and Figure 7(b) with Figure 8(b). Please note the poor initial estimation of the flow for the starting meshes (Figure 6).

4.4.1. The θ -criterion and comparison with the Venditti and Darmofal indicator. The θ -criterion for the mono-block transonic case adaptation is shown in Figure 9 for all adaptation steps. It is possible to notice the progressive reduction of the θ values as well as the crucial zones for this operational condition, for example, region of the flow upstream, the airfoil, and the wake and the shock regions at $x/C \approx 0.55$. Please note that the relation (14) that defines θ depends both on the local mesh size through $r_{i,j}$ and also on the $dJ/d\mathbf{X}$. For this reason, θ may also vary in regions where no mesh refinement has been performed, for example, pressure side in Figure 9(e)–(f).

A comparison between the refinement zones detected by the θ -criterion (Figure 9) and by the ECC of the classic error-based method (Figure 10) has been performed on the six meshes of the θ -based adaptation. As expected, the two different indicators highlight the same zones, namely,

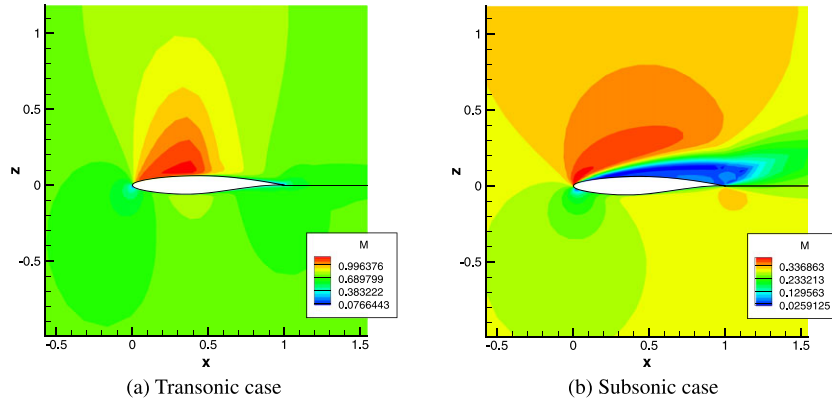


Figure 6. Comparison of the Mach iso-contour with the mono-block starting mesh. (a) Transonic case and (b) subsonic case. [Colour figure can be viewed at wileyonlinelibrary.com]

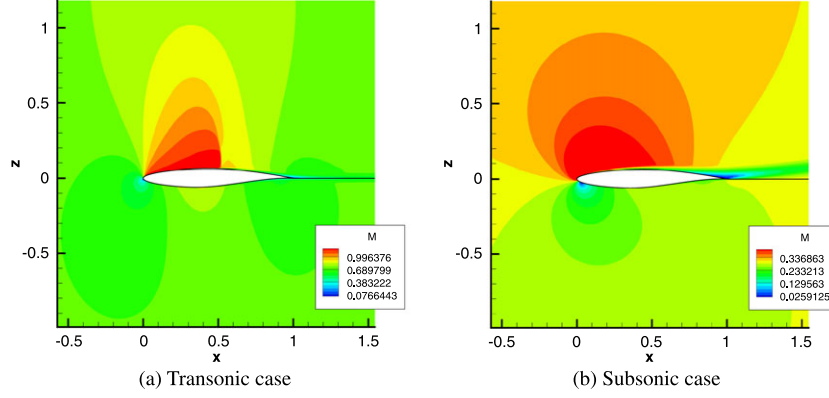


Figure 7. Comparison of the Mach iso-contour with the base mono-block finest mesh. (a) Transonic case and (b) subsonic case. [Colour figure can be viewed at wileyonlinelibrary.com]

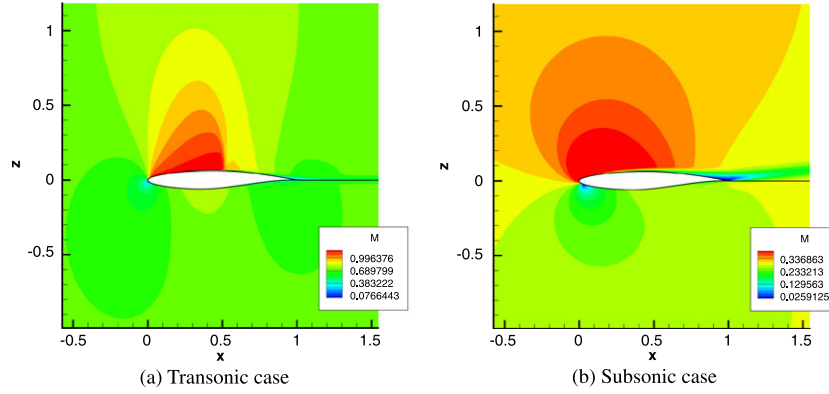


Figure 8. Comparison of the Mach iso-contour with the C_D -based adapted mono-block mesh. (a) Transonic case and (b) subsonic case. [Colour figure can be viewed at wileyonlinelibrary.com]

the wake, the zone upwind the profile, and the shock regions. Please refer to previous studies [18, 25] where the use of the more flexible unstructured grid allows to properly appreciate the agreement between the two approaches.

5. APPLICATION TO 3D RANS FLOW ABOUT AN M6 WING

In this section, a tri-dimensional application is presented. In this context, the proposed mesh adaptation technique is expected to reduce the mesh nodes needed for a proper estimation of the global aerodynamic functions. Nevertheless, the growth of mesh nodes due to the spread through the entire computational domain following a local refinement is expected to be stronger than in 2D case. The adaptation is carried out for 3D RANS transonic flow about an ONERA M6 wing [35]. The wing is symmetric w.r.t. $z = 0$. The most common flow condition used for CFD simulations corresponds to the test number 2308 of [35]. The efficiency of the proposed method is tested over a mono-block mesh configuration. As in Section 4, refinement based on the functional C_D is considered with the same convergence tolerance, namely, 0.0005.

5.1. Mesh characteristics and refinement

Figure 11 depicts the starting regular mesh with $N_i = N_j = 65$ and $N_k = 33$ (139,425 nodes). A structured O-O mesh has been used where the index i runs clockwise along the wing section from the trailing edge along the pressure side and back to the trailing edge along the suction side (green planes at i_{min} and $i = 40$ in Figure 11(b)), the index j goes from the wing surface to

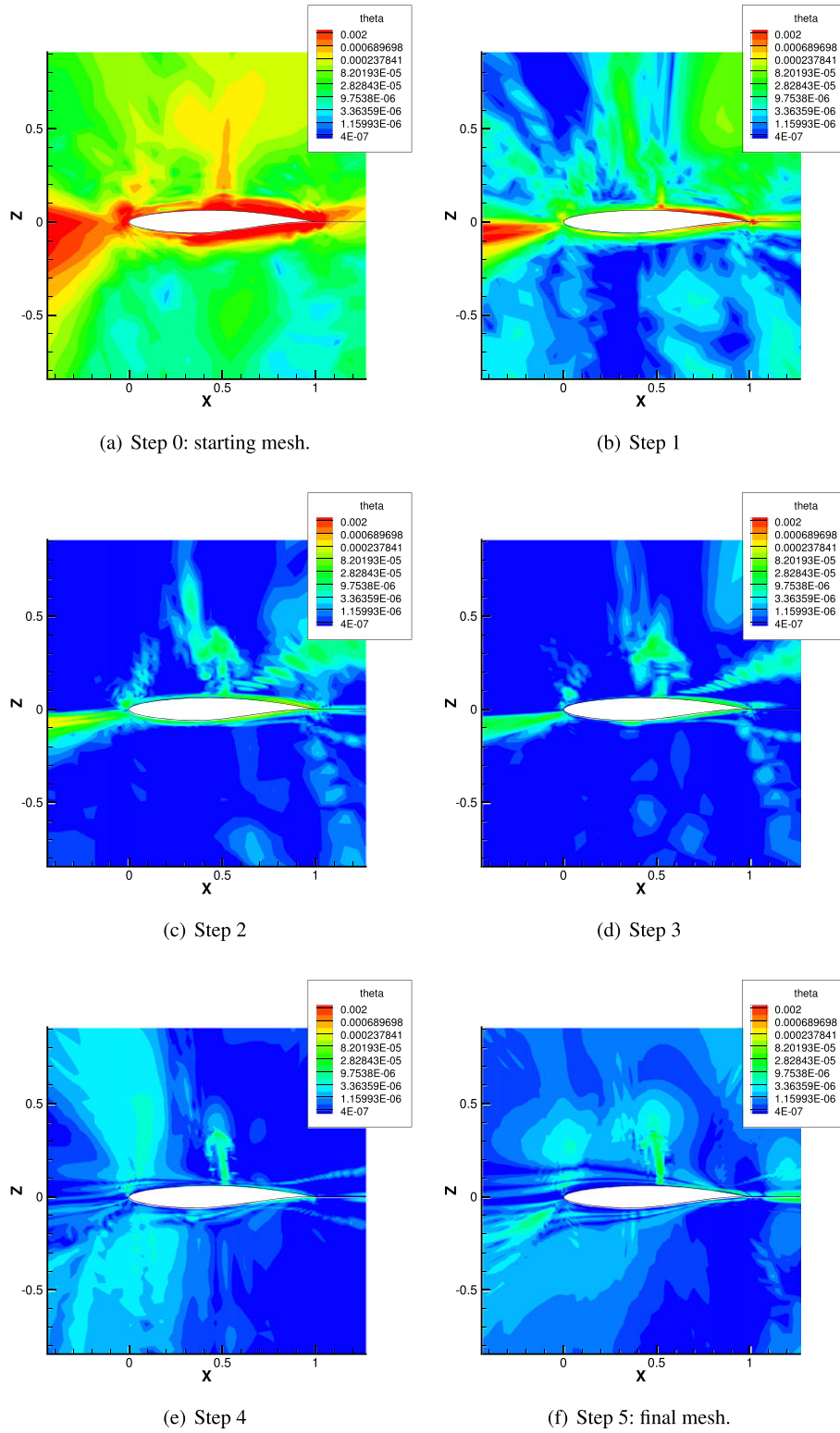


Figure 9. Comparison of the θ -criterion for the mono-block transonic case mesh adaptation. (a) Step 0: starting mesh, (b) step 1, (c) step 2, (d) step 3, (e) step 4, (f) step 5: final mesh. [Colour figure can be viewed at wileyonlinelibrary.com]

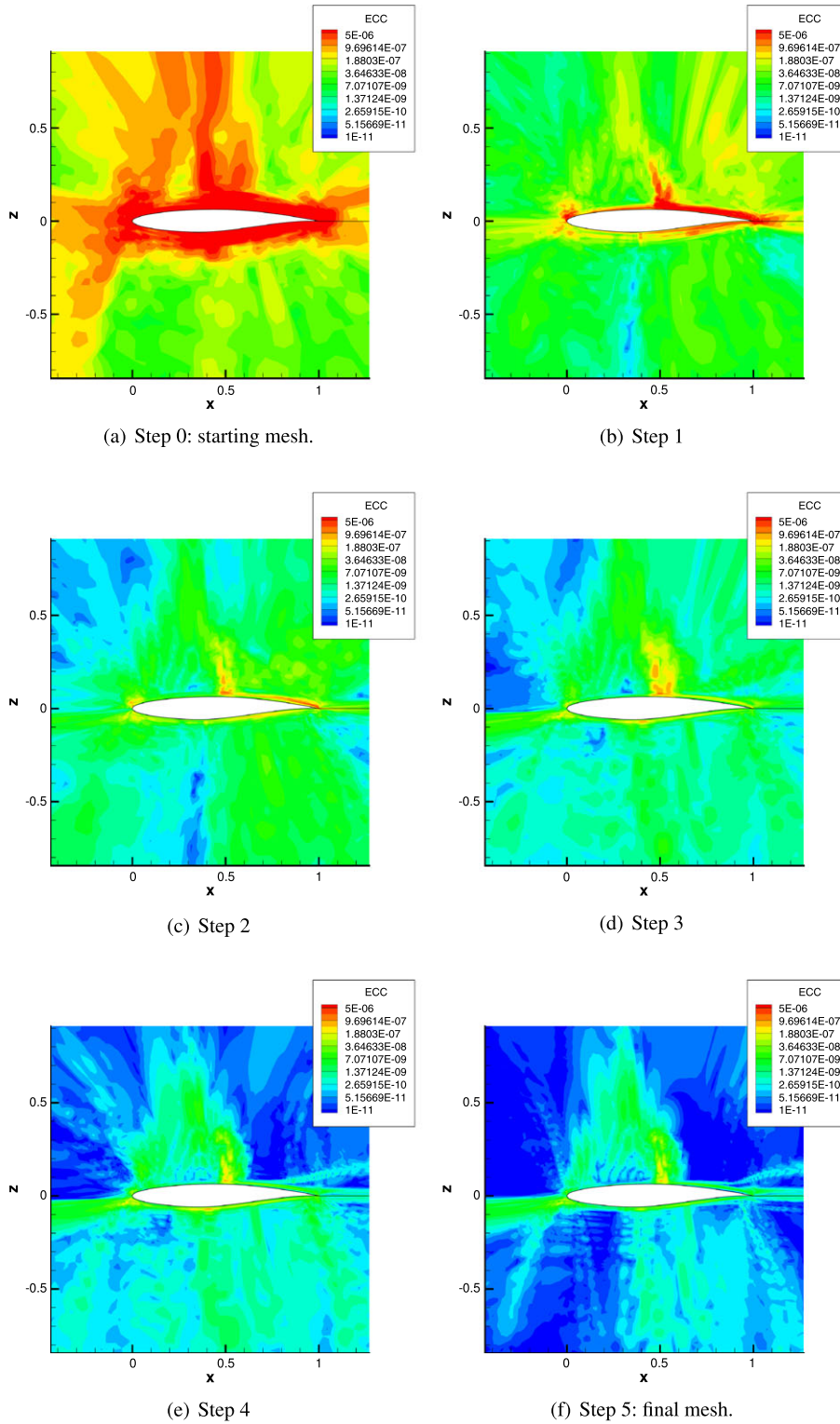


Figure 10. Comparison of the error in computable correction (ECC) of Venditti and Darmofal for the mono-block transonic case mesh adaptation. Please note that the meshes with which this analysis has been performed are the same of Figure 9. (a) Step 0: starting mesh, (b) step 1, (c) step 2, (d) step 3, (e) step 4, (f) step 5: final mesh. [Colour figure can be viewed at wileyonlinelibrary.com]

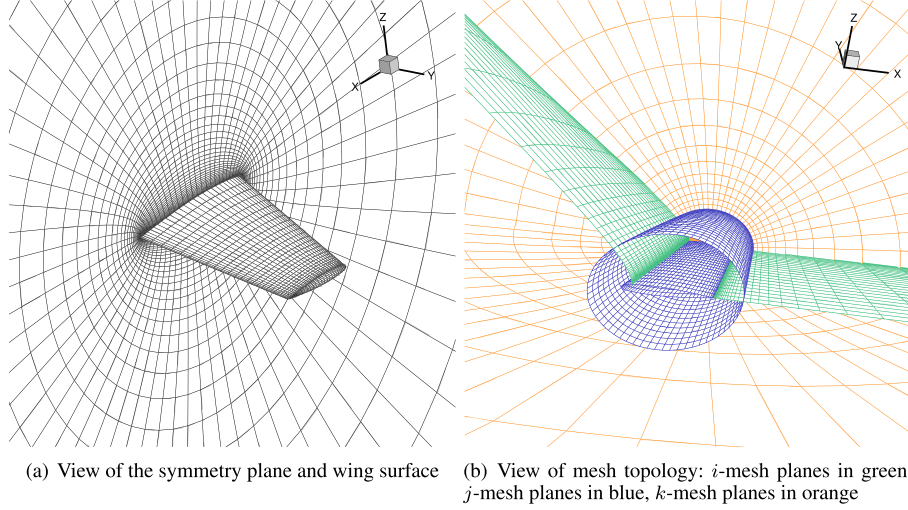


Figure 11. ONERA M6 wing. The starting regular structured mesh ($N_i = N_j = 65$, $N_k = 33$) prior to adaptation. (a) View of the symmetry plane and wing surface and (b) view of the mesh topology: i -mesh planes in green, j -mesh planes in blue, and k -mesh planes in orange.

the farfield boundary (blue planes at j_{min} , which corresponds to the wing surface and $j = 45$ in Figure 11(b)), and the index k runs from the tip to the root of the wing (orange planes at $k = 10$ in Figure 11(b)). Please note that due to mesh assembly, the k -mesh planes are symmetric w.r.t. $z = 0$. The hemispheric farfield boundary is placed at 145 chords from the leading edge. The mesh is interpolated in the reference fine grid with $N_i = N_j = 513$ and $N_k = 257$ in the same fashion as in the 2D case (Section 4.1.1). Thus, all i -mesh planes are identified by the vector ϕ_i , similarly by vector ϕ_j and ϕ_k for all j -mesh and k -mesh planes. The interpolated mesh is then uniquely identified by the three vectors ϕ_i , ϕ_j , and ϕ_k with dimension N_i , N_j , and N_k , respectively.

5.2. Transonic flow

As mentioned earlier, a transonic condition is retained. The flow is at $Re = 11.72 \cdot 10^6$ based on the mean aerodynamic chord of 0.64607 m and Mach number at 0.8395 with $AoA = 3.06^\circ$ and null angle of sideslip. Under this condition, the classical λ -shock structure is observed on the wing surface. For this study, the trailing edge closure leads to a root chord of length 0.8131, and the trapezoidal part of the original M6 wing leads to a reference surface of 0.7532 [18, 36].

The starting mesh prior to adaptation is the same for the standard base refinement (i.e., by doubling the total number of nodes at each step) and for the C_D -based mono-block adaptation: namely, with $N_i = N_j = 65$ and $N_k = 33$, uniformly distributed in i and k directions and $N_{BL} = 20$. The C_D convergence is plotted in Figure 12 for regularly refined and θ -based adapted grids. The asset given by the proposed method is patent. The limit values may be found in Table IV. The C_D estimation error in the C_D -based adapted mesh is under five drag counts w.r.t. both the estimation from the finest standard grid and from the Richardson method [37]. Concerning the C_L , the estimation error is under one lift count. In the context of 3D meshes, in order to easily identify the refinement zones, Figure 13 has been added. The figure shows the index vectors ϕ for the mesh interpolation of the starting base mesh and of the C_D -based adapted mesh. Please note the uniform distribution in i and k while the exponential distribution in j (due to the BL) for the starting base mesh. The low slope areas in the ϕ distribution of the C_D -based adapted mesh correspond to refinement areas. For instance, in Figure 13(a), a refinement is undergone around i_{min} and i_{max} (trailing edge – please recall the mesh topology explained in Section 5.1) as well as around $i = 0.5$ (leading edge). Similarly, in Figure 13(c), a refinement is identified around k_{min} (wing tip) and k_{max} (symmetry plane). This kind of analysis is more difficult for j indices because of the exponential growth in the BL.

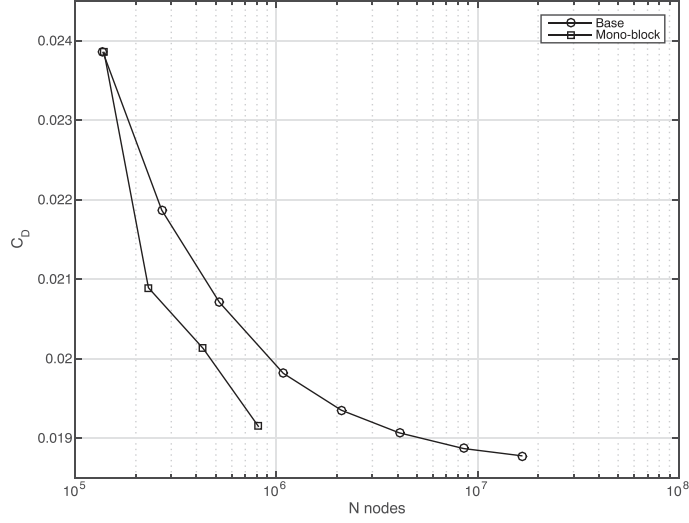


Figure 12. Convergence of C_D for the regular and adapted meshes in the transonic 3D flow condition about the M6 wing.

Table IV. 3D ONERA M6 wing case.

	N_i	N_j	N_k	N_{BL}	N_{TOT}	C_D	C_L
Richardson	—	—	—	—	—	0.01870	0.274
Base (finest)	321	321	161	64	$16.59 \cdot 10^6$	0.01877	0.275
Mono-block	109	131	57	34	$0.813 \cdot 10^6$	0.01915	0.268

Comparison of C_D and C_L values obtained from the standard base refinement (finest grid) and from the C_D -based adapted mono-block mesh. The number of nodes in i and j directions is shown as well as the number of nodes in the BL and the total mesh size, N_{TOT} . The limiting values from the Richardson method are equally given.

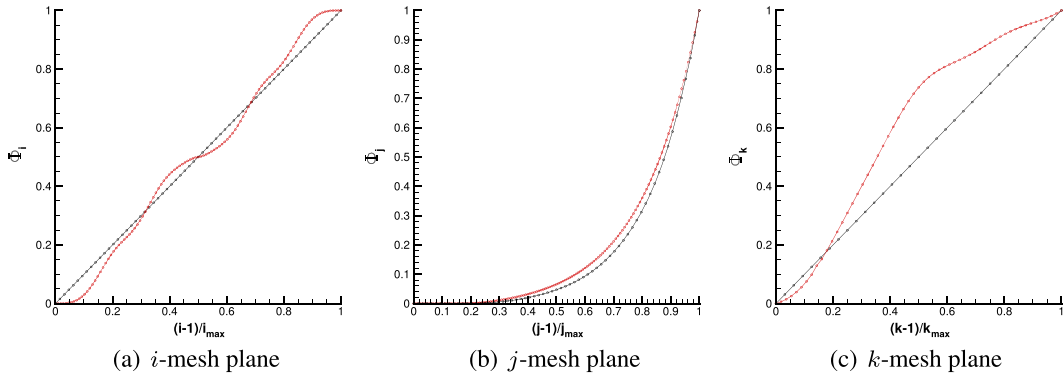


Figure 13. Index vectors ϕ for the mesh interpolation of the starting base mesh (black line) and of the C_D -based adapted mono-block mesh (red line). Please note the uniform distribution in i and k while the exponential distribution in j for the starting base mesh. (a) i -Mesh plane, (b) j -mesh plane, and (c) k -mesh plane.

6. CONCLUSION

An enhanced goal-oriented mesh adaptation method has been presented based on aerodynamic functional total derivatives w.r.t. mesh nodes in a finite-volume mono-block and non-matching multi-block-structured grid framework for RANS flow. The linearization of the SA turbulence model

improves the precision of the estimated adjoint. The retained refinement indicator θ based on the $dJ/d\mathbf{X}$ and on the local mesh size has been proved to be a robust tool for identifying zones considered as influential in the QoI estimation. Applications to 2D RANS flow about a RAE2822 airfoil have been presented. For the mono-block mesh case, for both aerodynamic conditions, the obtained C_D -based adapted meshes lead to a good estimation of forces. Although structured grid has been used, a clear anisotropy in the mesh pattern is visible. The constraints given by this kind of meshes have been relaxed thanks to non-matching multi-block technique, which limits the spread of local mesh refinement through all the domain. In this context, the proposed method has proven to be efficient also to adapt the multi-block mesh in the presence of a more refined frozen-wall mesh block. This recalls how also regions of the domain, which are not close to the airfoil boundary, need refinement. This is equally true for QoI (thus J) whose support is only the airfoil surface mesh (e.g., C_D), which is entirely included in the well-refined frozen-wall mesh block. The 3D RANS transonic flow about a M6 wing case has proven the asset of the refinement approach even for a more complex geometry. The numerical method has anyway shown its efficiency and versatility in terms of flow types (Euler [18] and RANS here), of aerodynamic operational conditions (transonic and detached subsonic), and of mesh types (mono-block and multi-block under constraints). It has been recently extended to unstructured grids [25] for Euler flows, and this could be the next line of work for RANS flows.

ACKNOWLEDGEMENTS

The authors express their gratitude to A. Dumont for the verification computations presented in this paper. The authors are greatly thankful to J.C. Vassberg and A. Jameson for the hierarchy of NACA0012 O-meshes made available for the co-workers of D. Destarac.

This research is fully supported by the Marie-Curie ITN-ANADE funded through the European Union FP7 under grant agreement N:289428.

REFERENCES

1. Oberkampf WL, Oberkampf and Blottner FG. Issues in computational fluid dynamics code verification and validation. *AIAA Journal* 1998; **36**(5):687–695.
2. Roache PJ. Quantification of uncertainty in computational fluid dynamics. *Annual Review of Fluid Mechanics* 1997.
3. Resmini A, Peter J, Lucor D. Sparse grids-based stochastic approximations with applications to aerodynamics sensitivity analysis. *International Journal for Numerical Methods in Engineering* 2016; **106**(1):32–57.
4. Johnson C, Rannacher R, Boman M. Numerics and hydrodynamics theory: towards error control in CFD. *SIAM Journal on Numerical Analysis* 1995; **32**:1058–1079.
5. Becker R, Rannacher R. Weighted a posteriori error control in FE methods. In *Proceedings of ENUMATH-97*, World Scientific Publishing, Heidelberg, 1998.
6. Becker R, Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica* 2001:1–102.
7. Giles M, Larson MG, Levenstam JM, Süli E. Adaptive error control for finite element approximation of lift and drag coefficient in viscous flow. *Technical Report NA-97/06*: Comlab, Oxford University, 1997.
8. Prudhomme S, Oden J. On goal oriented error estimation for elliptic problems: application to the control of pointwise error. *Computers Method in Applied Mechanics and Engineering* 1999; **176**(1–4):313–331.
9. Larson MG, Barth TJ. A posteriori error estimation for discontinuous Galerkin approximation of hyperbolic systems. *NAS Technical Report NAS 99-010*, 1999.
10. Machiels L, Peraire J, Patera AT. A posteriori finite element output bounds for the incompressible Navier–Stokes equations. Application to a natural convection problem. *Journal of Computational Physics* 2001; **172**:401–425.
11. Hartmann R, Houston P. Adaptive discontinuous Galerkin methods for the compressible Euler equations. *Journal of Computational Physics* 2002; **182**(2):508–532.
12. Leicht T, Hartmann R. Error estimation and anisotropic mesh refinement for 3D laminar aerodynamic flow simulations. *Journal of Computational Physics* 2010; **229**:7344–7360.
13. Hartmann R, Held J, Leicht T. Adjoint-based error estimation and adaptive mesh refinement for the RANS and (k- ω) turbulence model equations. *Journal of Computational Physics* 2011; **230**:4268–4284.
14. Loseille A, Dervieux A, Alauzet F. Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations. *Journal of Computational Physics* 2010; **229**(2):2860–2897.
15. Venditti DA, Darmofal DL. Adjoint error estimation and grid adaptation for functional outputs: application to quasi-one-dimensional flow. *Journal of Computational Physics* 2000; **164**:204–227.
16. Venditti DA, Darmofal DL. Grid adaptation for functional outputs: application to two-dimensional inviscid flow. *Journal of Computational Physics* 2002; **176**:40–69.

17. Venditti DA, Darmofal DL. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flow. *Journal of Computational Physics* 2003; **187**:22–46.
18. Peter J, Nguyen-Dinh M, Trontin P. Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates – with applications to Euler flows. *Computers & Fluids* 2012; **66**:194–214.
19. Nguyen-Dinh M, Peter J, Sauvage R, et al. Mesh quality assessment based on aerodynamic functional output total derivatives. *European Journal of Mechanics B/Fluids* 2014; **45**:51–57.
20. Dwight RP. Goal-oriented mesh adaptation using a dissipation based error indicator. *International Journal for Numerical Methods in Fluids* 2007; **56**(8):1193–2000.
21. Dwight RP. Heuristic a posteriori estimation of error due to dissipation in finite volume schemes and application to mesh adaptation. *Journal of Computational Physics* 2008; **227**:2845–2863.
22. Jameson A, Schmidt W, Turkel E. Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time stepping schemes. In *AIAA Paper Series*. Paper, 1981; 1981-1259.
23. Nielsen E, Park M. Using an adjoint approach to eliminate mesh sensitivities in computational design. *American Institute Aeronaut. Astronaut* 2005; **44**:948–953.
24. Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *La Recherche Aéronautique* 1994; **1**:5–21.
25. Todarello G, Vonck F, Bourasseau S, et al. Finite-volume goal-oriented mesh adaptation for aerodynamics using functional derivative with respect to nodal coordinates. *Journal of Computational Physics* 2016; **313**:799–819.
26. Cambier L, Heib S, Plot S. The ONERA elsA CFD software: input from research and feedback from industry. *Mechanics & Industry* 2013; **14**:159–174.
27. Fillola G, Le Pape MC, Montagnac M. Numerical simulations around wing control surfaces. *Proceedings of 24th ICAS, 29 August - 3 September 2004*, Yokohama, Japan, 2004.
28. Salah El Din I, Peter J, Dumont A, Raud P, Ben Khelil S. PR ADER – Rapport d’avancement. *Technical Report RA 1/22243 DAAP*, ONERA, 2015.
29. Peter P, Renac F, Dumont A, Méheut M. Discrete adjoint method for shape optimization and mesh adaptation in the elsA code. Status and challenges. *50th 3AF International Conference on Applied Aerodynamics*, Toulouse, France, 2015.
30. Cook PH, McDonald MA, Firmin MCP. Aerofoil RAE 2822 – pressure distributions, boundary layer and wake measurements. *AGARD-AR-138* 1979.
31. Li D, Hartmann R. Adjoint-based airfoil optimization with discretization error control. *International Journal for Numerical Methods in Fluids* 2015; **77**:1–17.
32. Fraysse F, Valero E, Ponsin J. Comparison of mesh adaptation using the adjoint methodology and truncation error estimates. *AIAA JOURNAL* 2012; **50**(9):1920–1932.
33. Hartman R, Held JLT, Leicht T, Prill F. Error estimation and adaptive mesh refinement for aerodynamic flows. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design* 2010; **113**:339-353.
34. Vassberg JC, Jameson A. In pursuit of grid convergence, Part I: two-dimensional Euler solution. *AIAA paper series*, Paper 2009:2009-4114.
35. Schmitt V, Charpin A. Pressure distributions on the ONERA-M6-wing at transonic Mach numbers. Experimental data base for computer program assessment. *AGARD-R-138* 1979:179.
36. Mayeur J, Dumont A, Destarac D, Gleize V. Reynolds-averaged Navier–Stokes simulations on NACA0012 and ONERA-M6 wing with the ONERA elsA solver. *AIAA Journal* 2016; **54**:2671–2687.
37. Richardson L. The approximate arithmetical solution by finite difference problems involving differential equations, with an application to the stresses in a masonry dam. *Transactions of the Royal Society* 1910; **110**:307–357.