Unstructured h- and hp- adaptive strategies for discontinuous Galerkin methods based on a posteriori error estimation for compressible flows

Francesca Basile, Jean-Baptiste Chapelier, Marta de La Llave Plata, Romain Laraufie, Pascal Frey

Unstructured *h*- and *hp*- adaptive strategies for discontinuous Galerkin methods based on *a posteriori* error estimation for compressible flows.

Francesca Basile[a,c,d], Jean-Baptiste Chapelier[a], Marta de la Llave Plata[b], Romain Laraufie[c], Pascal Frey[d]

[a] *ONERA - Université Paris-Saclay, Châtillon, 92322, France*
[b] *ONERA - Université de Toulouse, Toulouse, 31055, France*
[c] *Airbus Operations SAS, Toulouse, 31060, France*
[d] *Sorbonne Université, Paris, 75005, France*

Abstract

In this paper, we present *h*- and *hp*-adaptative strategies suited for the discontinuous Galerkin formulation of the compressible laminar and Reynolds-averaged Navier-Stokes equations on unstructured grids, relying on a metric-based simplicial remeshing approach. An *a posteriori* error estimator, combining the measure of the energy associated with the highest-order modes and the inter-element jumps, is used to build both the metric field and the polynomial degree distribution map. The choice of refining either in h or p is driven by a smoothness indicator based on the decay of modal coefficients in each element. The performance of the developed adaptation algorithms is assessed for the 2D laminar viscous flow past a NACA0012 airfoil, and for the 3D laminar viscous flows past a sphere and past a delta wing. The adaptive *hp*-strategy is applied to a 3D turbulent jet issued from a nozzle. Finally, the gain in accuracy provided by the adaptive algorithms with respect to uniformly refined simulations, for a given number of degrees of freedom with polynomial degrees $p = 1, 2, 3$, is demonstrated.

1. Introduction

The efficient numerical solution of the Navier-Stokes and the Reynolds-averaged Navier-Stokes (RANS) partial differential equations has gained a key role in industrial applications. Fluid flow simulations often require the definition of a very large number of spatial degrees of freedom to capture accurately features stemming from complex physics. In this context, adaptive strategies that concentrate the degrees of freedom in regions of interest of the flow are of great interest for optimizing the cost and accuracy of the simulations. In particular the research community

has begun to focus on *hp*-adaptive methods using piecewise polynomial approximations, in which, one not only locally adapts the size h of the mesh, but also the degree of the polynomials, p, within the element. These adaptive methods offer better flexibility and efficiency than adaptive methods implementing only h-refinement or p-enrichment. In practice, adaptive methods lead to the concentration of the degrees of freedom (dofs) in regions of interest of the flow, in order to optimize both the computational cost and the accuracy of the simulations [1, 2, 3], guided by an error estimate. In particular, in these regions h-adaptation aims at decreasing the size of the elements while p-adaptation increases the polynomial degree of the approximation of the solution, and *hp*-adaptation combines both techniques.

In recent years, discontinuous Galerkin (DG) methods [4, 5] have become very popular for the solution of nonlinear convection dominated flow problems [6]. DG methods are high-order finite element discretizations based on the variational formulation of the governing equations and combine features from Finite Volume (FV) and Finite Element (FE) methods. In addition to their suitability to implement *hp*-adaptive methods, a number of aspects make DG methods particularly interesting, such as their high-order of accuracy achieved on arbitrary unstructured meshes, accurate description of curved boundaries and suitability to parallel computing thanks to a compact stencil. Moreover, efficient, local, simple and low-computational cost *a posteriori* jump and spectral error indicators [7, 8, 9, 10] can be exploited for mesh or polynomial adaptation.

As regards pure h-adaptation, two main approaches are identified. A first approach well suited for structured meshes involves the isotropic or anisotropic subdivision of elements supporting non-conforming and hanging nodes [11, 12, 13, 14]. A second approach is used for simplicial meshes and involves a metric-based global meshing. The use of a metric-based approach provides high flexibility in terms of prescribing a precise size to an element. This avoids the constraint on the element size, imposed by the element splitting history. Moreover, the metric-based remeshing strategy circumvents the complications associated with handling the hanging nodes generated by an element splitting-based strategy, which require a particular numerical treatment at the non-conforming interfaces of the elements. Such treatments can introduce numerical errors while not being readily available in standard flow solvers. Additionally, a number of remeshing software using a metric-based strategy, based on the Delaunay method [15, 16, 17, 18, 19, 20, 21, 22], are available, often

under open-source licensing. In metric-based mesh adaptation a new mesh is generated for the entire computational domain. The original mesh is then used to store the characteristics of the new mesh during regeneration. This new mesh is described using a Riemannian metric, based on the assumption that for an optimal mesh all edge lengths will have unit measure in the metric space. Both the isotropic [23, 24, 25] and the anisotropic [26, 22, 27, 28] variants have shown promising results in the literature.

Regarding remeshing strategies, several criteria that dictate the modifications to be applied to the mesh can be found in the literature. As regards anisotropic mesh adaptation, the most common strategy, typically used in FE/FV contexts, consists in computing a metric based on the Hessian matrix of the solution [29, 30, 28, 31]. In the context of DG methods, the Hessian-based metric has also been used by Remacle et al. in [32] and by Alauzet at al. [33] using a second-order discretization ($p = 1$) to prescribe an anisotropy to the element. However, this strategy is not easily extensible to higher orders, as the standard Hessian matrix approach found in the literature for low-order methods is based on a linear approximation of a scalar quantity. An extension to higher-order methods has been proposed for 2D by Dolejsi, Balan, Rangarajan et al. [34, 35, 36]. It relies on computing the direction of the maximum $(p + 1)^{th}$ reconstructed derivative of the solution. For 3D problems Coulard et al. [37] have developped an approach based on a strongly non-linear optimization problem needing the solution of an adjoint problem. Thus, we see that the 3D metric-based anisotropic mesh adaptation for high-order methods is today an open research topic. However, a simple approach not involving expensive complex optimization problems, which are generally not suitable for unsteady configurations, seems still out of reach in an industrial context.

For isotropic global-remeshing approaches, more easily adaptable to 3D, the metric field can be built by defining only one size per element, while anisotropic approaches would need an efficient approximation of a $(p+1)^{th}$ derivative tensor, which is not readlily accessible. Isotropic approaches are featured in the literature, mostly in the context of unsteady complex applications. Bernard et al. [24] used a dynamic mesh adaptation strategy for ocean modelling, solving two saddle point optimization problems that can be solved in a closed form for constant polynomial degree. The first minimizes the global error in the domain while keeping constant the number of elements. The

second is based on minimizing the number of elements while keeping the global error constant in the domain. In the context of Large-Eddy Simulations (LES), Daviller et al. [23] used a sensor on the time-averaged dissipation of kinetic energy, provided as field function to the remesher. Benard et al. [38] predicted the new mesh sizes to respect two criteria: one minimizes a quantity depending on the second derivatives of the time-averaged flow velocity and one ensures that a sufficient part of turbulent scales is explicitly resolved.

Regarding *hp*-adaptivity, the local error estimator is no longer sufficient to guide the adaptivity. Indeed, it indicates the elements that should be refined, but does not indicate whether it is better to refine the element by h or p. A method for making that choice is called an *hp*-decision strategy. Generally, this choice is made according to an estimate of the solution smoothness in an element. If the solution is sufficiently smooth, the adaptive algorithm opts for p-enrichment, while non-smooth zones are h-refined.

Several strategies have been proposed as regards *hp*-decision. In the context of octree-based mesh adaptation, Gui and Babuska [39], followed by [40] assessed the solution smoothness thanks to the ratio between error estimates based on p and $p-1$ approximations. Houston et al. in [41] developed an *hp*-algorithm based on the estimation of the local Sobolev regularity index of a given function by monitoring the decay rate of its Legendre expansion coefficients. These two approaches, however, can not be applied to $p=1$ computations because they are based on lower-order estimates. Mavriplis [7] determined if the solution is locally smooth by computing the decay rate of the Legendre expansion coefficients of the solution under the assumption that for non-smooth solutions, the discontinuities in the solutions deteriorate this decay rate. Leicht et al. [2] and Chalmers et al. [42] extended this approach to multiple dimensions. A sensor based on a measure of the inter-element jumps of quantities as density, pressure and velocities in the solution, was used by Burgess and Mavriplis in [43] and by Wang et al. in [44]. This latter proposed to couple the jump-based sensor with a criterion based on the ratio of the full and the truncated expansion of coefficients for a flow quantity to assess the smoothness of the solution. Ceze et al. [12] computed the mesh anisotropy and the approximation order distribution from the optimization of a merit function that incorporates both an output sensitivity and a measure of solution cost on the new mesh, without performing smoothness measurements. In [45], Mitchell and McCain

summarized several strategies proposed over the years for making this determination between h and p adaptation.

In the continuous mesh context, Dolejsi [34] presented a continuous mesh model in which a local interpolation error estimate guides the determination of the size and the shape of mesh elements, as well as the polynomial degree. He extended this concept to the corresponding continuous (interpolation) error model in [46] to create an optimal mesh given a constraint on the number of elements or on the global error. Based on this continuous model, Rangarajan et al. [36] and Balan et al. [35] combined this methodology to the adjoint-based error estimates to produce anisotropic adapted hp-meshes. The latter evaluates the smoothness of the solution relying on a sensor based on the inter-element jumps of the solution.

Our work is motivated by the research of a simple, easily implementable, mesh adaptation algorithm, which is expected to be rapidly converging towards the optimal mesh, in terms of both mesh elements and polynomial degree distributions. The final focus of this research is placed on flow configurations relevant to industry. For these, a minimal number of adaptation steps is sought since the size and complexity of the problems are significant, and thus the optimization of the simulation costs is essential. In particular, the final applications targeted in this research project involve hybrid RANS-LES simulations of turbulent jets, and in general 3D configurations with a high number of degrees of freedom. The possibility to extend the algorithm to 3D in a straightforward manner is, in this context, an essential requirement.

An isotropic mesh-refinement based strategy is selected, which is fit for adapting the resolution in free-shear regions, while the anisotropic flow regions (e.g. boundary layers), can be meshed with fixed anisotropic elements which resolution is modulated through p-adaptation. The extension to unsteady hp-adaptation for hybrid RANS-LES computations will be addressed in future research, in which a RANS adapted mesh can be used as the initial mesh for unsteady adaptations, as in an LES context an initial very coarse mesh is not fit to represent accurately turbulent eddies.

In the present paper, a metric based h-adaptive strategy is proposed, together with an extension to hp-adaption intended to improve the behaviour of pure h-adaptive algorithms, thanks to the introduction of a hp-decision strategy. An efficient *a posteriori* error estimator, which employs the solution itself to derive estimates of the discretization error, is used to control the element size for

h-adaptation, and both the element size and polynomial degree for hp-adaptation. As regards the hp-adaptation algorithm we propose that the choice on whether to h-refine or p-enrich an element be driven by the decay rate of the modal coefficients of the DG approximation, which characterizes the smoothness of the solution in the element. The MMG library, supporting both 2D and 3D remeshing, has been chosen in this work as remeshing tool. The prescribed sizes are given to MMG as an input size map, which in turn outputs the newly adapted mesh. The prescribed polynomial degrees are then interpolated onto the new mesh provided by MMG, as explained in the next sections. Finally the solution on the previous mesh is projected onto the newly adapted mesh.

The gain in terms of number of dofs provided by the adaptive algorithms with respect to uniformly refined meshes is evaluated by performing uniform and adaptive simulations of laminar flows. To complete the validation, we study a turbulent test case in the context of RANS simulations, by adapting the hp-adaptive algorithm to flows requiring particular treatments in the boundary layers. The present methodology also solves potential robustness issues associated with high orders of accuracy, due to the progressive increase of the polynomial degree p in the computational domain during the adaptation process.

These adaptation procedures are applied in the context of the unstructured compressible flow solver CODA, developed in partnership by Airbus, ONERA and DLR [47] and targets research and industrial aerodynamic problems of interest. This new CFD platform is designed for efficient parallel and heterogeneous architectures, applying modern software techniques to a wide range of multidisciplinary applications. The object-oriented CODA framework permits the integration of advanced inter-operable CFD components, including the implementation of two different flow solvers based on Finite Volume and discontinuous Galerkin schemes, applied to the Navier-Stokes and the RANS equations.

The paper is organized as follows. Section 2 presents the details of the discretization of the Navier–Stokes and the RANS equations using the discontinuous Galerkin method, Section 3 recalls the metric formalism and a short introduction to MMG, Section 4 provides a thorough description of the h- and hp- adaptive procedures, Section 5 presents the assessment of the implemented h- and hp-adaptation algorithms to three subsonic laminar configurations, namely the 2D flow past a NACA0012 airfoil, the 3D flow past a sphere and the 3D flow past a delta wing. In section 6, the hp-

adaptive algorithm is applied and validated on a turbulent configuration of an isothermal subsonic jet issued from a nozzle. Concluding remarks and directions for future research are discussed in section 7.

2. Discontinuous Galerkin discretization of the compressible Navier-Stokes equations

This section describes the model equations for solving compressible fluid flow problems as well as the discontinuous Galerkin discretization implemented in the flow solver CODA used in the present work.

2.1. The compressible Navier-Stokes equations

The motion of a compressible fluid in a three-dimensional domain $\Omega \in \mathbb{R}^3$ is described by the compressible Navier-Stokes (NS) equations, which take the form

$$\begin{aligned} \partial_t \mathbf{u} + \nabla \cdot (\mathbf{F}_c(\mathbf{u}) - \mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u})) = 0, \quad \forall \mathbf{x} \in \Omega, t > 0 \\ \mathbf{u}(\mathbf{x}) = \mathbf{u}_0, \quad \forall \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

with appropriate boundary conditions prescribed on $\partial\Omega$. The vector $\mathbf{u} = \left(\rho, \rho\mathbf{v}^{\mathrm{T}}, \rho E\right)$ represents the conservative variables, with ρ being the density, $\mathbf{v}$ being the velocity vector and ρE the specific total energy. $\mathbf{F}_c(\mathbf{u})$ and $\mathbf{F}_v(\mathbf{u}, \nabla\mathbf{u})$ are the convective and diffusive fluxes defined, respectively, as:

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho\mathbf{v}^{\mathrm{T}} \\ \rho E \end{bmatrix}, \ \mathbf{F}_c(\mathbf{u}) = \begin{bmatrix} \rho\mathbf{v}^{\mathrm{T}} \\ \rho\mathbf{v}\mathbf{v}^{\mathrm{T}} + p\mathbf{I} \\ \rho E\mathbf{v}^{\mathrm{T}} + p\mathbf{v}^{\mathrm{T}} \end{bmatrix} \text{ and } \mathbf{F}_v(\mathbf{u}, \nabla\mathbf{u}) = \begin{bmatrix} 0 \\ \boldsymbol{\tau} \\ \mathbf{v}^{\mathrm{T}}\boldsymbol{\tau} - \mathbf{q}^{\mathrm{T}} \end{bmatrix} \tag{2}$$

with:

$$\boldsymbol{\tau} = 2\mu\mathbf{S}^D = \mu(T)\left(\boldsymbol{\nabla}\mathbf{v} + (\boldsymbol{\nabla}\mathbf{v})^{\mathrm{T}} - \frac{2}{3}(\boldsymbol{\nabla}\cdot\mathbf{v})\mathbf{I}\right), \quad \mathbf{q} = -\lambda\boldsymbol{\nabla}T \tag{3}$$

$$\mathrm{p} = (\gamma - 1)(\rho E - \rho\mathbf{v}\cdot\mathbf{v}/2) \tag{4}$$

where p is the static pressure, $\gamma = \dfrac{C_p}{C_v}$ is the ratio of specific heats, $\mu(T)$ is the dynamic viscosity, $\mathbf{S}^D$ is the deviatoric component of the strain-rate tensor $\mathbf{S} = \dfrac{1}{2}(\boldsymbol{\nabla}\mathbf{v} + (\boldsymbol{\nabla}\mathbf{v})^{\mathrm{T}})$, T is the temperature, R is the specific gas constant, and $\lambda = \mu\dfrac{C_p}{Pr}$ is the thermal conductivity, with Pr the Prandtl number.

In this work, a value of $Pr = 0.72$ is used. The dependence of the viscosity on the temperature is expressed using the Sutherland's law.

2.2. The Reynolds-averaged Navier-Stokes equations with the Spalart-Allmaras turbulence model

In the Reynolds-averaged Navier-Stokes (RANS) formulation the mean flow equations are coupled with the one-equation turbulence model of Spalart-Allmaras (SA) [48] with the modification described in [49]. The Reynolds-averaged Navier-Stokes equations with the SA turbulence model can be written in conservative form by adding a source term to equation (1) as:

$$\begin{aligned} \partial_t \mathbf{u} + \nabla \cdot (\mathbf{F}_c(\mathbf{u}) - \mathbf{F}_v(\mathbf{u}, \nabla \mathbf{u})) = \mathbf{S}(\mathbf{u}, \nabla \mathbf{u}), \quad \forall \mathbf{x} \in \Omega, t > 0 \\ \mathbf{u}(\mathbf{x}) = \mathbf{u}_0, \quad \forall \mathbf{x} \in \Omega \end{aligned} \tag{5}$$

where we assume that $\mathbf{u}$ is the vector of time-averaged conservative variables over a given time interval and t is a pseudo time. The turbulent conservative variable $\rho\tilde{\nu}$ is added to the vector of conservative variables and the convective and diffusive fluxes $\mathbf{F}_c(\rho\tilde{\nu})$, $\mathbf{F}_v(\rho\tilde{\nu}, \nabla\rho\tilde{\nu})$ are defined as:

$$\mathbf{F}_c(\rho\tilde{\nu}) = \rho\tilde{\nu}\mathbf{v}^{\mathrm{T}}, \quad \mathbf{F}_v(\rho\tilde{\nu}) = \frac{1}{\sigma}(\mu + f_{n1}\rho\tilde{\nu})\boldsymbol{\nabla}\tilde{\nu}^{\mathrm{T}} \tag{6}$$

In the diffusive fluxes, the turbulent stress tensor $\boldsymbol{\tau}_\mathbf{t}$ and the turbulent heat fluxes $\mathbf{q}_\mathbf{t}$ are added respectively to $\boldsymbol{\tau}$ and $\mathbf{q}$ defined in equation (3):

$$\boldsymbol{\tau}_\mathbf{t} = 2\mu_t \mathbf{S}^D, \quad \mathbf{q}_\mathbf{t} = -\frac{\mu_t}{Pr_t} C_p \nabla T \tag{7}$$

where $Pr_t = 0.9$ is the turbulent Prandtl number and μ_t is the turbulent dynamic viscosity:

$$\mu_t = \begin{cases} \rho\tilde{\nu} f_{v1}(\chi) & \text{for } \tilde{\nu} \geq 0 \\ 0 & \text{for } \tilde{\nu} < 0 \end{cases}, \quad f_{v1}(\chi) = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\rho\tilde{\nu}}{\mu} \tag{8}$$

The source terms act on the conservation equation only for the turbulent variable $\rho\tilde{\nu}$ and read:

$$\mathbf{S}(\rho\tilde{\nu}, \nabla(\rho\tilde{\nu})) = -\rho(P - D) - \frac{c_{b2}}{\sigma}\rho\nabla\tilde{\nu} + \frac{1}{\sigma}(\nu + f_{n1}\tilde{\nu})\nabla\rho \cdot \nabla\tilde{\nu} \tag{9}$$

where the production and destruction terms, P and D, are defined by:

$$P = \begin{cases} c_{b1}(1 - f_{t2})\tilde{\omega}\tilde{\nu} & \text{for } \tilde{\nu} \geq 0 \\ c_{b1}(1 - c_{t3})\omega\tilde{\nu} & \text{for } \tilde{\nu} < 0 \end{cases} \qquad D = \begin{cases} \left(c_{w1} f_w - \frac{c_{b1}}{\kappa^2} f_{t2}\right)\left(\frac{\tilde{\nu}}{\mathrm{d}}\right)^2 & \text{for } \tilde{\nu} \geq 0 \\ -c_{w1}\left(\frac{\tilde{\nu}}{\mathrm{d}}\right)^2 & \text{for } \tilde{\nu} < 0 \end{cases} \tag{10}$$

and

$$f_{n1} = \begin{cases} 1 & \text{for } \tilde{\nu} \geq 0 \\ \dfrac{c_{n1} + \chi^3}{c_{n1} - \chi^3} & \text{for } \tilde{\nu} < 0 \end{cases}, \qquad f_{t2} = c_{t3} \exp(-c_{t4}\chi^2), \qquad f_w = g\left(\frac{1 + c_{w3}^6}{g^6 + c_{w3}^6}\right)^{\frac{1}{6}} \tag{11}$$

with

$$g = r + c_{w2}(r^6 - r), \qquad r = \min\left(r_{lim}, \frac{\tilde{\nu}}{\omega\kappa^2 \mathrm{d}^2}\right) \tag{12}$$

d is the distance to the nearest wall and ω the vorticity magnitude.

The modified vorticity magnitude $\tilde{\omega}$ is given by

$$\tilde{\omega} = \begin{cases} \omega + \overline{\omega} & \text{for } \overline{\omega} > -c_{v2}\omega \\ \omega + \dfrac{\omega(c_{v2}^2\omega + c_{v3}\overline{\omega})}{(c_{v3} - 2c_{v2})\omega - \overline{\omega}} & \text{for } \overline{\omega} < -c_{v2}\omega \end{cases} \tag{13}$$

where $\overline{\omega}$ and f_{v2} are given by

$$\overline{\omega} = \frac{\tilde{\nu}}{\kappa^2 \mathrm{d}^2} f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1 - f_{v1}} \tag{14}$$

For the sake of completeness, we give the values of the constants in the above expressions: $c_{v1} = 7.1$, $\sigma = 2/3$, $c_{b1} = 0.1355$, $c_{b2} = 0.622$, $\kappa = 0.41$, $c_{w2} = 0.3$, $c_{w3} = 2$, $r_{lim} = 10$, $c_{t3} = 1.2$, $c_{t4} = 0.5$, $c_{v2} = 0.7$, $c_{v3} = 0.9$, $c_{n1} = 16$.

2.3. DG Discretization

The DG discretization used in this work is based on a modal approach that relies on the use of a hierarchical and orthogonal polynomial basis for the Galerkin projection. Numerical integration on tetrahedra and triangles is efficiently performed by means of optimized quadrature rules proposed

by Witherden et al. [50]. Prismatic elements employ a combination of a Gaussian quadrature in the extrusion direction and the aforementioned optimized quadrature in the other two directions.

In this work steady solutions are obtained by employing an implicit backward-Euler scheme relying on the matrix-free GMRES block-Jacobi method with LU-Jacobi as preconditioner. The DG method implemented in the CODA solver is briefly outlined below.

We start by defining a shape-regular partition of the domain Ω, into a tessellation $\mathcal{T}_K$ of N non-overlapping and non-empty simplicial elements K of characteristic size h. We also define the sets $\mathcal{E}_i$ and $\mathcal{E}_b$ of interior and boundary faces in $\mathcal{T}_K$, such that $\mathcal{E}_h = \mathcal{E}_i \cup \mathcal{E}_b$.

Let $\mathcal{V}_h^p = \{\phi_h \in L^2(\Omega) : \phi \mid_K \in \mathcal{P}^p(K), \forall K \in \mathcal{T}_K\}$ be the functional space formed by piece-wise polynomials of total degree at most p, and $(\phi_K^1, ..., \phi_K^{N_p}) \in \mathcal{P}^p(K)$ a hierarchical and orthonormal basis of $\mathcal{V}_h^p$, of dimension N_p, confined to K. The solution in each element is thus expressed as

$$\mathbf{u}_h(\mathbf{x}, t) = \sum_{l=1}^{N_p} \phi_K^l(\mathbf{x}) \mathbf{u}_K^l(t), \forall \mathbf{x} \in K, \quad K \in \mathcal{T}_K, \forall t \geq 0 \tag{15}$$

The polynomial coefficients $(\mathbf{u}_K^l)_{1 \leq l \leq N_p}$ represent the degrees of freedom of the discrete problem in element K. The shape functions are polynomials that can be chosen arbitrarily. A methodology developed by Bassi et al. [51] consists in defining a starting set of monomial basis functions in each (arbitrarily shaped) element and applying a modified Gram-Schmidt orthonormalization procedure. The resulting basis yields a diagonal mass matrix in each element of the discretization, simplifying the resolution of the sets of equations in variational formulation. We denote by $\Phi_K = \{\phi_K^1, ..., \phi_K^{N_p}\}$ the generic basis of the polynomial space $\mathcal{P}^p(K)$. Φ_K^p is a hierarchical basis if it is contained in higher-order basis, that is $\Phi_K^p \subset \Phi_K^{p+1}$. We can rewrite equation (15) for an element K of degree p_K as:

$$\mathbf{u}_h|_K = \sum_{q=0}^{p_K} \sum_{l \in d_q} \phi_K^l \mathbf{u}_K^l, \forall \mathbf{x} \in K, \quad K \in \mathcal{T}_K \tag{16}$$

where $d_0 = \{1\}$ and $d_q = \left\{ l \in 2...N_p(K) \mid \phi_l \in \mathcal{P}_K^q \setminus \mathcal{P}_K^{q-1} \right\}$ is the set of indices of the basis functions of total degree at most q. The conservation law is discretized in physical space by using a discontinuous Galerkin method and the semi-discrete variational form of the system of equations

(1) thus reads: find $\mathbf{u}_h$ in $\mathcal{V}_h^p$ such that $\forall \phi_h \in \mathcal{V}_h^p$ we have

$$\int_{\mathcal{T}_K} \phi_h \partial_t \mathbf{u}_h dV + \boldsymbol{\mathcal{L}}_c(\mathbf{u}_h, \phi_h) + \boldsymbol{\mathcal{L}}_v(\mathbf{u}_h, \phi_h) = 0 \tag{17}$$

In equation (17) $\boldsymbol{\mathcal{L}}_c$ and $\boldsymbol{\mathcal{L}}_v$ represent the weak form of the convective and viscous terms respectively. The following notations are introduced: for a given interface e in $\mathcal{E}_i$ we define the average operator as $\{\{\mathbf{u}\}\} = (\mathbf{u}^+ + \mathbf{u}^-)/2$, the jump operator is defined as $[\![\mathbf{u}]\!] = \mathbf{u}^+ \otimes \mathbf{n} - \mathbf{u}^- \otimes \mathbf{n}$ where $\mathbf{u}^+$ and $\mathbf{u}^-$ are the traces of the variable $\mathbf{u}$ at the interface between elements K^+ and K^-. The DG discretization of the convective terms then reads

$$\boldsymbol{\mathcal{L}}_c(\mathbf{u}_h, \phi_h) = -\int_{\mathcal{T}_K} \mathbf{F}_c(\mathbf{u}_h) \cdot \nabla_h \phi_h dV + \int_{\mathcal{E}_i} [\![\phi_h]\!] \mathbf{h}_c(\mathbf{u}_h^+, \mathbf{u}_h^-, \mathbf{n}) dS + \int_{\mathcal{E}_b} \phi_h^+ \mathbf{F}_c(\mathbf{u}_b) \cdot \mathbf{n} dS \tag{18}$$

where the boundary values $\mathbf{u}_b = \mathbf{u}_b(\mathbf{u}_h^+, \mathbf{u}_{ext}, \mathbf{n})$, with $\mathbf{u}_{ext}$ a reference external state computed such that the boundary conditions are satisfied on $\mathcal{E}_b$. The numerical flux $\mathbf{h}_c$ is chosen such that it is consistent and conservative. In this work we use the Roe flux [52] with an entropy fix similar to that of Harten [53] for all simulations.

The discretization of the viscous terms is performed using the BR1 approach of Bassi and Rebay [54]. This approach relies on the definition of the conservative variable gradients as auxiliary variables $\boldsymbol{\sigma} = \nabla \mathbf{u}$ which verify the following equations:

$$\boldsymbol{\sigma} - \nabla \mathbf{u} = 0 \tag{19}$$

$$\partial_t \mathbf{u} + \nabla \cdot \mathbf{F}_c(\mathbf{u}) + \nabla \cdot \mathbf{F}_v(\mathbf{u}, \boldsymbol{\sigma}) = 0 \tag{20}$$

This leads to the introduction of the so called global lifting operator $\mathbf{L}_h$ such that:

$$\boldsymbol{\sigma}_h = \nabla_h \mathbf{u}_h + \mathbf{L}_h \tag{21}$$

and $\mathbf{L}_h$ satisfies the following condition

$$\int_{\mathcal{T}_K} \phi_h \mathbf{L}_h dV = -\int_{\mathcal{E}_i} \{\{\phi\}\} [\![\mathbf{u}_u]\!] dS - \int_{\mathcal{E}_b} \frac{\phi^+}{2} (\mathbf{u}_h^+ - \mathbf{u}_b) \otimes \mathbf{n} dS \tag{22}$$

The discrete variational form of the viscous term for the BR1 method therefore takes the form

$$\begin{aligned}\boldsymbol{\mathcal{L}}_v(\mathbf{u}_h,\phi_h) = &-\int_{\mathcal{T}_K}\mathbf{F}_v(\mathbf{u}_h,\nabla_h\mathbf{u}_h+\mathbf{L}_h)\cdot\nabla_h\phi_h dV-\int_{\mathcal{E}_i}[\![\phi_h]\!]\{\{\mathbf{F}_v(\mathbf{u}_h,\nabla_h\mathbf{u}_h+\mathbf{L}_h)\}\}\cdot\mathbf{n}dS\\ &-\int_{\mathcal{E}_b}\phi_h^+\mathbf{F}_v(\mathbf{u}_b,\nabla\mathbf{u}_b+\mathbf{L}_h)\cdot\mathbf{n}dS\end{aligned} \tag{23}$$

3. Continuous mesh framework

We recall in this section some theoretical notions underlying the metric-based mesh adaptation considered in the present study.

3.1. Basic notions of metric-based mesh generation

The mesh is modeled as a continuous medium: at the continuous level, we consider mesh elements being represented by ellipsoids (ellipses in 2D). In this geometric representation, the size of the element is its volume (area in 2D), its shape is associated with the ratio of the lengths of its semi-axes and its orientation is provided by its principal axis vectors [55]. Therefore, the control of the element size can be achieved by specifying a metric tensor $\mathcal{M}(\mathbf{x})$ to prescribe the size, shape and orientation of mesh elements over the whole domain. $\mathcal{M}(\mathbf{x})$ is a $d\times d$ symmetric positive definite matrix, with d the number of dimensions of the problem. This metric tensor guides the generation of a quasi-uniform mesh of Ω in the metric $\mathcal{M}$.

In the continuous framework a metric tensor $\mathcal{M}$ is a continuous element [26], that can be geometrically represented by its unit ball. A discrete element K is said to be unit with respect to a continuous element $\mathcal{M}$ if the length of all its edges is unit in the metric $\mathcal{M}$. The standard Euclidean scalar product is then modified using a proper metric tensor field, in order to prescribe a unit edge length

$$l_{\mathcal{M}}(\mathbf{e}) = ||\mathbf{e}||_{\mathcal{M}} = \sqrt{{}^{\mathrm{T}}\mathbf{e}\mathcal{M}\mathbf{e}} = 1 \tag{24}$$

where $l_{\mathcal{M}}(\mathbf{e})$ is the distance between the extremities of the vector $\mathbf{e}$. The set of points satisfying the relation above describes an ellipsoid in three dimensions (ellipse in two dimensions), for which the radius of each axis is given by the square root of the inverse of the corresponding eigenvalues of $\mathcal{M}$. As the metric is not constant in an element but varies across the domain, after parametrizing

the curve $\mathbf{e}(s) : [0,1] \to \mathbb{R}^d$ the average length of a mesh edge $\mathbf{e}$ is introduced:

$$l_{\mathcal{M}}(\mathbf{e}) = \int_0^1 \sqrt{{}^{\mathrm{T}}\mathbf{e}\mathcal{M}(s)\mathbf{e}}\, ds \tag{25}$$

The desired adapted mesh is then a unit mesh, i.e., a mesh such that for each edge $l_{\mathcal{M}}(\mathbf{e}) \simeq 1$. In practice, the remesher builds a mesh such that the edges lengths are close to 1. In particular every edge of the mesh l verifies: $\frac{1}{\sqrt{2}} < l < \sqrt{2}$.

In order to deal with a single metric at the vertices, metric intersection and interpolation procedures are then employed. For a more complete treatment of these specific subjects the reader should refer to [29, 55].

The metric tensor can be diagonalized and decomposed as:

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})\,\Lambda(\mathbf{x})\,\mathcal{R}^{\mathrm{T}}(\mathbf{x}) \tag{26}$$

where $\mathcal{R}$ is the square matrix whose i-th column is the eigenvector $(\mathbf{v}_i)_{i=1,..,d}$ of $\mathcal{M}$ and Λ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues $\Lambda_{ii} = \lambda_i$. The matrix Λ thus prescribes the size and shape of the element, while the matrix $\mathcal{R}$ prescribes its orientation.

The element sizes are then linked to the eigenvalues by $h_i = 1/\sqrt{\lambda_i}$ with $i = 1, ..., d$. The metric tensor can be rewritten as:

$$\mathcal{M}(\mathbf{x}) = \mathcal{D}^{2/n}(\mathbf{x})\mathcal{R}(\mathbf{x})\,\zeta(\mathbf{x})\,\mathcal{R}^{\mathrm{T}}(\mathbf{x}) \tag{27}$$

where the density function is defined as $\mathcal{D}(\mathbf{x}) = \sqrt{\det\mathcal{M}(\mathbf{x})} = \left(\Pi_j^d \lambda_j\right)^{1/2} = \left(\Pi_j^d h_j\right)^{-1}$. The anisotropic quotients tensor $\zeta(\mathbf{x})$ is defined as a diagonal matrix with the d anisotropic quotients equal to $r_k(\mathbf{x}) = h_k^d \left(\Pi_j^d h_j\right)^{-1}$ as diagonal entries.

For an isotropic metric the semi-axes of the ellipsoid (ellipse in 2D) have the same length, and the d eigenvalues associated with the ellipsoid coincide, i.e. $h_1 = h_2 = h_3 = h_n$, the *j-th* eigenvectors are simply the *j-th* coordinate vectors, and the anisotropic quotients are all unitary. We obtain then the sphere (and the circle in 2D) of radius 1.

For a thorough review about metric-based theory, the reader can refer to [56, 29, 57, 26].

3.2. MMG library

MMG is an open source software for simplicial remeshing [58, 22], supporting both 2D and 3D remeshing (MMG2D and MMG3D). The MMG library has been already used extensively in the context of FV and FE methods [23, 38, 59, 60]. We provide a short description of the MMG mesh adaptation and optimization algorithm. MMG's goal is to remesh $\mathcal{T}_{K,i}$, where i is the index corresponding to the mesh adaptation step, into a new mesh $\mathcal{T}_{K,i+1}$, which is a close approximation of the computational domain Ω, adapted to a specified local size feature. From the user point of view, MMG takes as inputs a mesh $\mathcal{T}_{K,i}$ and a discrete metric map $\mathcal{M}^*_{i+1}$ defined at each vertex. We employ the superscript * to remind that the desired metric field $\mathcal{M}^*_{i+1}$ is imposed by the user on the mesh $\mathcal{T}_{K,i}$, even though the real metric associated with the mesh $\mathcal{T}_{K,i+1}$ is $\mathcal{M}_{i+1}$, which is different due to quality constraints. These constraints need to be satisfied by MMG, in addition to reproducing the prescribed metric field. MMG defines a continuous metric field using interpolation schemes, local modifications and quality checks. It then outputs the mesh $\mathcal{T}_{K,i+1}$ with a corresponding metric field $\mathcal{M}_{i+1}$, which is an approximation of the desired metric $\mathcal{M}^*_{i+1}$.

The approach MMG uses for three-dimensional domain remeshing is a local, iterative remeshing procedure. It consists in conducting local modification operations both on the surface and volume parts, which affect very limited areas of the meshes at hand, and a sequence of meshes is produced, until it converges toward the final optimal mesh $\mathcal{T}_{K+1}$. Among the local mesh modifications that the software performs inside the volume there are edge split, edge collapse, edge swap and node relocation.

Regarding the handling of boundaries, MMG locally reconstructs the "ideal" surface from the discrete geometry of the input mesh, using cubic Bézier triangles. The Hausdorff distance, defined as the distance between the ideal and the discrete mesh surface is controlled in order to ensure a good boundary approximation.

The parameters needed by MMG to perform the adaptation are listed below :

- h_{max}: the maximum allowed size of an edge in $\mathcal{T}_{K,i+1}$;
- h_{min}: the minimum allowed size of an edge in $\mathcal{T}_{K,i+1}$;
- h_{grad}: the gradation value which controls the ratio between two adjacent edges (with a grada-

tion of h_{grad}, two adjacent edges h_1 and h_2 in $\mathcal{T}_{K,i+1}$ must respect that $\frac{1}{h_{grad}} \leq \frac{h_1}{h_2} \leq h_{grad}$;

- *hausd*: controls the geometric approximation of the boundaries $\partial\mathcal{T}_{K,i+1}$. It imposes the maximal distance between the piecewise linear representation of the boundary and the reconstructed ideal boundary.

The parameters h_{max}, h_{min} and *hausd* depend on the length scale of the problem, while h_{grad} controls the quality of the mesh. The lower the gradation, the better the mesh quality. However, higher gradation values impose less constraints, and allow for a better match between the prescribed metric $\mathcal{M}^*_{i+1}$ and the actual metric $\mathcal{M}_i$ computed by MMG, without introducing extra elements.

A more detailed explanation of MMG features can be found in [22].

4. *h*- and *hp*- *adaptative algorithms*

In the present section, we describe in detail the adaptation strategies used in this research. It is based on the coupling of accurate DG-based indicators extracted from the DG flow solver, with the external remeshing library. At the end of the section we propose a summary of the two adaptation algorithms. The adaptation procedure is driven by an *a posteriori* error estimator which controls the solution accuracy within the domain, identifying the regions lacking the requested resolution. The resolution in these regions is improved by either decreasing the size of the element or increasing the polynomial degree which approximates the solution. A smoothness indicator based on the spectral decay of the coefficients of the Legendre polynomials guides the *hp*-decision, leading to p-enrichment for smooth regions and h-refinement for non-smooth regions. A metric-based *hp*-mesh combines continuous and discrete settings, to prescribe respectively to the nodes and to the elements of the mesh: the geometry of elements changes smoothly, and in particular the remesher needs a nodal size map, while the approximation polynomial degree is a discrete quantity unequivocally defined for each element. The following sections describe out strategies to create an *hp*-mesh in the context of metric-based mesh adaptation, without using optimization algorithms. We briefly summarize the two procedures in algorithms 1 and 2.

4.1. Error estimator

The first step in the construction of the present *hp*-adaptation strategy is to devise an accurate and simple error estimator suited to various polynomial degrees.

The interest of coupling two different indicators has been emphasized in previous research by Colombo et al. [61] and Bassi et al. [62]. They observed by numerical experiments that an error estimator based on the jumps at the interfaces of the elements is reliable for any polynomial degree, whereas an error indicator based on the high-order modes (hereafter termed SD for spectral decay) is reliable only for high polynomial degrees. Indeed, the SD estimator is accurate but can yield strong cell-to-cell variations. Hence the interest of coupling such indicator with a jump error estimator that identifies a larger region for adaptation and smooths out the overall indicator map. According to these observations, they implemented a combination of the two indicators, after normalization over the minimum and maximum values.

We follow the same idea by combining two error estimators, one based on the energy of the highest-order modes, the so-called Small Scale Energy Density (SSED) estimator ϵ_{SSED}, and the other one based on the jumps across element interfaces, ϵ_{JUMP}. In our work the solution component on which we estimate the error is the norm of the momentum, while Colombo and Bassi employed the pressure. Moreover the error estimators we use in this work are not normalized respectively by the sum of the energy of all modes and the norm of the solution component itself, but have the dimensions of an energy $[\rho\mathbf{u}^2]$, based on the work of Naddei in [63], who observed better performances of the SSED estimator with respect to its "normalized version", the SD. In the context of our research, this formulation showed better results, especially in recirculation zones where a low value of momentum can bias the value of the momentum-based normalized error estimators [64].

The SSED is based on a measure of the error $\mathbf{E}_K$ of the numerical solution $\mathbf{u}$ with respect to the exact solution $\mathbf{u}_{ex}$ in the element by computing the norm of the difference between the numerical solution $\mathbf{u}_{h,p}$ and the projection of the numerical solution on the reduced-order space $\mathcal{V}_h^{p-1}$, namely $\mathbf{u}_{h,p-1}$:

$$\mathbf{E}_K = ||\mathbf{u}_h - \mathbf{u}_{ex}|| \simeq ||\mathbf{u}_{h,p} - \mathbf{u}_{h,p-1}|| \tag{28}$$

The error based on the norm of the high-order modes of the momentum vector is normalized by the volume of the element to better perform on meshes with large variations in element sizes. The final formulation reads:

$$\epsilon^2_{\text{SSED},K} = \frac{\int_K ||(\rho\mathbf{v})_{h,p} - (\rho\mathbf{v})_{h,p-1}||^2 dV}{|K|} = \frac{||(\rho\mathbf{v})_{h,p} - (\rho\mathbf{v})_{h,p-1}||^2_{L^2(K)}}{|K|} \tag{29}$$

where $(\rho\mathbf{v})_h$ is the momentum vector and $|K|$ is the volume of the element.

The second error estimator is a modified version of the one proposed by Bernard et al. in [24]. The average value of the conservative variables at a given interface e in $\mathcal{E}_i$ is used to approximate the exact solution $\mathbf{u}^+_{ex}$:

$$\mathbf{u}^+_{ex} \simeq \frac{1}{2}(\mathbf{u}^+_h + \mathbf{u}^-_h) \tag{30}$$

where $\mathbf{u}^+_h$ and $\mathbf{u}^-_h$ are respectively the traces of the variable $\mathbf{u}$ at the interface between elements K^+ and K^-. We then define the error $\mathbf{E}_e$ on the interface e (which is a face of the element for 3D configurations, and an edge of the element for 2D configurations) as half the jump of the variable traces across the interfaces:

$$\mathbf{E}_e = ||\mathbf{u}^+_h - \mathbf{u}^+_{ex}|| \simeq \frac{1}{2}||\mathbf{u}^+_h - \mathbf{u}^-_h|| \tag{31}$$

and we consider an averaged error over each interface e, for the norm of the momentum vector

$$\epsilon^2_{\text{JUMP},e} = \frac{\int_{\partial K_e} ||(\rho\mathbf{v})^+_h - (\rho\mathbf{v})^-_h||^2 dS}{4|\partial K_e|} = \frac{||(\rho\mathbf{v})^+_h - (\rho\mathbf{v})^-_h||^2_{L^2(\partial K_e)}}{4|\partial K_e|} \tag{32}$$

For consistency with the dimensions of the SSED indicator in equation (29), the error indicator here is normalized by the area of each interface $|\partial K_e|$ (or for 2D problems the length of the edge e). For each element K we compute the error estimator by the following the simple rule:

$$\epsilon^2_{\text{JUMP},K} = \frac{1}{N_e}\sum_{e=1}^{N_e} \epsilon^2_{\text{JUMP},e} \tag{33}$$

with N_e the number of faces of the element in 3D (or edges in 2D).

Both indicators are normalized by their respective maximum and minimum values over the

whole domain $\mathcal{T}_K$(min-max normalization) before the coupling:

$$\epsilon_K = \epsilon_{\text{SSED},K,norm} + \epsilon_{\text{JUMP},K,norm} \tag{34}$$

$$= \frac{\epsilon_{\text{SSED},K} - \min(\epsilon_{\text{SSED}})\mathcal{T}_K}{\max(\epsilon_{\text{SSED}})\mathcal{T}_K - \min(\epsilon_{\text{SSED}})\mathcal{T}_K} + \frac{\epsilon_{\text{JUMP},K} - \min(\epsilon_{\text{JUMP}})\mathcal{T}_K}{\max(\epsilon_{\text{JUMP}})\mathcal{T}_K - \min(\epsilon_{\text{JUMP}})\mathcal{T}_K} \tag{35}$$

In references [61, 62] which focus on p-adaptation and our previous work [64] focused on h-adaptation, the contribution of the error estimator based on the solution jump decreased as the polynomial degree increased and the contribution of the modal estimator was switched off for $p = 1$. In this paper we prefer a uniform approach to combine the two error estimators for all polynomial degrees. We observed that even though the estimator based on the highest order modes is not always optimal for $p = 1$ computations, its coupling with a jump error estimator still performs well for low-degree polynomials.

4.2. Smoothness indicator

The difficulty in hp-adaptive methods lies in the choice whether to adapt an element with h-refinement or p-enrichment. In this work we follow the approach proposed in 1D by Mavriplis [7]. It is based on the assumption that the rate of decay of the spectrum of the DG modal coefficients is related to the convergence rate of the solution. This information is exploited to evaluate the smoothness of the solution in the mesh elements. It is assumed that for a 1D Legendre expansion with coefficients $a^{(q)}$ with $q = 0, ..., p_K$, the modal coefficients decay exponentially fast after the asymptotic range has been reached:

$$|a^{(q)}| \simeq C \exp(-\sigma q) \tag{36}$$

where C and σ are constants determined by a least-squares best fit of $\log(a^{(q)})$ vs. q. The decay coefficient σ is then used as smoothness indicator. The reliability of the smoothness estimate increases with the polynomial degree. High decay rates imply that the solution is smooth, while the solution deviates from analytical behaviour in presence of low decay rates. In this work, following Mavriplis' choice, we use a threshold value of 1: elements with $\sigma_K > 1$ are assumed to be smooth, and subject to p-enrichment; if this condition is not fulfilled, then h-refinement is used.

For 2D and 3D computations, several modal coefficients can contribute to an index q. Therefore we need to gather in some way the coefficients of the modes to retrieve one single value per index q. Based on Mavriplis' work [7], the extension to 2D has been performed by Chalmers et al. [42], who computed a one-dimensional smoothness indicator for each mode, and used the minimum value for σ. Leicht et al. in [2] proposed to extend the approach in 3D through the accumulation of all coefficients of the Legendre polynomials of the corresponding multi-dimensional degree. They tested the classical threshold between smooth and non-smooth element $\sigma = 1$ as well as different threshold values.

The approach we follow in our work to retrieve one single coefficient for indices q for 2D and 3D configurations consists in computing the coefficient $a^{(q)}$ as the L^2-norm of $u^{(q)}$ of the coefficients of the polynomial basis as:

$$a_K^{(q)} = \sqrt{\sum_{l \in d_q} {u_K^{(l)}}^2} \quad \forall q \in (1, p_K) \tag{37}$$

with d_q the set of indices of the basis functions of total degree at most q (introduced in equation (16)). We notice that the coefficient associated with $q = 0$ corresponds to the average of the solution over the element, which can severely bias the decay rate. Therefore we chose not to employ the coefficient $a^{(0)}$ in the log-linear regression $\log(a^{(q)})$ vs. q. As a result, $p = 1$ computations lack information about the smoothness of the solution, and the smoothness indicator can be computed only for high-order elements, i.e. $p \geq 2$. In this work $p = 1$ elements are always fictitiously marked as smooth elements, always requiring p-adaptation. A different strategy, employing smoothness indicators better suited for shock capturing, will be used in future work for test cases with strong physical discontinuities, where a very low polynomial degree is in these regions. The decay rate is computed from the modal coefficients associated with the norm of the momentum. The smoothness indicator can also serve as an indicator of the convergence rate of the solution, which will be exploited in the mesh refinement strategy described below.

4.3. Metric prescription for h-adaptation

At the end of the simulation i, the error estimator in each mesh element is computed from the solution as $\epsilon_{K,i}$. The updated element size $h^*_{K,i+1}$ is then imposed for each mesh element (K, i), similarly to the approach proposed by Bernard et al. [24] and by Remacle for finite elements

[65]. We follow the assumption that the local error converges asymptotically to zero at a certain convergence rate. In particular, for a DG method the L^2-norm of the error in the solution decays asymptotically as $\mathcal{O}(h^{p+1})$ [5].

In a pure h-adaptive context, under the assumption of asymptotic regular behaviour, the error for one element K can be thus seen as:

$$\epsilon_K \simeq \mathcal{C} h_K^{\min(k,p+1)}, \qquad \forall K \in \mathcal{T}_{K,i} \tag{38}$$

with $\mathcal{C}$ depending only on the solution, for a fixed polynomial degree, and the Sobolev index k depending on the regularity of the solution. In the presence of a physical or geometrical discontinuity, the smooth convergence order $p+1$ is lost. Houston et al. [41] approximated the Sobolev index k solution to estimate the regularity of the solution. Remacle et al. [66], define *a priori* a given resolution to resolve the discontinuities, by prescribing a fixed small desired size to elements presenting non-smooth behaviour. In our work we exploit the fact that the convergence order of the solution is limited in the vicinity of discontinuities to $\epsilon_K \simeq O(1)$. Therefore we impose the convergence rate $m = p+1$ if the solution is smooth, and $m = 1$ if the solution is non-smooth. For $p = 1$ computations for which the smoothness indicator cannot be computed, the convergence rate is always set to $m = p+1$.

We can then define a size field on Ω: for each element K of the mesh $\mathcal{T}_{K,i}$, a new size $h_{K,i+1}$ is defined as:

$$h^*_{K,i+1} = h_K \left(\frac{\epsilon^*_{i+1}}{\epsilon_{K,i}} \right)^{\frac{1}{m_K}}, \qquad \forall K \in \mathcal{T}_{K,i} \tag{39}$$

where $h^*_{K,i}$ is the characteristic size of the element K at the adaptation step i, $\epsilon_{K,i}$ is the error estimator computed on the element K at i, m_K is the imposed convergence rate on the element K, and ϵ^*_{i+1} is the globally imposed target error of the new mesh $i+1$, which will be explained below.

Despite the fact that DG methods are element-based methods, where the characteristic size associated with the element K has a more natural definition, the input size-field that MMG requires is node-based. Therefore a volume-weighted average of the element-based values of the error estimator $\epsilon_{K,i}$ is computed based on its values in the elements surrounding each node. This node-based error estimator will be denoted as $\epsilon_{n,i}$. The first step consists in obtaining from the initial mesh

the metric $\mathcal{M}_0$, which is calculated by MMG from a preliminary remeshing step, and the initial characteristic lenghts h_0. Adapting the equation (39), the size h_n^* at mesh nodes reads

$$h_{n,i+1}^* = h_{n,i} \left(\frac{\epsilon_{i+1}^*}{\epsilon_{n,i}} \right)^{\frac{1}{m_n}} \qquad \forall n \in \mathcal{T}_{K,i} \tag{40}$$

where subscript n refers to quantities defined at nodes. The element-wise smoothness indicator σ_K is computed first, then the nodal value σ_n is obtained by the means of a weighted average. When the solution is considered smooth, i.e. $\sigma_n > 1$ we choose $m_n = p + 1$. For the rest of the nodes corresponding to non-smooth regions, the value $m_n = 1$ is chosen.

In an isotropic context h_n measures the length of the edge of the equilateral tetrahedra surrounding the node n in 3D (triangle for 2D). In practical terms MMG provides as output of the remeshing process a non-scalar matrix, which means that it introduces some anisotropy. In this case the eigenvalues associated with the metric are not identical. In order to make the output compatible with our isotropic strategy, we extract the characteristic length h_n from MMG by recalling the notions of density with respect to the metric tensor $\mathcal{M}$ introduced in section 3.1. The characteristic length is then considered equal to the cubic root (square root in 2D) of the mesh density in the metric $\mathcal{M}$ at the node n

$$h_n = \mathcal{D}_n^{-1/d} = (\Pi_{j=1}^d h_j)^{1/d} = (\Pi_{j=1}^d \lambda_j)^{-1/(2d)} \tag{41}$$

where d is the number of dimensions and λ_j are the eigenvalues associated with the metric tensor output of MMG, obtained by eigen-decomposition.

$\epsilon_{n,i}$ is the volume-weighted error estimator at the node n and ϵ_{i+1}^* the target error estimator we impose to all the mesh nodes as described below. A user-defined parameter, the maximum refinement factor r_h is needed to avoid uncontrolled or unwanted size changes between two subsequent adaptation steps. The value of $1/r_h$ represents the minimum allowed ratio $h_{n,i+1}^*/h_{n,i}$ such that we have $h_{n,i+1}^* \geq h_{n,i}/r_h$, i.e. in one adaptation step the edge can decrease its size by at most r_h times. Adopting this strategy, the mesh regions presenting a value of the error estimator lower than or equal to the target error are not refined, while the regions with the highest values of error estimators are maximally refined. Finally the regions with intermediate values are gradually refined

of a factor depending on the ratio between the target error and the value of the error estimator.

The input size-field is provided to MMG for each node of the mesh i as a diagonal matrix $\mathcal{M}^*_{i+1} = \Lambda^*_{i+1} = (h^*_{n,i})^{-2}\mathcal{I}$, where $\mathcal{I}$ is the identity matrix. This input field allows MMG to perform the refinement process which is seamless for the user.

Once the new mesh has been output by MMG, the solution from the previous mesh is projected onto the adapted mesh by means of an L_2 projection.

4.4. Degree and metric prescription for hp-adaptation

Similarly to h-adaptation, at the end of the simulation i, we compute the error estimator as well as the smoothness indicator from the solution in each mesh element as $\epsilon_{K,i}$. Then both the updated polynomial degree $p_{K,i+1}$ and the new element size $h_{K,i+1}$ are imposed for each mesh element K, as described below. We start by a constant polynomial degree map $p = p_{min}, \ \forall K \in \mathcal{T}_K$. We can define three non overlapping sets of elements of the mesh

$$\mathcal{T}_{K,i} = \mathcal{T}_{h,i} \cup \mathcal{T}_{p,i} \cup \mathcal{T}_{=,i} \tag{42}$$

where $\mathcal{T}_{h,i}$ is the subset of the elements marked for h-adaptation,$\mathcal{T}_{p,i}$ is the subset of the elements marked for p-adaptation and $\mathcal{T}_{=,i}$ is the subset of the elements not marked for any adaptation. An element cannot be selected for both h- and p-adaptation.

Firstly the smooth elements, i.e. the elements with a smoothness indicator $\sigma_K > 1$, and the $p = 1$ elements, which lack in information about their smoothness, are marked for p-adaptation if they present an error estimator $\epsilon_{K,i}$ greater than a target error ϵ^*_{i+1}, and their polynomial degree is increased by one. A threshold on the maximum polynomial degree $p_{\max}$ that the algorithm can address is defined by the user. After the threshold is attained, h-adaptation is performed also on elements presenting a smooth solution. Thus, for smooth and $p = 1$ elements lacking resolution, the new polynomial degree is defined as:

$$p^*_{K,i+1} = \max(p_{K,i} + 1, p_{\max}), \qquad \forall K \in \mathcal{T}_{p,i} \tag{43}$$

We use the superscript * also for the polynomial degree to remind that this corresponds to the

desired polynomial degree on the mesh $i+1$, but is imposed on the mesh i. The degree map is not going to be exactly preserved in each location of the the mesh $i+1$, due to the interpolation of the polynomial degree from one mesh to another.

Now that elements subject to p-refinement have been chosen for the subset of the mesh $\mathcal{T}_{p,i}$

$$\mathcal{T}_{p,i} = \{K \in \mathcal{T}_i \mid \epsilon_{K,i} > \epsilon^*_{K,i+1} \text{ and } [(\sigma_{K,i} > 1 \text{ and } p_{K,i} + 1 \leq p_{\max}) \text{ or } p_{K,i} = 1]\} \tag{44}$$

we still need to address the remaining elements $K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i}$. Similarly to equation (38), the error in an element K at the adaptation step i verifies:

$$\epsilon_K \simeq \mathcal{C} h_K^{m_K}, \qquad \forall K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i} \tag{45}$$

Assuming that $\mathcal{C}$ remains constant for non p-refined elements $K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i}$ between iterations i and $i+1$, we can use the formula (39) to compute the new sizes for elements $K \in \mathcal{T}_{h,i}$. This allows for h-refining the elements belonging to the subset $\mathcal{T}_{h,i}$:

$$\mathcal{T}_{h,i} = \{K \in \mathcal{T}_i \mid \epsilon_{K,i} > \epsilon^*_{K,i+1} \text{ and } [\sigma_{K,i} \leq 1 \text{ or } (\sigma_{K,i} > 1 \text{ and } p_{K,i}+1 > p_{\max})] \text{ and } p_{K,i} \neq 1\} \tag{46}$$

To be consistent with the node-based input size-field required by MMG, the quantities $\epsilon_{n,i}$, $p_{n,i}$ and $\sigma_{n,i}$ are computed at nodes using weighted average with contributions only from surrounding elements that are not marked for p-enrichment at the current step. This choice is illustrated in figure 1. This prevents an excessive refinement in p-refined zones, and keeps valid the hypothesis of a constant $\mathcal{C}$ between two adaptive steps in equation (45). This selective weighted-average allows for an hp-choice despite the fact that p is evaluated for each element and h is given to the nodes.

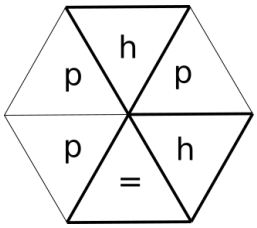


Figure 1: Simplified central node with surrounding simplicial elements (marked for p-adaptation, for h-adaptation and not marked for adaptation). The nodal weighted average of the error estimator, the polynomial degree and the smoothness indicator for the prescription of the sizes is performed only on highlighted elements.

The formula in equation (39) is then evaluated for all the nodes, and the size h_n^* is imposed at the node n as done in equation (40), but applied to the subset $\forall n \in \mathcal{T}_{h,i}$. The theoretical convergence order m_n is $p_n + 1$ if $\sigma_n > 1$ and 1 if $\sigma_n \leq 1$.

It should be noted that the nodal polynomial degree p_n is no longer an integer after the nodal average. Again, $p = 1$ elements are considered as smooth and the prescribed convergence order is $m_n = p_n + 1$.

The regions presenting a higher error estimator value than the target, with smooth solution or $p = 1$ approximation, are p-enriched, while the regions with non-smooth solution or already at their maximum polynomial degree are subject to h-refinement, again when the error is higher than the target.

The input size-field $h^*_{n,i+1}$ is provided to MMG for each node of the mesh i analogously to what is done for h-adaptation. MMG provides the new adapted mesh $i+1$ and the new polynomial degree map $p^*_{K,i+1}$ is interpolated from mesh $\mathcal{T}_{K,i}$ to mesh $\mathcal{T}_{K,i+1}$ with a nearest-neighbour interpolation: the new degree prescribed to an element of the old mesh is assigned to all the elements of the new mesh presenting their centers inside the old cell.

Algorithm 1: h-adaptation algorithm

```
set i=0;
create the initial mesh T_0;
obtain from MMG the initial characteristic lengths field h_0;
while NDOF < NDOF_MAX do
    perform the simulation on T_i;
    compute the a posteriori error estimator ε_{K,i} on T_i;
    choose the target error ε*_{i+1} in order to fulfill the given refinement criterion;
    average the error estimator at the nodes ε_{n,i} (volume-weighted average);
    average the smoothness indicator at the nodes σ_{n,i} (volume-weighted average);
    if ε_{n,i} > ε*_{i+1} then
        adapt the current size h_{n,i} to the new h*_{n,i+1} provided by the formulation (40);
    else
        leave the current size h*_{n,i+1} = h_{n,i} unmodified;
    end
    give the new metric to MMG, which generates the refined mesh T_{K,i+1};
    project the solution of T_{K,i} on T_{K,i+1};
    i+=1;
end
```

4.5. Choice of the target error

To complete the adaptation algorithm, we need to define a target error $\epsilon^*_{n,i+1}$ at all mesh nodes. This is given by a revisited fix-fraction marking strategy, which allows for a rapid convergence of the mesh adaptation algorithm towards the optimal mesh. The classical version of the fix-fraction marking strategy controlling the percentage of the mesh to refine (and often the percentage to coarsen) shows poor results in this context. This is due to the fact that the prescribed decrease of sizes here directly depends on the magnitude of the error estimator. A strong gap in the error estimator magnitude between the highest value and the mean value over the domain would lead to a localised refinement only in zones with highest intensity of the error estimator, and an insufficient refinement in zones with medium intensity of the estimator. This leads to an early and unwanted stop of the refinement algorithm.

The strategy we propose in this work allows for controlling the number of dofs in the mesh, in order to prescribe the appropriate target error. We assume that the desired number of elements $N^*_{K,i+1}$ in the new mesh can be computed thanks to the prescribed reduction of element sizes [24, 65]:

$$\mathrm{N}^*_{i+1} = \mathrm{NE}^*_{i+1} = \sum_K \left(\frac{h_{K,i}}{h^*_{K,i+1}} \right)^d \tag{47}$$

where d is the number of dimensions of the mesh, and the characteristic sizes of the elements $h_{K,i}$ $h^*_{K,i+1}$ considered here are an average of the lengths associated with the nodes n of the element K.

In the context of *hp*-adaptation the formula above can be rewritten keeping into account a variable polynomial degree for the element K.

$$\mathrm{N}^*_{i+1} = \mathrm{NDOF}^*_{i+1} = \sum_K N_p(p^*_{K,i+1}) \left(\frac{h_{K,i}}{h^*_{K,i+1}} \right)^d \tag{48}$$

where $N_p(p^*_{K,i+1})$ is the number of dofs associated with the desired polynomial degree in the element K.

We introduce a parameter f_r used to define the target error which corresponds to the increase of dofs wanted at each adaptation step.

The target error ϵ^*_{i+1} is then defined based on an iterative procedure, using the bisection method

from the package python SciPy to solve the equation:

$$\mathrm{N}^*_{i+1}(\epsilon^*_{i+1}) - f_r \cdot \mathrm{N}_i = 0 \tag{49}$$

where N_{i+1} is the number of dofs wanted in the following adaptation step.

Starting from the lowest error as a guess of the target error, for h-adaptation, the new lengths are computed from the equation (40), the new number of elements is computed from the equation (47) and the iterative procedure results in the target error ϵ^*_{i+1} needed to obtain $\mathrm{NE}^*_{i+1} = f_r \cdot \mathrm{NE}_i$ elements. For hp-adaptation, the new lengths and the new polynomial degrees are computed respectively with the equations (40) and (43), the new number of dofs is computed with the equation (48) and the iterative procedure results in the target error ϵ^* needed to obtain $\mathrm{NDOF}^*_{i+1} = f_r \cdot \mathrm{NDOF}_i$.

4.6. Particular treatment for hybrid meshes

The definition of prisms or hexahedra in boundary layers is beneficial for computing the gradients with high-accuracy or satisfying precisely near-wall resolution in terms of y^+ for turbulent flows. A common practice involves the definition of a structured (extruded regular quadrilaterals) or pseudo-structured (extruded unstructured quadrilaterals or triangles) zone capturing the boundary layer around the body, which blends with an outer region composed of tetrahedra. [67, 68, 69]. The adaptation methodology proposed in sections 4.3 and 4.4 is straightforwardly extended to hybrid meshes. During the adaptation process, boundary layer structured or pseudo-structured elements are unmodified while the remeshing is applied on tetrahedra only. Regarding the hp-adaptation approach, the p-refinement applies to any type of element, overcoming the constraint that elements with a fixed geometry impose in a pure h-adaptive context. This allows for improving the resolution in boundary layers as well, if the error indicator is active in these regions. No specific treatment is required as regards MMG, which preserves quadrilaterals and prisms by default. The preservation of hexaedra, pyramids and general polygonal elements can be implemented in a straightforward manner.

Algorithm 2: *hp*-adaptation algorithm

set i=0;
create the initial mesh $\mathcal{T}_0$;
obtain from MMG the initial characteristic lengths field h_0;
set an initial polynomial degree map $p_0 = p_{min}$;
while NDOF $<$ NDOF$_{\text{MAX}}$ **do**
  perform the simulation on $\mathcal{T}_{K,i}$;
  compute the *a posteriori* error estimator ϵ_K^i on $\mathcal{T}_{K,i}$;
  compute the smoothness indicator σ_K^i on $\mathcal{T}_{K,i}$;
  choose the target error ϵ_{i+1}^* in order to fulfill the given refinement criterion;
  if $\epsilon_{K,i} > \epsilon_{i+1}^*$ **and** $[(\sigma_{K,i} > 1$ **and** $p_{K,i} + 1 \leq p_{\max})$ **or** $p = 1)]$ **then**
    adapt the current polynomial degree of the elements $K \in \mathcal{T}_{p,i}$ as $p_{K,i+1}^* = p_{K,i} + 1$;
  else
    leave the current polynomial degree of the element $p_{K,i+1}^* = p_{K,i}$ unmodified;
  end
  average the polynomial degree $p_{K,i}$ of elements $K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i}$ to the nodes $p_{n,i}$;
  average the error estimator $\epsilon_{K,i}$ of elements $K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i}$ at the nodes $\epsilon_{n,i}$;
  average the smoothness indicator $\sigma_{K,i}$ of elements $K \in \mathcal{T}_{h,i} \cup \mathcal{T}_{=,i}$ at the nodes $\sigma_{n,i}$;
  if $\epsilon_{n,i} > \epsilon_{i+1}^*$ **then**
    adapt the current size $h_{n,i}$ to the new $h_{n,i+1}^*$ provided by the formulation (40);
  else
    leave the current size $h_{n,i+1}^* = h_{n,i}$ unmodified;
  end
  give the new metric to MMG, which generates the refined mesh $\mathcal{T}_{K,i+1}$;
  project the new polynomial degree map $p_{K,i+1}^*$ from $\mathcal{T}_{K,i}$ to $\mathcal{T}_{K,i+1}$;
  project the solution of $\mathcal{T}_{K,i}$ on $\mathcal{T}_{K,i+1}$;
  i+=1;
end

5. Results on steady laminar configurations

In this section we assess the presented adaptive algorithms on three 2D/3D steady laminar test cases: the 2D flow past a NACA0012 airfoil, the 3D flow past a sphere and the 3D flow past a delta wing.

The parameters of the adaptation process for the laminar test cases are chosen as follows:

- $h_{grad} = 1.5$, which is higher than the default value of MMG $h_{grad} = 1.3$. This value shows a good compromise between excessive refinement induced by a low value of h_{grad}, and the loss of isotropy and mesh quality induced by a high value.

- $hausd = 0.01$ for the sphere, which has constant curvature and diameter equal to 1, and

$hausd = 0.002$ for the NACA0012, which has a chord size equal to 1, but maximum thickness equal to 0.12 and a strong curvature at the leading edge. In general $hauds = 0.01$ is a suitable value for an object of size 1 in each direction [58]. For smaller (respectively larger) objects, the value of the Hausdorff parameter needs to be decreased (respectively increased). This parameter is irrelevant for the delta wing, which does not present curved boundaries.

- h_{min} is set to a low value, which is never reached in the presented adaptation processes.
- h_{max} is set as the size of the domain where far-field boundary conditions are imposed.
- $r_h = 2$ such that between two steps of the adaptation process, the edge of the equilateral triangle can at most halve its size.
- $f_r = 1.5$ is used for all the laminar test cases. A sensitivity study of the adaptation process to this parameter is carried out for the NACA0012 case.
- $p_{\min} = 1$ is the minimum polynomial degree of the spatial discretization allowed for *hp*-adaptation. The initial mesh is provided with a constant polynomial degree map $p = 1$.
- $p_{\max} = 3$ is the maximum polynomial degree that the spatial discretization is allowed to reach for *hp*-adaptation.

Initial meshes are easily created with MMG by providing a uniform coarse metric and forcing the location of the mesh nodes on the wall. By using the mesh smoothing capability of MMG with $hgrad = 1.5$, the resulting mesh is uniform and very coarse far away from the body and presents a smooth refinement towards the solid boundaries. By using this approach, the initial metric at the nodes needed by the adaptation module is provided directly by the remesher.

The error on drag and lift coefficients is computed as the absolute value of the difference between the coefficient of the adaptive simulation at the $i-th$ step, Coef_i, and a reference coefficient Coef_{ref} extracted from a simulation with a finer resolution, normalized by the latter.

$$|\Delta \mathrm{Coef}| = \left| \frac{\mathrm{Coef}_i - \mathrm{Coef}_{\mathrm{ref}}}{\mathrm{Coef}_{\mathrm{ref}}} \right|$$

The error on h- and hp-adapted meshes is compared to uniformly refined meshes, obtained with MMG by prescribing at all nodes a size which is half the size of the previously refined mesh.

5.1. Laminar flow past a NACA0012 airfoil

The adaptive algorithms are assessed in the present section for a flow representative of aerodynamic applications. We consider the steady, subsonic and viscous flow around the NACA0012 airfoil, which is a well-known test case used in CFD for validation purposes (turbulence models, shape optimization, numerical schemes), and among others for steady mesh adaptation [70, 71, 72, 73].

The configuration studied here has a free-stream Mach number of $M_\infty = 0.5$, angle of attack $\alpha = 1^\circ$ and Reynolds number of $Re = 5000$. The main feature of this flow is the thin, laminar boundary layer developing over the airfoil.

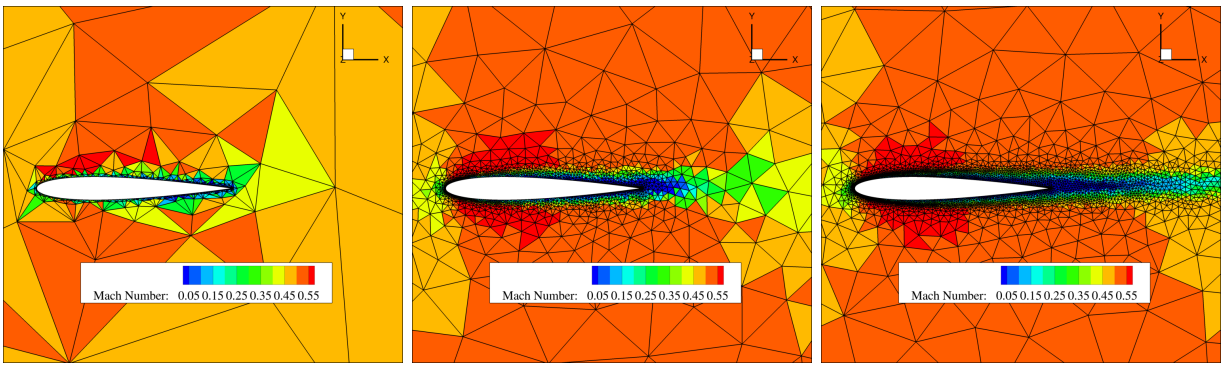
Reference results are obtained on a finer mesh with a $p = 3$ discretization highly refined around the body. Those results compare favorably with other results present in the literature, as seen in Table 1.

	C_D	C_{Dfric}	C_{Dpres}	C_L
Swanson et al. (NASA report) [74]	0.055980	0.0327577	0.023222	0.0184635
Yano and Darmofal [72]	0.0553167	-	-	0.018274
Haga et al. [75]	0.05589	-	-	0.018923
Balan et al. [73]	0.0553168	-	-	-
current	0.0556622	0.0328013	0.0228610	0.0199830

Table 1: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Integral flow quantities found in the literature and for the present reference.

The small discrepancies in the C_L with respect to other references are due to the different analytical definition of the NACA0012 used in this paper. We employed the revised definition [76], while the reference values provided in table 1 have been obtained using the exact formula of the NACA0012 airfoil.

The mesh shown in the left panel in figure 2 contains 789 triangular elements and is used as initial mesh. It is a C-type mesh, centered at the airfoil mid chord with a radius of 80 chords and 100 chords in the rear region. Adiabatic viscous conditions have been imposed on the walls of the airfoil, and free-stream far-field non-reflecting boundary conditions have been imposed on the external boundaries.



(a) (b) (c)

Figure 2: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Zoom on the Mach number iso-contours of the cell-averaged solution (a) on the initial mesh (789 elements/4734 dofs), (b) on the 3^{rd} adapted mesh (3446 elements/20676 dofs) and (c) on the 6^{th} adapted mesh (8974 elements/53844 dofs), for the $h(p = 2)$ simulations.

Seven series of computations are performed: three uniformly refined series with $p = 1, 2, 3$, three pure h-adaptation series with $p = 1, 2, 3$ referred later as $h(p = 1)$, $h(p = 2)$, $h(p = 3)$ and one hp series. Figure 2 presents the meshes generated during the adaptation steps for the $h(p = 2)$ series. We can see that the adaptation yields a concentration of the refinement in the boundary layer region around the airfoil, the stagnation region near the leading edge and the wake in the region downstream of the airfoil.

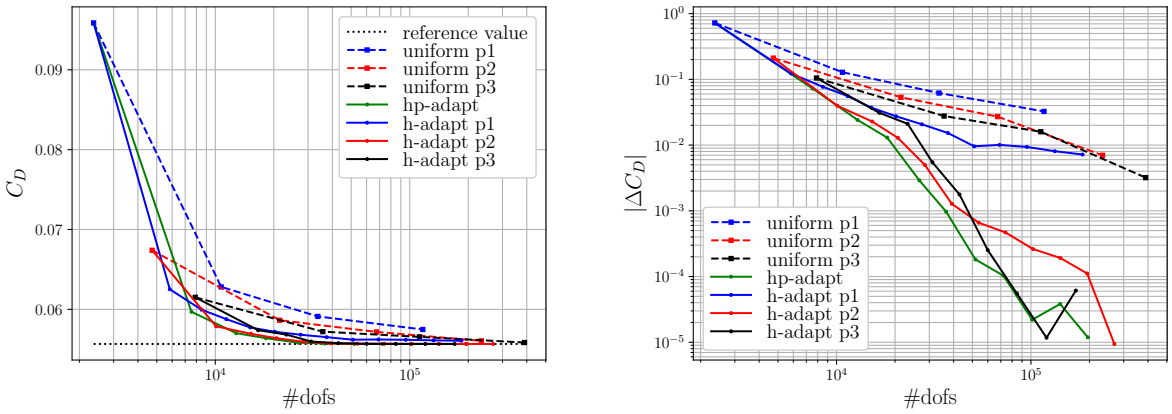


Figure 3: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Comparisons between the convergence history of the drag coefficient for hp-adapted meshes, h-adapted meshes and uniformly refined meshes in $p = 1$, $p = 2$, $p = 3$. C_D vs. number of dofs (left) and $|C_D - C_{D_{ref}}|/C_{D_{ref}}$ vs. number of dofs (right).

The error on the drag coefficient and lift coefficient integrated over the walls of the NACA0012 airfoil, C_D and C_L, is evaluated for the different meshes by computing the difference between the values obtained on the adapted mesh and those from the reference $p = 3$ simulation.

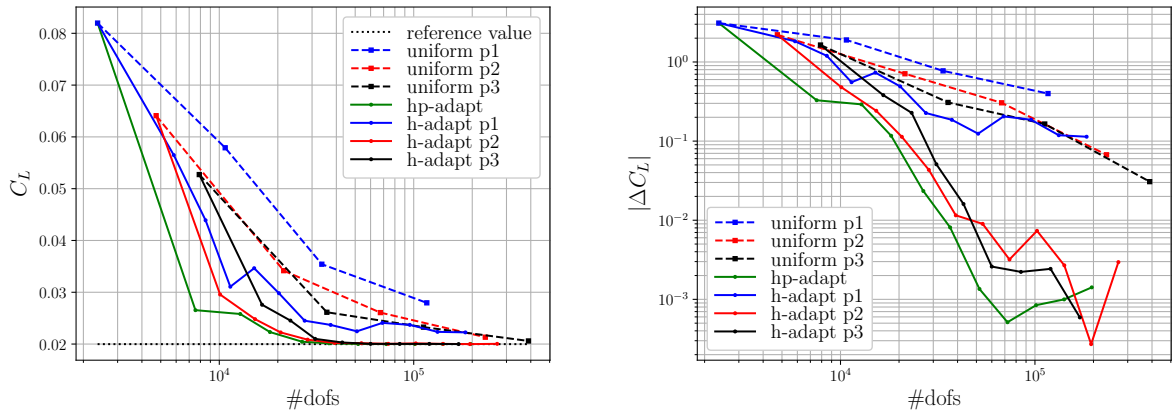
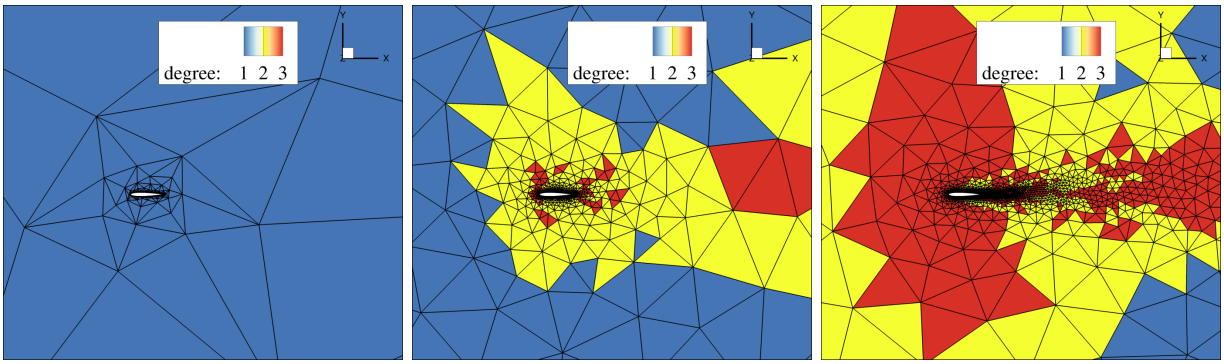


Figure 4: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Comparisons between the convergence history of the lift coefficient for *hp*-adapted meshes, *h*-adapted meshes and uniformly refined meshes in $p = 1$, $p = 2$, $p = 3$. C_L vs. number of dofs (left) and $|C_L - C_{L_{ref}}|/C_{L_{ref}}$ vs. number of dofs (right).

The convergence history of the C_D and C_L (left) and the error (right) versus the number of dofs of the performed simulations are shown respectively in figures 3 and 4. For all pure *h*-adaptive simulations, four adaptation steps are sufficient for $p = 1$, $p = 2$ and $p = 3$ to reach a level of error on the drag coefficient lower than the error obtained on the corresponding finest uniformly refined meshes. The *hp*-adaptive process is also capable in only four adaptations to reach an error level lower than the uniform $p = 3$ most refined mesh, and with less degrees of freedom than $p = 1$, $p = 2$ and $p = 3$ *h*-adaptive simulations. A similar behaviour is observed for the lift coefficient C_L in figure 4.

For the higher order adapted simulations, $h(p = 2)$, $h(p = 3)$ and the *hp*-simulations, we observe that at the end of the adaptive process, the error in the drag and lift coefficients is reduced by around two orders of magnitude with respect to the uniformly refined simulations. Note that the stagnating behavior of the error below a certain threshold is explained by the relative uncertainty about the reference value which is not exact but obtained from a computation on a very fine mesh.

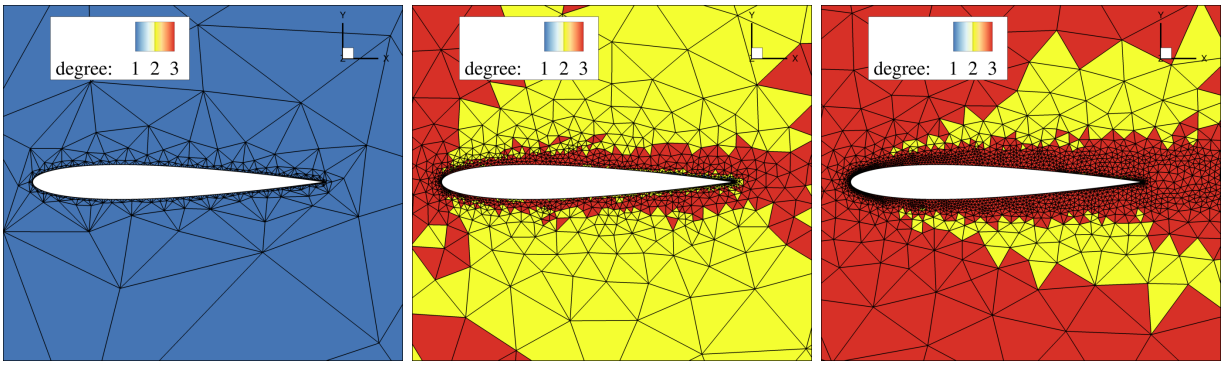
The error on second order $p = 1$ simulations decreases at a lower rate than $h(p = 2)$, $h(p = 3)$ and *hp* simulations, which is due to the combination of several factors. Namely, the initial lower count of dofs, the increased numerical error compared to higher orders and, as discussed in section 4.1, the lower accuracy of the SSED part of the error estimator for low-order discretizations because of the related limited spectral content inside the cells.



(a) (b) (c)

Figure 5: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Contours of the polynomial degree distribution (a) on the initial mesh (4734 dofs), (b) on the 3^{rd} adapted mesh (18219 dofs) and (c) on the 6^{th} adapted mesh (51834 dofs), for the *hp* simulations.

Figure 5 displays the polynomial degree distribution for the first, third and sixth adaptation steps obtained from the *hp*-adaptation process. A part of the wake region is first detected as non-smooth where the elements are still too large to accurately capture the physical features, and present values of the smoothness indicator lower than 1. As the mesh is refined, these regions are detected as smooth and the wake is adapted first by increasing the polynomial degree and then by decreasing the size of the element (see figures 5(b) and 5(c)).



(a) (b) (c)

Figure 6: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Zoom of figure 5.

A close-up view of the p distribution around the airfoil is displayed in figure 6. Since this flow presents globally smooth features, the general trend of the *hp*-adaptation is to first increase the polynomial degree, and then decrease the size when the polynomial degree in the element saturates at the maximum allowed value $p_{\max}$. We can still notice from figure 6(c) that the exterior part of the shear layers of the aifoil remains at $p = 2$ over the course of the adaptation process, while the

element size decrease in that area. This behavior is explained by the gradation of the remesher (see section 3.1) which ensures a smooth increase in the element size between the highly refined near wall area and the outer part. This results in a slight over-refinement in zones where the error estimator is not necessarily high. However, this behavior is compensated by the *hp*-adaptation algorithm which does not increase p in these regions.

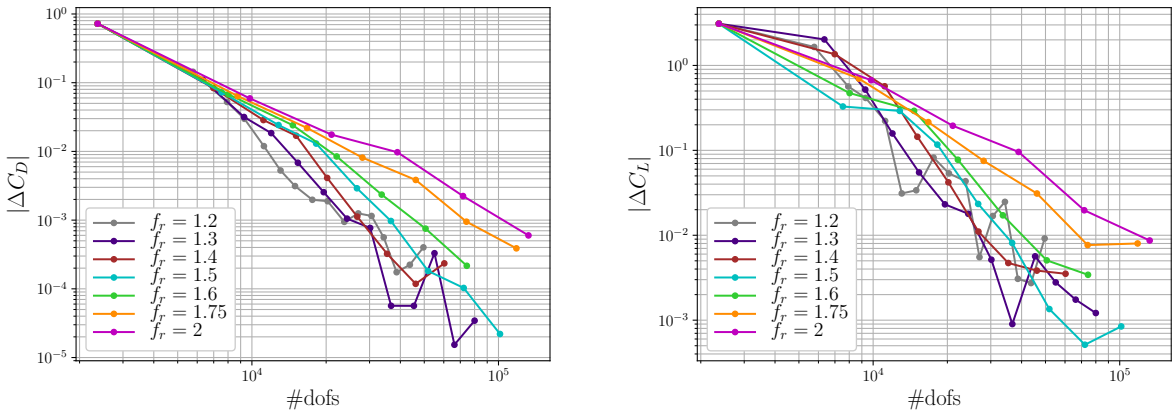


Figure 7: Laminar flow past a NACA0012 airfoil at $Re = 5000$. Study on the independence from the refinement factor on *hp*-adaptive simulations.

The influence of the refinement factor f_r, which controls the increase of number of dofs at each adaptation step has also been assessed for *hp*-adaptive simulations. The corresponding error plots are presented in figure 7. The algorithm is tested by selecting seven different values of f_r from 1.2 to 2, and by analyzing the effect of this parameter on the convergence history of the C_L and C_D values. As we can see from figure 7, all values of f_r drive the integral errors to convergence, showing a low sensitivity to this parameter. This parameter, however, affects the rate of convergence in terms of number of dofs. In particular, by increasing f_r, more elements are selected for refinement, including some elements presenting a low error. On the other hand, a small value of f_r mainly refines the zones presenting a strong error, leading to a more rapid convergence. However, despite the fact that a very small value of f_r makes the algorithm converge to the same value with much less dofs, it needs more adaptation steps to achieve convergence. This process can be time consuming since more remeshing steps and more flow simulations must be realized. In addition, too small refinement factors lead to oscillations in the error because too few elements have been adapted to ensure a smooth convergence behaviour. The choice of the increment in terms of number of dofs

from one step to another can thus be seen as a trade-off between the number of adaptation steps and the computational time. For this reason, in the present work the three laminar configurations are adapted using a value $f_r = 1.5$, which proved to be an efficient choice to ensure a good convergence for a low number of iterations.

5.2. Laminar flow past a sphere at low Reynolds number

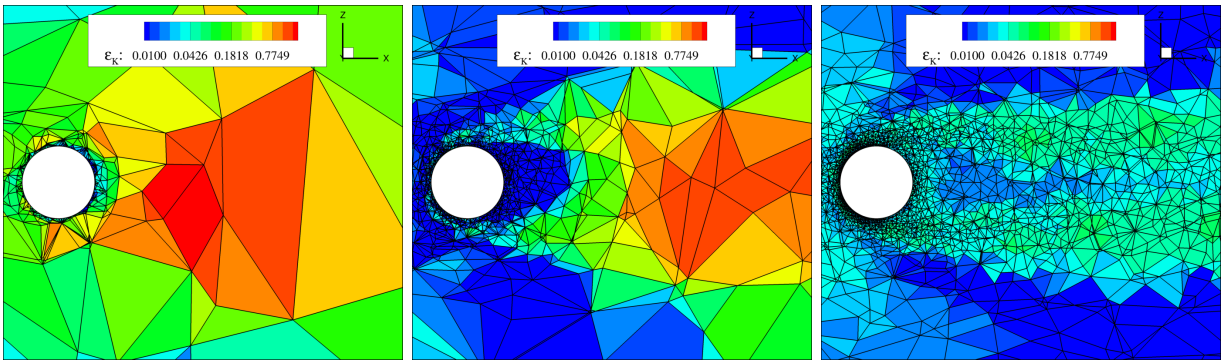
In this section, the *h*- and *hp*-adaptive algorithms are validated in the context of the three-dimensional flow past a sphere at low Reynolds number ($Re = 200$). This particular flow, already studied by Rueda et al. [77] in the context of *p*-adaptative DG methods, is characterized by a separation determined by the viscous processes at the wall, which creates a recirculation region behind the body. This flow regime preserves a steady and axi-symmetrical behaviour, which is lost for higher Reynolds number. Reference results are obtained from a $p = 3$ computation on a highly-refined mesh around the body and compared to results from the literature [77, 78, 79] in table 2.

	C_D
Rueda et al. [77]	0.7771
Fadlun et al. [79]	0.7567
Fornberg [78]	0.7683
current	0.77815623

Table 2: Laminar flow past a sphere at $Re = 200$. Integral flow quantities found in the literature and for the present reference.

The initial mesh counts 6651 tetrahedral elements (plot in figure 8(a)): the density of the mesh on the surface of the body is high enough to obtain an accurate representation of the geometry. This allows the remesher to keep the right curvature of the surface when the mesh is successively refined during the adaptation steps. The mesh is spherical with a radius equal to 100 diameters of the sphere of unit diameter, and is centered at the center of the solid body. The surface of the sphere is treated as a viscous adiabatic wall, and the external boundaries are modeled with free-stream far-field non-reflecting boundary conditions.

Three *h*-adaptive simulations named $h(p = 1)$, $h(p = 2)$, $h(p = 3)$ are performed for 3 different polynomial degrees. An *hp*-adaptive simulation is also performed using $p_{\max} = 3$ and a constant $p = 1$ degree distribution on the initial mesh.



(a) (b) (c)

Figure 8: Laminar flow past a sphere at $Re = 200$. Close-up view of the error estimator iso-contours computed (a) on the initial mesh (6651 elements/26604 dofs), (b) on the 3^{rd} adapted mesh (37920 elements/758400 dofs) and (c) on the 6^{th} adapted mesh (154171 elements/3083420 dofs), for the $p = 3$ simulations. Slices of $y = 0$.

Close-up views of the $h(p = 3)$ meshes are shown in figure 8, namely the initial mesh (figure 8(a)) and the meshes obtained after 3 and 6 adaptation steps (respectively figures 8(b) and 8(c)). The adaptation procedure is able to refine the mesh in flow regions of interest, around the body and in the sphere wake. The iso-values of the error estimator computed at the three steps of the adaptive process are presented, in which we have used the same color-scale for the three plots. The initial mesh presents very high values of the error estimator in the wake, resolved here by a few very coarse elements, and low values on some small surface elements which appear to be already sufficiently refined. We remind here that the values of the error estimator used for the adaptation are normalized by their min/max values in the domain. As the adaptation process progresses, the error estimator varies in a more limited range of values. This is a manifestation that by improving the resolution, the distribution of the error estimator becomes more regular. At the 6th adaptation step, the zone of interest of the flow does not present any element with a significant difference in the error estimator value from the average. The adaptation process can be considered as converged, as from this point on, the h-refinement is going to take place mainly in the far wake, where the error estimator remains relatively high until the wake is fully refined up to the downstream boundary. When this happens, it is not necessary to pursue the adaptation process.

Figure 9 shows the convergence history of the drag coefficient (right) and its error with respect to the reference value (left) for h-/hp-adapted and uniformly refined simulations. We can observe how the adaptive algorithms are capable of reaching the right values of C_D with less dofs than on

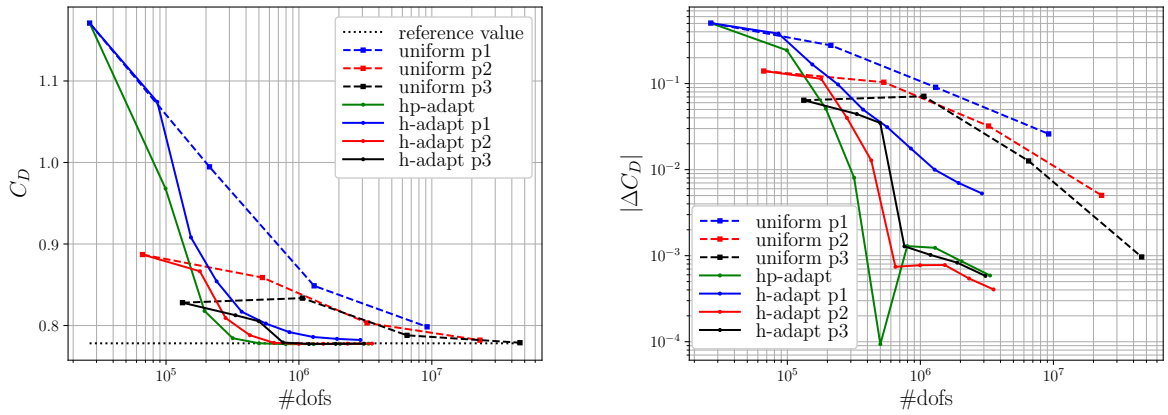


Figure 9: Laminar flow past a sphere at $Re = 200$. Comparisons between the convergence history of the drag coefficient for *hp*-adapted meshes, *h*-adapted meshes and uniformly refined meshes in $p = 1$, $p = 2$, $p = 3$. C_D vs. number of dofs (left) and $|C_D - C_{D_{ref}}|/C_{D_{ref}}$ vs. number of dofs (right).

the uniform meshes. Pure $h(p = 1)$ adaptation converges more slowly than high-order *h*-adaptive simulations, as expected for the considerations already made for the 2D flow past a NACA0012 airfoil. Nonetheless a consistent gain of around 10 times the number of dofs is observed with respect to $p = 1$ uniformly adapted meshes. The *hp*-adaptive procedure converges slightly faster than $h(p = 2)$ and $h(p = 3)$ adaptations in the first steps. Similarly to the NACA0012 case, we observe an oscillating and stagnating behavior of the error below a certain threshold, mostly due to the precision of the reference value which is not exact but obtained from a highly-refined discretization.

When considering stiff problems (e.g. featuring geometrical, physical singularities or turbulence), it can be challenging to obtain a high-order solution on the initial coarse mesh without increasing gradually the polynomial degree first due to robustness issues. This difficulty is alleviated when considering our *hp*-adaptation method as the progressive *p*-refinement based on smoothness detection is integrated in the adaptation process and reinforces the high-order solver robustness. We choose here not to compare the performance including the CPU cost of *hp*-adaptation with respect to pure high-order *h*-adaptation since a fair comparison would require an accurate load balancing strategy [80] and optimized implicit schemes.

5.3. Laminar flow past a delta wing at low Reynolds number and high angle of attack

In this section, the DG h- and hp-adaptation strategies are assessed for a delta wing configuration at laminar conditions with inflow Mach number equal to $M = 0.3$, the angle of attack $\alpha = 12.5°$ and Reynolds number (based on a mean chord length $L = 1$) $Re = 4000$. The delta wing features a sharp leading edge and a blunt trailing edge.

The main feature of this case is the rolling-up of the flow at the leading edge yielding the development of a vortex together with a secondary vortex which persists in the wake.

This test case has been studied in the first three High-Order Workshops [75], as well as by Leicht et al. [70], Ceze et al. [12] and Tsolakis et al. [81] in the framework of h/hp-adaptation. The reference values of drag and lift coefficients C_D and C_L for the current computations have been obtained by extrapolation of the last points of $p = 3$ uniform simulations. The results are shown in table 3 and compared to values used in the literature.

	C_D	C_L
Hartmann et al. [75]	0.1658	0.347
Ceze et al. [75]	0.16578	0.34771
current	0.165617	0.34716

Table 3: Steady laminar flow past a delta wing at $Re = 4000$. Integral flow quantities found in the literature and for the present reference.

The initial mesh of 3775 tetrahedral elements, as for the two previous test cases, is slightly refined near the surface of the wing, with a fast coarsening when moving towards the far-field boundaries. The geometry and the boundaries are built according to the High-Order Workshop meshes: the domain is a square with side $10.62L$, extruded in the streamwise direction of $12.74L$. Half of the model is simulated with symmetric boundary conditions applied on the wall-normal/streamwise plane of symmetry. The wing surface is modeled as a no slip isothermal wall with $T_w = T_\infty$, and the external boundaries as free-stream far-field non-reflecting boundary conditions.

In figure 10 the high-order solution on the initial coarse mesh (left) and on the eighth iteration of the hp-adaptive algorithm, has been reconstructed on a very fine post-processing mesh, and sliced on the $y = 0$ and $x = 1.77L$ planes. The Mach number iso-contour map is shown in the sliced planes, the friction coefficient colors the surface of the wing, and the transparent iso-contour of the Mach number represents $M = 0.2$. Looking at the surface of the wing, we can notice the clustering

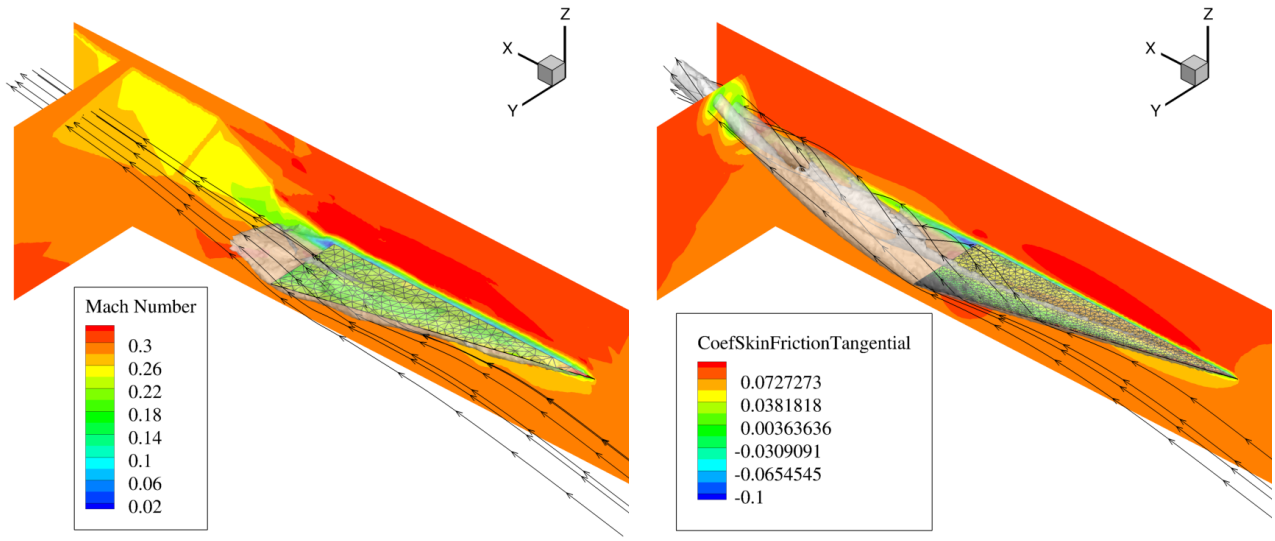


Figure 10: Laminar flow past a delta wing at $Re = 4000$. Slices of $y = 0$ and $x/L = 1.77$ planes of the reconstructed high-order solution on a finer post-processing mesh. Initial mesh (15100 dofs) at left, and 8^{th} hp-adapted mesh (1603248 dofs) at right. Mach contour for volume slices, friction coefficient for surfaces, isosurface $M = 0.2$ transparence.

of the elements along the sharp leading edge, and especially in the rear part, where the geometry changes abruptly into the blunt trailing edge. This refinement allows for an accurate generation of the trailing vortices. A sharp representation of these vortices is also observed further away in the wake, as seen from the x-cut plane and the Mach iso-surfaces enveloping the vortical structures behind the body.

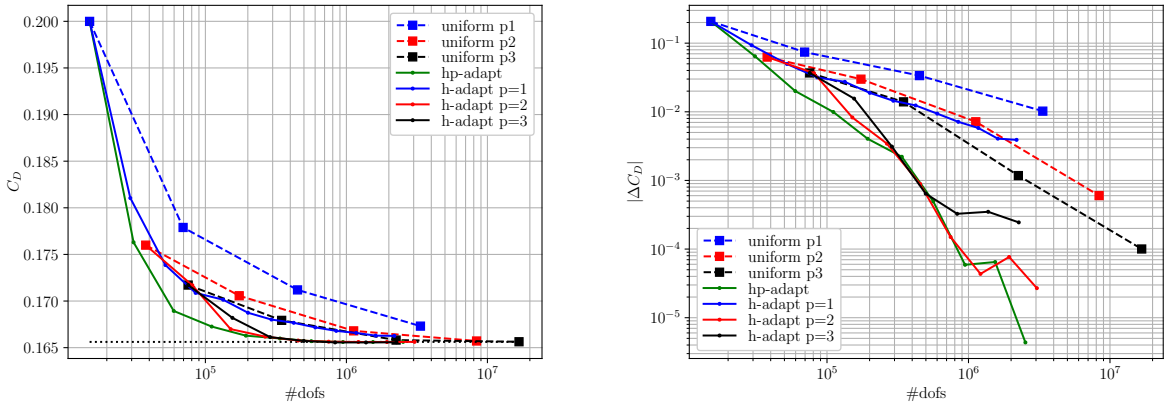


Figure 11: Laminar flow past a delta wing at $Re = 4000$. Comparisons between the convergence history of the drag coefficient for hp-adapted meshes, h-adapted meshes and uniformly refined meshes in $p = 1$, $p = 2$, $p = 3$. C_D vs. number of dofs (left) and $|C_D - C_{D_{ref}}|/C_{D_{ref}}$ vs. number of dofs (right).

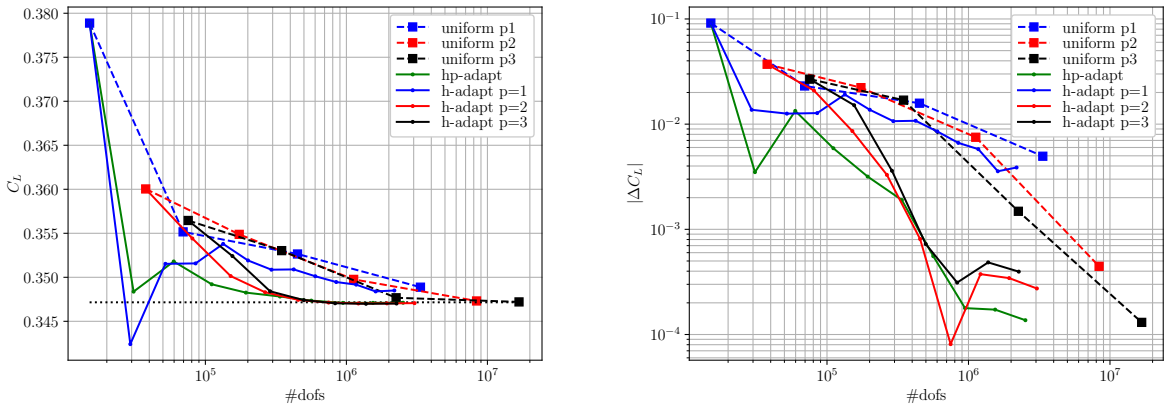
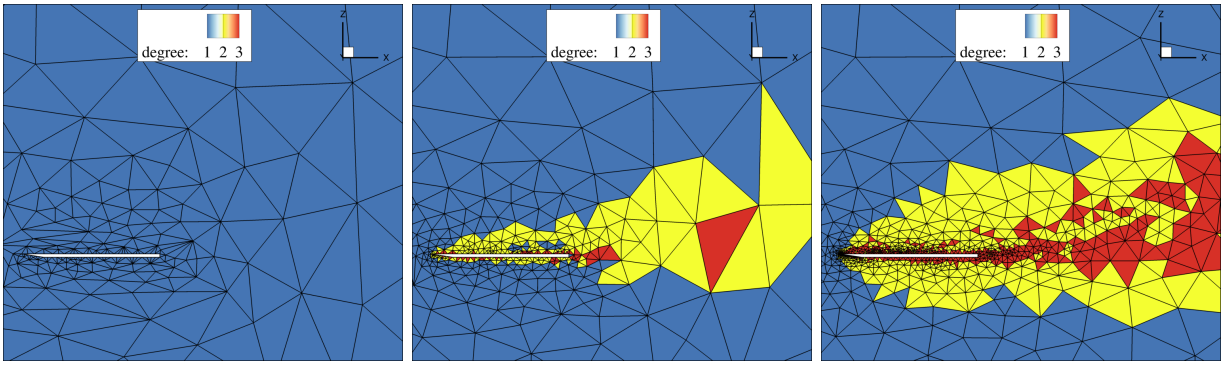


Figure 12: Laminar flow past a delta wing at $Re = 4000$. Comparisons between the convergence history of the lift coefficient for hp-adapted meshes, h-adapted meshes and uniformly refined meshes in $p = 1$, $p = 2$, $p = 3$. C_L vs. number of dofs (left) and $|C_L - C_{L_{ref}}|/C_{L_{ref}}$ vs. number of dofs (right).

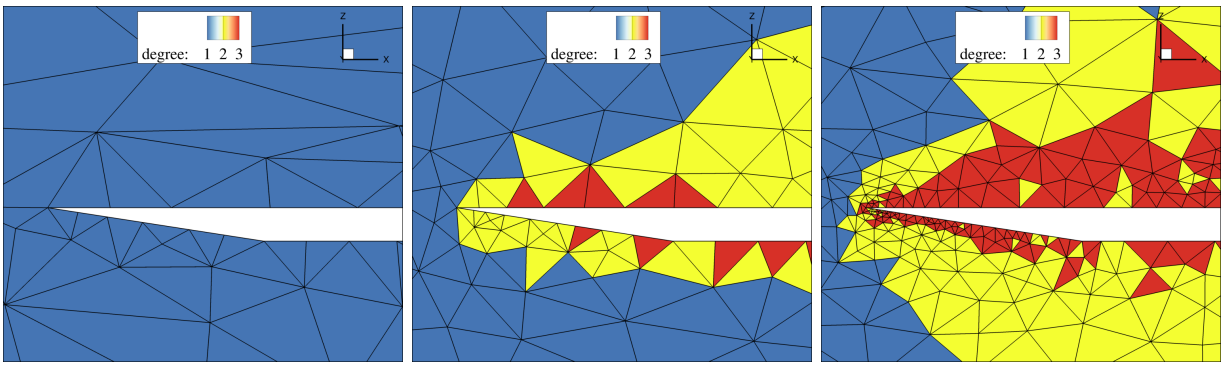
Figures 11 and 12 collect the results obtained on the sequences of uniformly refined meshes with $p = 1, 2, 3$, h-adapted meshes ($h(p = 1)$, $h(p = 2)$, $h(p = 3)$) and $hp-$adapted meshes. C_D and C_L values are shown in right plots, while their errors with respect to the reference solution can be found in left plots. $h(p = 1)$ simulations, despite being clearly the least competitive strategy to reduce the number of dofs, still reaches the same level of accuracy on the errors on target quantities compared to the most uniformly refined $p = 1$ mesh with around respectively 90% (for C_D) and 50% (for C_L) less dofs. As observed for the previous test cases, the use of an hp-adaptive strategy ensures a faster convergence of the error for the first steps of the adaptive process, while the error curve is superposed to that from the $h(p = 2)$ simulations for a higher number of dofs (and lower error). For this test case slightly lower performances of the adaptive process are observed for $h(p = 3)$ simulations, where the dofs in regions far from the body have a strong impact on the dofs count, while not improving the solution in a satisfactory way. However the slower convergence achieved for $h(p = 3)$ simulations with respect to $h(p = 2)$ and hp simulations starts showing for values of the error on the C_D and C_L lower than 10^{-3}, in a region where the extrapolation procedure of the reference value could affect the reliability of the evaluation of the error.

The same remarks made for the previous test cases regarding the peaks of errors in the low error region of the plot are still valid. In this region, oscillations around the reference value can lead to strong low and high peaks lacking of physical meaning in the adaptive procedure.



(a) (b) (c)

Figure 13: Laminar flow past a delta wing at $Re = 4000$. Contours of the polynomial degree distribution (a) on the initial mesh (15100 dofs), (b) on the 3^{rd} adapted mesh (110836 dofs) and (c) on the 6^{th} adapted mesh (563580 dofs), for the *hp* simulations.



(a) (b) (c)

Figure 14: Laminar flow past a delta wing at $Re = 4000$. Close-up view of figure 13

For the *hp*-adaptive simulations, figures 13 and 14 display the polynomial degree distribution on the xz plane at $y = 0$ for the initial mesh, as well as the meshes obtained after 3 and 6 adaptation steps. The geometric singularities are refined first, then the shear layers around the body and the wake are progressively refined over the course of the adaptation process. In particular, the elements around the singularity keep a $p = 2$ polynomial degree for several adaptive steps. This is due to the smoothness indicator marking this zone as non-smooth. As the singular region gets better resolved, those $p = 2$ elements located around the corner progressively switch to $p = 3$.

6. Results on a turbulent configuration: the isothermal subsonic turbulent jet flow issued from the PPRIME nozzle at $Re_D = 10^6$

In this section, the DG-*hp* methodology is assessed for RANS simulations of a turbulent jet configuration described in [82], for which experiments have been performed at the Bruit & Vent jet-noise facility of the Institut PPRIME, Poitiers, France, and reference data are available [82]. Computational references are available as well in the literature. This configuration has been simulated using LES by Brès et al. [82] to analyse features as near-wall adaptive mesh refinement, synthetic turbulence and wall modelling inside the nozzle. Hybrid RANS/LES simulations have been performed by Gand et al. in [83] to assess the generation of a turbulent inflow, and RANS simulations by Neifeld et al. [84] for jet noise prediction purposes with an eddy relaxation source model.

The operating conditions are defined in terms of the total pressure ratio $P_t/P_\infty = 1.7$ and total temperature ratio $T_t/T_\infty = 1.15$. The jet is isothermal ($T_j/T_\infty = 1.0$), the Mach number is $M_j = U_j/c_j = 0.9$, and the Reynolds number $Re_D = \rho_j U_j D/\mu_j \simeq 10^6$, where the subscript j refers to jet properties. U is the mean jet exit longitudinal (x-direction) velocity, c is the speed of sound, $D = 0.05m$ is the exit diameter of the nozzle, ρ is the density and μ is the dynamic viscosity.

The axi-symmetric computational domain extends from approximately -10D (the length of the nozzle in the x-direction) to 50D in the streamwise direction x, and from -30D to 30D in the radial direction, and includes the nozzle geometry with its exit centred at $(x, y, z) = (0, 0, 0)$ (slices shown in figure 15).

A slow coflow at Mach number $M_\infty = 0.01$ is imposed outside the nozzle (the flow is at rest in the experiment). This prevents spurious recirculation and facilitates flow entrainement. All solid surfaces of the nozzle are treated as no-slip adiabatic walls.

The initial mesh of the RANS test case is generated with the pre-processing software ANSA [85]. The geometry of the body and the far field boundaries are created and meshed using triangles, and the surface of the body (the internal and external walls of the nozzle) is projected normal to the wall to generate the prismatic boundary layer, whose first cell measures $0.0003D$ in the wall-normal direction. The rest of the volume is then filled with tetrahedra. The initial mesh created by ANSA is processed by MMG employing the size field output of a first "analysis step". The remesher is

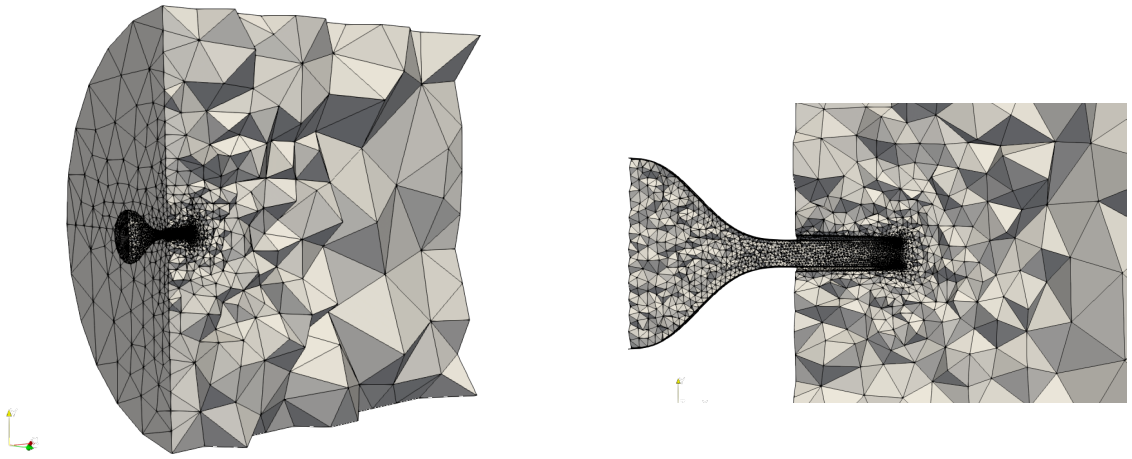


Figure 15: RANS PPRIME nozzle configuration. Slices of the initial mesh, entire domain (left) and zoom on the nozzle (right)

actually able to output an isotropic size field from any given mesh. The parameters considered for the *hp*-adaptive simulation are the same used for the adaptations previously described. The exception is the higher $h_{min} = 8 \cdot 10^{-4} = 0.016D$, which is the length of the smallest edge of the boundary layer prisms which constitute the interface with tetrahedra.

6.1. Modified degree and metric prescription for the PPRIME nozzle

In the field of nozzle/jet configurations an unstructured mesh approach has been extensively used in the literature, both for RANS simulations [86, 87] and resolved LES of jets [88, 89]. Nevertheless, the use of anisotropic elements on boundary layer regions is of crucial importance for RANS equations, where the constraint of $y^+ \simeq 1$ is fundamental to capture the correct boundary layer profile.

During the RANS adaptation process, boundary layer prisms (which include the surface) are kept unchanged, and the remeshing adaptive algorithm acts only for tetrahedra. Due to the twofold nature of the RANS test case chosen here, which couples internal and external aerodynamics, it has been decided not only to preserve the boundary layer, but also all the tetrahedral elements internal to the nozzle. The reason is that a remeshing of the tetrahedra inside the nozzle, which are constrained by the fixed surface sizes can severely affect the quality of the mesh. This step is handled by MMG, which is able to preserve a set of tetrahedra specified by the user.

The *hp*-adaptative algorithm is modified according to the consideration of a frozen region of mesh elements. We can now define three main zones of the mesh $\mathcal{T}_K$: $\mathcal{T}_{BL,i}$, $\mathcal{T}_{TI,i}$, $\mathcal{T}_{TE,i}$, which are respectively the prismatic layer of the near-body pseudo-structured region, the portion of tetrahedral mesh inside the nozzle, and the outer tetrahedral region.

In the regions $\mathcal{T}_{BL,i}$ and $\mathcal{T}_{TI,i}$, which are not subject to *h*-adaptation, the choice between *h*-refinement and *p*-enrichment is relaxed, and an under-resolved element does not require anymore smooth features to be *p*-adapted. A value of the error estimator greater than the target value is sufficient to mark for *p*-refinement. The tetrahedral region external to the nozzle $\mathcal{T}_{TE,i}$ is free to be *h*- and *p*-adapted according to the usual criteria adopted for laminar computations (see section 4.4).

Therefore, to sum up, the only condition that elements with a fixed geometry (prisms and internal tetrahedra, regions $\mathcal{T}_{BL,i}$ and $\mathcal{T}_{TI,i}$,) need to fulfill to be *p*-enriched is to present an error estimator greater than the target error. Elements allowed to be *h*- or *p*- adapted (tetrahedra external to the nozzle $\mathcal{T}_{TE,i}$) are *h*- or *p*-enriched following the procedure described in section 4.4.

6.2. Analysis of results

We present here the results obtained with *hp*-adaptation on the PPRIME configuration. The initial mesh, shown in figure 15, contains around 1.5 millions of dofs, with a constant polynomial degree $p = 1$ over the whole computational domain. In figure 16, a first qualitative comparison can be done on the cell-averaged Mach number solution obtained on the initial mesh and on the *hp*-adapted mesh after 5 adaptation steps, with $f_r = 1.5$, counting 10 million dofs. The initial coarse mesh clearly leads to an early dissipation of the jet and produces strongly asymmetric features. In contrast, the *hp*-adapted mesh in the right panel of figure 16 clearly symmetrizes the flow, and the improved resolution permits a better development of the jet.

The quality of *hp*-adaptation results is quantitatively assessed by extracting the streamwise velocity profiles in the jet at four different positions from the nozzle exit, $x/D = 1, 5, 10, 15$ (see figure 17) and on the jet centerline ($r = 0$) and lipline ($r = D/2$) in figure 18. The curves obtained for different steps of the adaptation process using a refinement factor of $f_r = 1.5$ are compared to the reference experimental and LES results obtained by Brès et al. [82]. Moreover, in order to

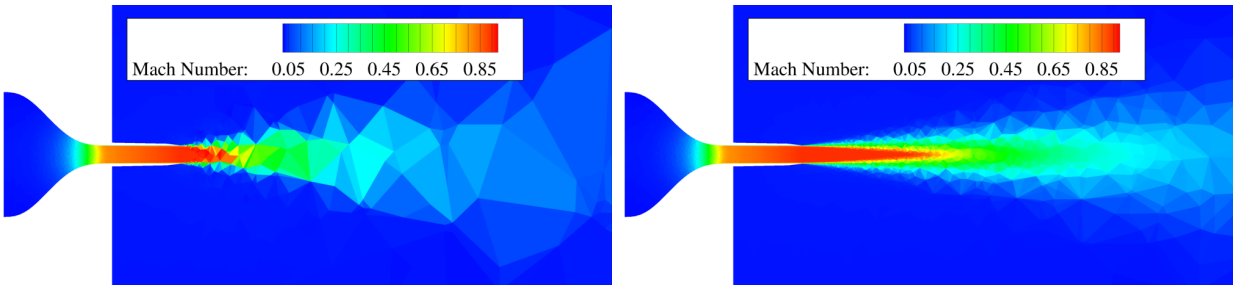


Figure 16: RANS PPRIME nozzle configuration. Contour of the cell-averaged Mach number for the initial mesh (left) and the mesh after 5 adaptation steps with $f_r = 1.5$ (10M dofs).

assess our DG *hp*-adaptive RANS results, we performed a highly resolved second order FV RANS computation on a hexahedral structured mesh counting 48 million elements, and used it as an additional numerical reference.

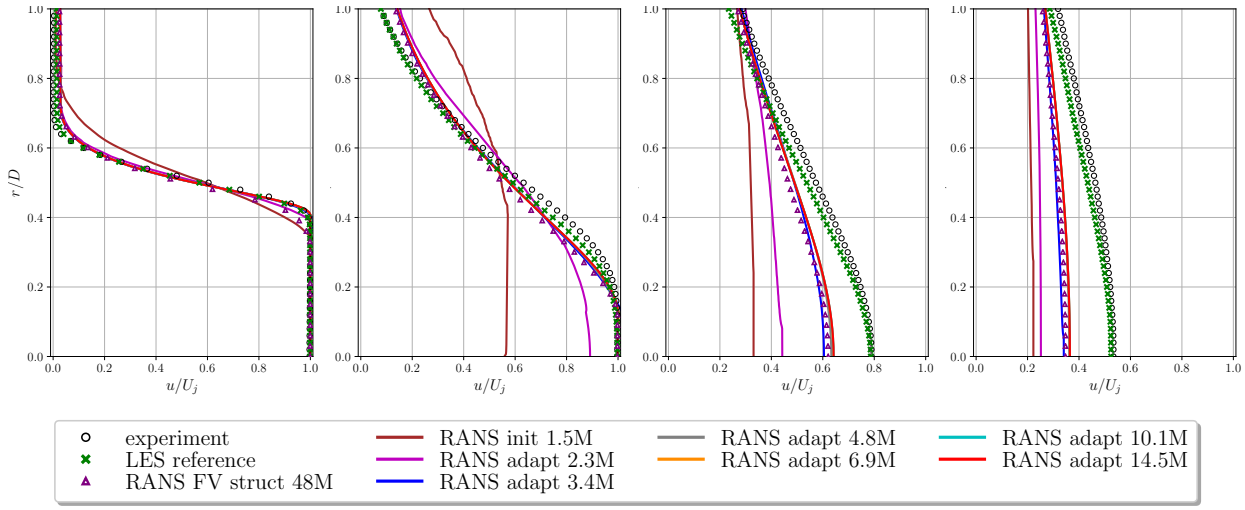


Figure 17: PPRIME nozzle configuration. Radial velocity profiles at $x/D = 1, 5, 10, 15$ (from left to right).

The results yielded by the three last iterations of the *hp*-algorithm with respectively 6.9, 10.1 and 14.4 million dofs practically collapse, showing that mesh convergence has been achieved for the adaptive RANS simulations.

The RANS computations, both DG adaptive and structured reference FV RANS, show a good agreement with the LES results from the literature. This is specially the case in the potential core of the jet and in its vicinity. However, from these plots we can also observe a significant underprediction of the centerline velocity for $x/D \geq 5$. This effect is associated with a lower length of the potential core. This behaviour is a well-known flaw of RANS models, see [87]. Nevertheless,

the adaptive simulation shows overall an improved behaviour with respect to the FV reference computation. This leads to a better agreement with the LES and the experiment, with around 15-20% the number of degrees of freedom of the structured FV simulation, i.e. 7, 10 and 15 versus 48 million dofs). In fact, the use of an automated adaptive process circumvents the difficulties that a classical manual structured meshing process may involve, especially when handling complex geometries. Moreover, high-order DG methods are more accurate than classical FV methods.

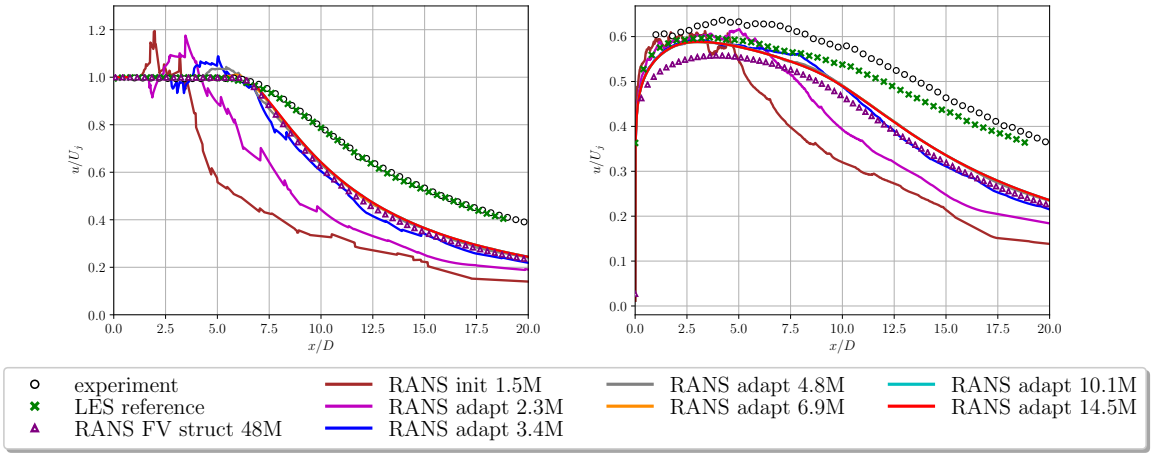


Figure 18: PPRIME nozzle configuration. Velocity profiles on the centerline (left) and on the lipline (right).

At $x/D = 1$ the shape of the top-hat profile is improved at each adaptation step, and the finest adaptive simulations match almost perfectly LES and experiment, presenting a higher accuracy with respect to the fine FV structured simulation. At $x/D = 5$, converged adaptive simulations are still very close to the LES and the experiment, yet showing a slight overprediction of the spreading rate of the jet. At $x/D = 10$ and $x/D = 15$, the radial velocity profiles obtained from RANS simulations show significant discrepancies with respect to the experimental and LES results. However, the *hp*-meshes still perform better than the FV computation. This is due to the weakness of RANS models in predicting some unsteady flow features and the need to perform scale-resolving simulations for this type of flow. Despite this fact, and considering that the *hp*-adaptive simulations performed are based on a RANS model, the results obtained from *hp*-adaptation all converge towards the same solution, in a low number of adaptation steps and in a limited CPU time. As a matter of fact, as already mentioned before, we observe the $hp-$adaptive procedure starting from a low order polynomial distribution gradually improves the robustness of the solution of the implicit system of

equations.

The last aspect we have studied is the influence of the refinement factor f_r on the adaptive process. We show in figure 19 two meshes containing both 10 million dofs, one obtained after five adaptation steps based on a refinement factor $f_r = 1.5$ (left panel in figure 19) and the other after three adaptation steps based on a refinement factor $f_r = 2$ (right panel in figure 19).

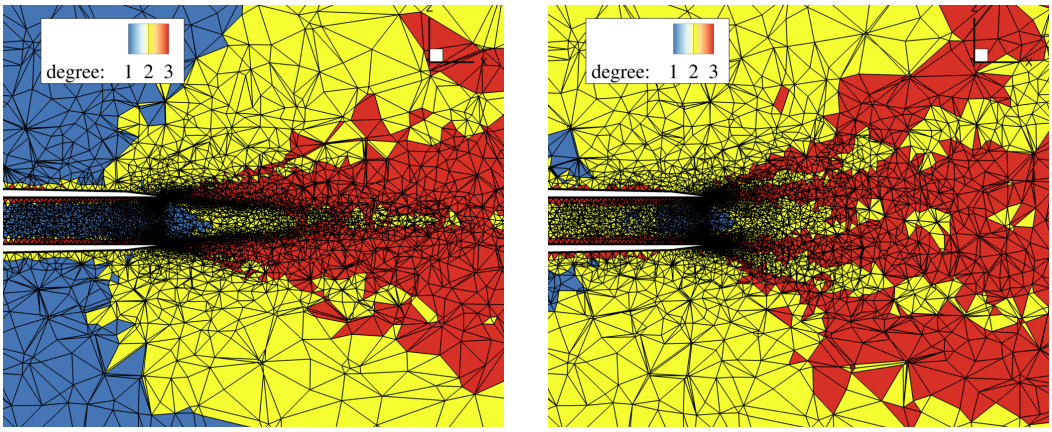


Figure 19: RANS PPRIME nozzle configuration. Zoom on *hp*-adapted meshes with around 10M dofs obtained with $f_r = 1.5$ (left) and $f_r = 2$ (right)

We clearly see how the mesh adaptation algorithm is capable, in both cases, of detecting the zones of interest. Most of the dofs are concentrated around the potential core and in the outmost part of the jet shear layer. In particular, for the mesh refined with a lower refinement factor (and thus more adaptation steps), we can remark a concentration of elements in the zone presenting the highest gradients in the solution. This corresponds to the cone enclosing the potential core. This feature is less marked if a higher refinement factor is used. In the second case (see right panel in figure 19), the adaptation focuses on a wider zone of interest and provides smoother mesh size distribution. This results in a slower convergence towards the optimal mesh, as observed for the NACA0012 test case based on different refinement factors (see figure 7 in section 5.2).

The differences in the results shown in the two meshes in figure 19 can be quantified by computing the L_2-norm of the error on the velocity profiles at $x/D = 1, 5, 10, 15$ along the radial line, and at $r = 0, r = 0.5$ along the respective streamwise line for each adaptation step as in equation (50)

$$error = \frac{||u - u_{ref}||_{L^2}}{||u_{ref}||_{L^2}} = \frac{\int_l (u - u_{ref})^2 dx}{\int_l u_{ref}^2 dx} \tag{50}$$

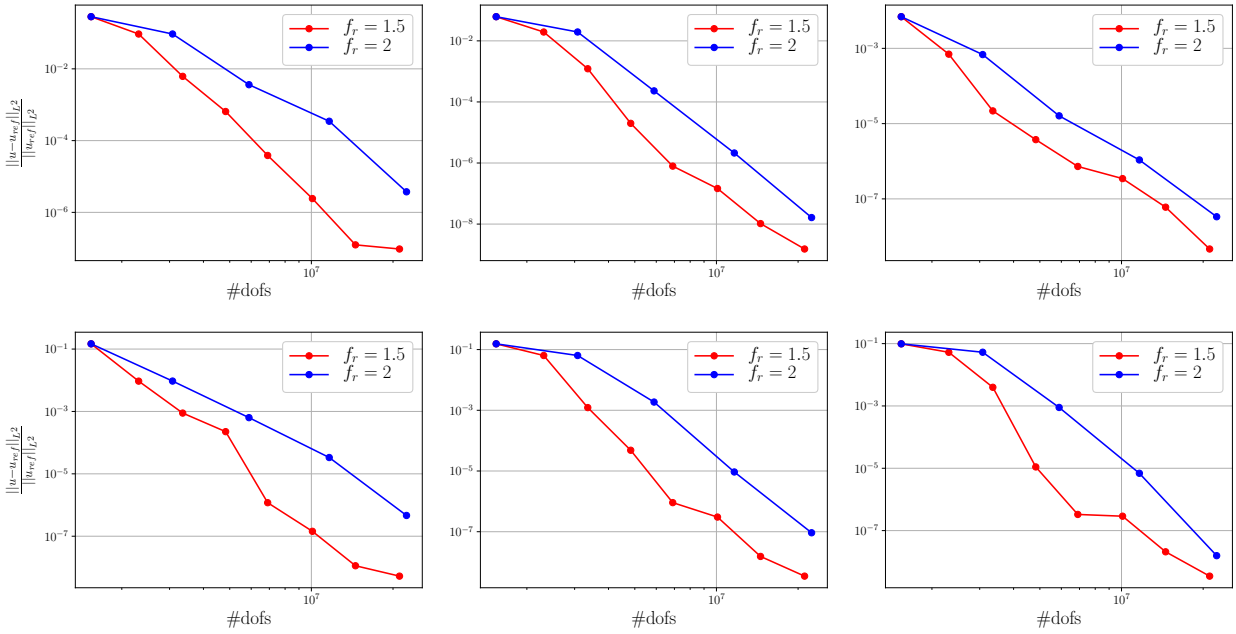


Figure 20: RANS PPRIME nozzle configuration. Comparisons between the convergence history of the norm of the integrated error along the profiles $x/D = 1, 5, 10, 15$ and $r/D = 0, 1/2$ (from top left to bottom right) for two different refinement factors $f_r = 1.5, 2$. *error* vs. number of dofs.

where l is the line on which the profile is extracted, and u_{ref} are the profiles obtained from the fifth adaptation step of the adaptive process based on $f_r = 2$, counting 41 million dofs.

The error plot as a function of number of dofs is shown in figure 20. As expected, the norm of the error on the velocity profiles decreases faster, yet requires more adaptation steps, for a lower value of f_r. In contrast, a higher value value of f_r reduces the convergence rate of the error, though globally needs less adaptation steps. As already pointed out, the optimal value yields a trade-off between maximizing the convergence rate and minimizing the number of adaptations.

This is a delicate topic if we keep in mind the possibility to generate adapted RANS meshes to use as the starting point of a static adaptive procedure for unsteady turbulent flows (currently under study [90]). For scale-resolving simulations, the necessity of reducing the overall computational cost is crucial. In this case, due to the costly transient and the time-averaging of the error estimator needed at each adaptation step, the user might prefer to use a higher refinement factor with the consequent lower number of adaptation steps.

7. Conclusions

In the present paper, h- and hp-adaptive strategies suitable to Discontinuous Galerkin methods have been proposed and assessed, for solving steady fluid flow problems on unstructured meshes. An *a posteriori* error estimator based on both the measure of the energy contained in the highest order polynomial modes and the jumps at the element interfaces has been assessed in the framework of isotropic metric-based h- and hp- adaptation. These methodologies have been combined with a smoothness indicator guiding both the choice between h- and p- adaptation and the reduction of the size of elements marked for h-adatation.

Adaptive DG computations of three configurations of 2D/3D laminar steady flows on triangular/tetrahedral meshes have been performed based on polynomial degrees $p = 1$, $p = 2$ and $p = 3$. The overall adaptation strategies are found to capture accurately the zones in which the solution needs higher refinement and zones already well refined. This yields a significant reduction in terms of number of degrees of freedom to reach a given error level, compared to simulations with uniform mesh refinement. Besides, we focused on the additional accuracy that both h- and hp-adapted meshes using high-polynomial degrees bring with respect to $p = 1$ simulations.

The performance and flexibilty of the proposed hp-algorithm have been also demonstrated in the context of turbulent jet 3D RANS simulations. The consideration of hybrid meshes composed of prisms in boundary layers subject to p-refinement and tetrahedra in the free stream regions subject to both h- and p-refinement allowed for a convergence of the DG-hp numerical solution using only 7 million dofs and improved results compared to a simulation on a structured mesh using classical FV schemes with 48 million dofs. The possibility to exploit RANS adapted meshes as a starting point for the hp-adaptation of hybrid RANS/LES simulations will be the object of future research. Future reasearch will also tackle the problem of higher polynomial degrees and adequate load balancing techniques, as well as the development of anisotropic hp-adaptation strategies to further reduce the number of dofs in this type of simulations.

Acknowledgments

The authors would like to thank Dr F. Naddei from ONERA for his precious advises on the subject of mesh adaptation, as well as Dr F. Gand from ONERA for providing the structured

PPRIME mesh and fruitful discussions on the simulation of jets. Dr A. Froely and Dr L. Cirrottola from INRIA are gratefully acknowledged for their help with MMG features, as well as Dr T. Leicht and Dr J. Löwe from DLR for the implementation of the interpolation tool and the variable-p discretization in CODA. This work has been performed using HPC resources from GENCI-CINES (Grant 2020-A0082A11470) and internal ONERA and Airbus HPC resources.

References

[1] Vít Dolejší. hp-dgfem for nonlinear convection-diffusion problems. *Mathematics and Computers in Simulation*, 87:87–118, 2013.

[2] Tobias Leicht and Ralf Hartmann. Error estimation and hp-adaptive mesh refinement for discontinuous galerkin methods. In *Adaptive high-order methods in computational fluid dynamics*, pages 67–94. World Scientific, 2011.

[3] Li Wang and Dimitri J Mavriplis. Adjoint-based h–p adaptive discontinuous galerkin methods for the 2d compressible euler equations. *Journal of Computational Physics*, 228(20):7643–7661, 2009.

[4] George Karniadakis and Spencer Sherwin. *Spectral/hp element methods for computational fluid dynamics*. Oxford University Press, 2013.

[5] Vít Dolejší and Miloslav Feistauer. Discontinuous galerkin method. *Analysis and Applications to Compressible Flow. Springer Series in Computational Mathematics*, 48, 2015.

[6] Bernardo Cockburn. *Discontinuous Galerkin methods for convection-dominated problems*. Springer, 1999.

[7] Catherine Mavriplis. A posteriori error estimators for adaptive spectral element techniques. In *Proceedings of the Eighth GAMM-Conference on Numerical Methods in Fluid Mechanics*, pages 333–342. Springer, 1990.

[8] Per-Olof Persson and Jaime Peraire. Sub-cell shock capturing for discontinuous galerkin methods. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, page 112, 2006.

[9] G Gassner, C Altmann, F Hindenlang, M Staudenmeier, and CD Munz. Explicit discontinuous galerkin schemes with adaptation in space and time. *36th CFD/ADIGMA course on hp-adaptive and hp-multigrid methods, VKI LS*, 2009.

[10] Jean-François Remacle, Joseph E Flaherty, and Mark S Shephard. An adaptive discontinuous galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM review*, 45(1):53–72, 2003.

[11] Göktürk Kuru, Marta de la Llave Plata, Vincent Couaillier, Rémi Abgrall, and Frédéric Coquel. An adaptive variational multiscale discontinuous galerkin method for large eddy simulation. In *54th AIAA Aerospace Sciences Meeting*, page 0584, 2016.

[12] Marco Ceze and Krzysztof J Fidkowski. Anisotropic hp-adaptation framework for functional prediction. *AIAA journal*, 51(2):492–509, 2013.

[13] Tobias Leicht and Ralf Hartmann. Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *Journal of Computational Physics*, 229(19):7344–7360, 2010.

[14] Jeroen Wackers, Ganbo Deng, Emmanuel Guilmineau, Alban Leroyer, Patrick Queutey, and Michel Visonneau. Combined refinement criteria for anisotropic grid refinement in free-surface flow simulation. *Computers & Fluids*, 92:209–222, 2014.

[15] Frédéric Hecht. Bamg: bidimensional anisotropic mesh generator. *User Guide. INRIA, Rocquencourt*, 17, 1998.

[16] Adrien Loseille and Rainald Lohner. Anisotropic adaptive simulations in aerodynamics. In *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 169, 2010.

[17] Todd Michal and Joshua Krakos. Anisotropic mesh adaptation through edge primitive operations. In *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 159, 2012.

[18] D Ibanez and M Shephard. Mesh adaptation for moving objects on shared memory hardware. *25th International Meshing Roundtable, Sandia National Laboratories*, pages 1–5, 2016.

[19] Gerard J Gorman, Georgios Rokos, James Southern, and Paul HJ Kelly. Thread-parallel anisotropic mesh adaptation. In *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, pages 113–137. Springer, 2015.

[20] Todd Michal, Deric Babcock, Dmitry Kamenetskiy, Joshua Krakos, Mortaza Mani, Ryan Glasby, Taylor Erwin, and Douglas L Stefanski. Comparison of fixed and adaptive unstructured grid results for drag prediction workshop 6. *Journal of Aircraft*, 55(4):1420–1432, 2018.

[21] Frédéric Alauzet and Adrien Loseille. Metrix user guide. error estimates and mesh control for anisotropic mesh adaptation, 01 2009.

[22] Charles Dapogny, Cécile Dobrzynski, and Pascal Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of computational physics*, 262:358–378, 2014.

[23] Guillaume Daviller, Maxence Brebion, Pradip Xavier, Gabriel Staffelbach, Jens-Dominik Müller, and Thierry Poinsot. A mesh adaptation strategy to predict pressure losses in les of swirled flows. *Flow, Turbulence and Combustion*, 99(1):93–118, 2017.

[24] Paul-Emile Bernard, Nicolas Chevaugeon, Vincent Legat, Eric Deleersnijder, and Jean-François Remacle. High-order h-adaptive discontinuous galerkin methods for ocean modelling. *Ocean Dynamics*, 57(2):109–121, 2007.

[25] Michael Andrew Park. Adjoint-based, three-dimensional error prediction and grid adaptation. *AIAA journal*, 42(9):1854–1862, 2004.

[26] Adrien Loseille and Frédéric Alauzet. Continuous mesh framework part i: well-posed continuous interpolation error. *SIAM Journal on Numerical Analysis*, 49(1):38–60, 2011.

[27] Masayuki Yano, James Modisette, and David Darmofal. The importance of mesh adaptation for higher-order discretizations of aerodynamic flows. In *20th AIAA Computational Fluid Dynamics Conference*, page 3852, 2011.

[28] Xiangrong Li, Mark S Shephard, and Mark W Beall. 3d anisotropic mesh adaptation by mesh modification. *Computer methods in applied mechanics and engineering*, 194(48-49):4915–4950, 2005.

[29] Pascal-Jean Frey and Frédéric Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer methods in applied mechanics and engineering*, 194(48-49):5068–5082, 2005.

[30] MJ Castro-Díaz, Frédéric Hecht, Bijan Mohammadi, and O Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, 25(4):475–491, 1997.

[31] Frédéric Alauzet and Adrien Loseille. A decade of progress on anisotropic mesh adaptation for computational fluid dynamics. *Computer-Aided Design*, 72:13–39, 2016.

[32] Jean-François Remacle, Xiangrong Li, Mark S Shephard, and Joseph E Flaherty. Anisotropic adaptive simulation of transient flows using discontinuous galerkin methods. *International Journal for Numerical Methods in Engineering*, 62(7):899–923, 2005.

[33] Frédéric Alauzet, Xiangrong Li, E Seegyoung Seol, and Mark S Shephard. Parallel anisotropic 3d mesh adaptation by mesh modification. *Engineering with Computers*, 21(3):247–258, 2006.

[34] Vít Dolejší. Anisotropic hp-adaptive method based on interpolation error estimates in the lq-norm. *Applied Numerical Mathematics*, 82:80–114, 2014.

[35] Aravind Balan, Michael Woopen, and George May. Adjoint-based hp-adaptivity on anisotropic meshes for high-order compressible flow simulations. *Computers & Fluids*, 139:47–67, 2016.

[36] Ajay Rangarajan, Georg May, and Vit Dolejsi. Adjoint-based anisotropic hp-adaptation for discontinuous galerkin methods using a continuous mesh model. *Journal of Computational Physics*, 409:109321, 2020.

[37] Olivier Coulaud and Adrien Loseille. Very high order anisotropic metric-based mesh adaptation in 3d. *Procedia engineering*, 163:353–365, 2016.

[38] Pierre Benard, Guillaume Balarac, Vincent Moureau, Cécile Dobrzynski, Ghislain Lartigue, and Yves d'Angelo. Mesh adaptation for large-eddy simulations in complex geometries. *International journal for numerical methods in fluids*, 81(12):719–740, 2016.

[39] Wen-zhuang Gui and Ivo Babuška. Theh, p andh-p versions of the finite element method in 1 dimension. *Numerische Mathematik*, 49(6):613–657, 1986.

[40] S Adjerid, M Aiffa, and JE Flaherty. Computational methods for singularly perturbed systems, singular perturbation concepts of differential equations (j. cronin and re o'malley, eds.). In *AMS Proceedings of Symposia in Applied Mathematics, AMS*, 1998.

[41] Paul Houston, Bill Senior, and Endre Süli. Sobolev regularity estimation for hp-adaptive finite element methods. In *Numerical mathematics and advanced applications*, pages 631–656. Springer, 2003.

[42] N Chalmers, G Agbaglah, M Chrust, and C Mavriplis. A parallel hp-adaptive high order discontinuous galerkin method for the incompressible navier-stokes equations. *Journal of Computational Physics: X*, 2:100023, 2019.

[43] Nicholas Burgess and Dimitri Mavriplis. An hp-adaptive discontinuous galerkin solver for aerodynamic flows on mixed-element meshes. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 490, 2011.

[44] Hongyu Wang, William C Tyson, and Christopher J Roy. Discretization error estimation for discontinuous galerkin methods using error transport equations. In *AIAA Scitech 2019 Forum*, page 2173, 2019.

[45] William F Mitchell and Marjorie A McClain. A survey of hp-adaptive strategies for elliptic partial differential equations. In *Recent advances in computational and applied mathematics*, pages 227–258. Springer, 2011.

[46] Vít Dolejší, Georg May, and Ajay Rangarajan. A continuous hp-mesh model for adaptive discontinuous galerkin schemes. *Applied Numerical Mathematics*, 124:1–21, 2018.

[47] Tobias Leicht, Jens Jägersküpper, Daniel Vollmer, Axel Schwöppe, Ralf Hartmann, Jens Fiedler, and Tobias Schlauch. Dlr-project digital-x-next generation cfd solver'flucs'. *CEAS Aeronautical Journal*, 2016.

[48] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.

[49] Steven R Allmaras and Forrester T Johnson. Modifications and clarifications for the implementation of the spalart-allmaras turbulence model. In *Seventh international conference on computational fluid dynamics (ICCFD7)*, pages 1–11, 2012.

[50] Freddie D Witherden and Peter E Vincent. On the identification of symmetric quadrature rules for finite element methods. *Computers & Mathematics with Applications*, 69(10):1232–1241, 2015.

[51] Francesco Bassi, Lorenzo Botti, Alessandro Colombo, Daniele A Di Pietro, and Pietro Tesini. On the flexibility of agglomeration based physical space discontinuous galerkin discretizations. *Journal of Computational Physics*, 231(1):45–65, 2012.

[52] Philip L Roe. Approximate riemann solvers, parameter vectors, and difference schemes. *Journal of computational physics*, 43(2):357–372, 1981.

[53] Ami Harten and James M Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *Journal of computational Physics*, 50(2):235–269, 1983.

[54] Francesco Bassi and Stefano Rebay. A high-order accurate discontinuous finite element method for the numerical solution of the compressible navier–stokes equations. *Journal of computational physics*, 131(2):267–279, 1997.

[55] Cécile Dobrzynski and Pascal Frey. Anisotropic delaunay mesh adaptation for unsteady simulations. In *Proceedings of the 17th international Meshing Roundtable*, pages 177–194. Springer, 2008.

[56] Paul Louis George, Houman Borouchaki, Frédéric Alauzet, Patrick Laug, Adrien Loseille,

David Marcum, and L Maréchal. Mesh generation and mesh adaptivity: theory and techniques. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–51, 2018.

[57] Pascal Jean Frey and Paul-Louis George. *Mesh generation: application to finite elements*. Iste, 2007.

[58] Mmg platform - upgrade your meshes. `https://www.mmgtools.org/`, note = Accessed: 2010-09-30.

[59] S Bourasseau. Contribution to a mesh refinement method based on the adjoint vector for the computation of aerodynamic outputs (theses). *Université Nice Sophia Antipolis*, 2015.

[60] Georges Sadaka, Aina Rakotondrandisa, Pierre-Henri Tournier, Francky Luddens, Corentin Lothodé, and Ionut Danaila. Parallel finite-element codes for the simulation of two-dimensional and three-dimensional solid–liquid phase-change systems with natural convection. *Computer Physics Communications*, 257:107492, 2020.

[61] A Colombo, G Manzinali, A Ghidoni, G Noventa, M Franciolini, A Crivellini, and F Bassi. A p-adaptive implicit discontinuous galerkin method for the under-resolved simulation of compressible turbulent flows. In *7nd European Conference on Computational Fluid Dynamics*, 2018.

[62] F Bassi, L Botti, A Colombo, A Crivellini, M Franciolini, A Ghidoni, and G Noventa. A p-adaptive matrix-free discontinuous galerkin method for the implicit les of incompressible transitional flows. *Flow, Turbulence and Combustion*, 105(2):437–470, 2020.

[63] Fabio Naddei, Marta de la Llave Plata, Vincent Couaillier, and Frédéric Coquel. A comparison of refinement indicators for p-adaptive simulations of steady and unsteady flows using discontinuous galerkin methods. *Journal of Computational Physics*, 376:508–533, 2019.

[64] Francesca Basile, Jean-Baptiste Chapelier, Marta de la Llave Plata, Romain Laraufie, and Pascal Frey. A high-order h-adaptive discontinuous galerkin method for unstructured grids based on a posteriori error estimation. In *AIAA Scitech 2021 Forum*, page 1696, 2021.

[65] Jean-Francois Remacle, Christophe Geuzaine, Gaëtan Compère, and BT Helenbrook. Adaptive mesh generation and visualization. *Encyclopedia of Aerospace Engineering*, 2010.

[66] Jean-François Remacle, Xiangrong Li, Nicolas Chevaugeon, and Mark S Shephard. Transient mesh adaptation using conforming and non conforming mesh modifications. In *IMR*, pages 261–272, 2002.

[67] David L Marcum and Frédéric Alauzet. Unstructured mesh generation using advancing layers and metric-based transition for viscous flowfields. In *21st AIAA Computational Fluid Dynamics Conference*, page 2710, 2013.

[68] Antonis F Antoniadis, Panagiotis Tsoutsanis, and Dimitris Drikakis. Assessment of high-order finite volume methods on unstructured meshes for rans solutions of aeronautical configurations. *Computers & Fluids*, 146:86–104, 2017.

[69] Maximilian Tomac and David Eller. Steps towards automated robust rans meshing. In *4th CEAS Air & Space Conference, 16-19 September 2013, Linköping, Sweden*, pages 114–123. Linköping University Electronic Press, 2013.

[70] Tobias Leicht and Ralf Hartmann. Anisotropic mesh refinement for discontinuous galerkin methods in two-dimensional aerodynamic flow simulations. *Int. J. Numer. Meth. Fluids*, 56:2111–2138, 04 2008.

[71] Doug Pagnutti and Carl Ollivier-Gooch. A generalized framework for high order anisotropic mesh adaptation. *Computers & structures*, 87(11-12):670–679, 2009.

[72] Masayuki Yano and David L Darmofal. Case c1. 3: Flow over the naca 0012 airfoil: Subsonic inviscid, transonic inviscid, and subsonic laminar flows. In *First international workshop on high-order CFD methods*, 2012.

[73] Aravind Balan, Michael Woopen, and Georg May. Adjoint-based hp-adaptation for a class of high-order hybridized finite element schemes for compressible flows. In *21st AIAA Computational Fluid Dynamics Conference*, page 2938, 2013.

[74] R Charles Swanson and S Langer. Comparison of naca 0012 laminar flow solutions: structured and unstructured grid methods, 2016.

[75] International workshop on high-order cfd methods. `https://www1.grc.nasa.gov/research-and-engineering/hiocfd/`, note = Accessed: 2010-09-30.

[76] 2dn00: 2d naca 0012 airfoil validation case. `https://turbmodels.larc.nasa.gov/naca0012_val.html`, note = Accessed: 2021-08-20.

[77] Andrés M Rueda-Ramírez, Juan Manzanero, Esteban Ferrer, Gonzalo Rubio, and Eusebio Valero. A p-multigrid strategy with anisotropic p-adaptation based on truncation errors for high-order discontinuous galerkin methods. *Journal of Computational Physics*, 378:209–233, 2019.

[78] Bengt Fornberg. Steady viscous flow past a sphere at high reynolds numbers. *Journal of Fluid Mechanics*, 190:471–489, 1988.

[79] EA Fadlun, Roberto Verzicco, Paolo Orlandi, and J Mohd-Yusof. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of computational physics*, 161(1):35–60, 2000.

[80] Fabio Naddei. *Adaptive Large Eddy Simulations based on discontinuous Galerkin methods: Simulation adaptative des grandes échelles d'écoulements turbulents fondée sur une méthode Galerkine discontinue.* PhD thesis, Université Paris-Saclay (ComUE), 2019.

[81] Christos Tsolakis and Nikos P Chrisochoides. Anisotropic mesh adaptation pipeline for the 3d laminar flow over a delta wing. In *Modeling, Simulation, and Visualization Student Capstone Conference*, 2019.

[82] Guillaume A Brès, Peter Jordan, Vincent Jaunet, Maxime Le Rallic, André VG Cavalieri, Aaron Towne, Sanjiva K Lele, Tim Colonius, and Oliver T Schmidt. Importance of the nozzle-exit boundary-layer state in subsonic turbulent jets. *Journal of Fluid Mechanics*, 851:83–124, 2018.

[83] Fabien Gand and Maxime Huet. On the generation of turbulent inflow for hybrid rans/les jet flow simulations. *Computers & Fluids*, 216:104816, 2021.

[84] Andrej Neifeld, Dirk Boenke, Juergen Dierke, and Roland Ewert. Jet noise prediction with eddy relaxation source model. In *21st AIAA/CEAS Aeroacoustics Conference*, page 2370, 2015.

[85] Ansa. the advanced cae pre-processing software for complete model build up. `https://www.beta-cae.com/ansa.htm`, note = Accessed: 2010-09-30.

[86] Vance F Dippold. Generating a grid for unstructured rans simulations of jet flows. In *2018 Fluid Dynamics Conference*, page 3223, 2018.

[87] Acoustic reference nozzle with mach 0.97, unheated jet flow. `https://www.grc.nasa.gov/www/wind/valid/arn/index.html`, note = Accessed: 2010-09-30.

[88] M Zhu, C Pérez Arroyo, A Fosso Pouangué, M Sanjosé, and S Moreau. Isothermal and heated subsonic jet noise using large eddy simulations on unstructured grids. *Computers & Fluids*, 171:166–192, 2018.

[89] Mathieu Lorteau, Marta De La Llave Plata, and Vincent Couaillier. Turbulent jet simulation using high-order dg methods for aeroacoustic analysis. *International Journal of Heat and Fluid Flow*, 70:380–390, 2018.

[90] G Balarac, F Basile, P Bénard, F Bordeu, J-B Chapelier, L Cirrottola, G Caumon, C Dapogny, P Frey, A Froehly, et al. Tetrahedral remeshing in the context of large-scale numerical simulation and high performance computing. 2021.