

A p-multigrid strategy with anisotropic p-adaptation based on truncation errors for high-order discontinuous Galerkin methods

Andrés M. Rueda-Ramírez^{a,b,*}, Juan Manzanero^{a,b}, Esteban Ferrer^{a,b}, Gonzalo Rubio^{a,b}, Eusebio Valero^{a,b}

^a ETSIAE-UPM (School of Aeronautics), Universidad Politécnica de Madrid, Plaza de Cardenal Cisneros 3, 28040 Madrid, Spain

^b Center for Computational Simulation, Universidad Politécnica de Madrid, Campus de Montegancedo, Boadilla del Monte, 28660 Madrid, Spain

ARTICLE INFO

Article history:

Received 26 June 2018

Received in revised form 8 October 2018

Accepted 5 November 2018

Available online 12 November 2018

Keywords:

High-order discontinuous Galerkin

Anisotropic p-adaptation

Error estimation

p-Multigrid

Compressible flows

ABSTRACT

High-order discontinuous Galerkin methods have become a popular technique in computational fluid dynamics because their accuracy increases spectrally in smooth solutions with the order of the approximation. However, their main drawback is that increasing the order also increases the computational cost. Several techniques have been introduced in the past to reduce this cost. On the one hand, local mesh adaptation strategies based on error estimation have been proposed to reduce the number of degrees of freedom while keeping a similar accuracy. On the other hand, multigrid solvers may accelerate time marching computations for a fixed number of degrees of freedom.

In this paper, we combine both methods and present a novel anisotropic p-adaptation multigrid algorithm for steady-state problems that uses the multigrid scheme both as a solver and as an anisotropic error estimator. To achieve this, we show that a recently developed anisotropic truncation error estimator [1, A. M. Rueda-Ramírez, G. Rubio, E. Ferrer, E. Valero, Truncation error estimation in the p-anisotropic discontinuous Galerkin spectral element method, J. Sci. Comput.] is perfectly suited to be performed inside the multigrid cycle with negligible extra cost. Furthermore, we introduce a multi-stage p-adaptation procedure which reduces the computational time when very accurate results are required.

The proposed methods are tested for the compressible Navier–Stokes equations, where we investigate two steady-state cases. First, the 2D boundary layer flow on a flat plate is studied to assess accuracy and computational cost of the algorithm, where a speed-up of 816 is achieved compared to the traditional explicit method. Second, the 3D flow around a sphere is simulated and used to test the anisotropic properties of the proposed method, where a speed-up of 152 is achieved compared to the explicit method. The proposed multi-stage procedure achieved a speed-up of 2.6 in comparison to the single-stage method in highly accurate simulations.

© 2018 Elsevier Inc. All rights reserved.

* Corresponding author at: ETSIAE-UPM (School of Aeronautics), Universidad Politécnica de Madrid, Plaza de Cardenal Cisneros 3, 28040 Madrid, Spain.
E-mail address: am.rueda@upm.es (A.M. Rueda-Ramírez).

1. Introduction

Discontinuous Galerkin (DG) methods have gained increasing popularity in the last decades for solving the compressible and incompressible Navier–Stokes equations [2–7]. The lack of a continuity constraint on element interfaces makes DG methods robust for describing advection-dominated problems when an appropriate Riemann solver is selected, and allows them to handle non-conforming meshes with hanging nodes and/or different polynomial orders efficiently [8–10]. This is advantageous for accelerating the computations through adaptation strategies that adjust the element size (h) or the polynomial order (p) locally. The main idea behind these methodologies is to reduce the number of degrees of freedom (DOFs) while maintaining a high accuracy, which translates into shorter computational times and reduced storage requirements. Multi-grid solvers have also been used to accelerate high-order DG time marching computations for a fixed number of degrees of freedom [11–20]. The Discontinuous Galerkin Spectral Element Method (DGSEM) [21,22] is a high-order nodal variant of the DG technique on hexahedral meshes that is especially suited for mesh adaptation strategies because, in addition to the mentioned properties, it handles p -anisotropic representations efficiently [9,23,1].

To fully exploit this feature, we can adapt the mesh locally and anisotropically (both in element size and approximation order), so that the solution captures the flow features of interest at a minimum cost. Local adaptation can be performed by subdividing or merging elements (h -adaptation) or by enriching or reducing the polynomial order in certain elements (p -adaptation). To that end, it is paramount to identify the flow regions that require refinement or coarsening. This has been done historically using three different approaches: the *feature-based adaptation*, the *adjoint-based adaptation*, and the *local error-based adaptation*. A comparison of these three approaches was performed by Frayssé et al. [24] in the context of finite volume approximations and by Kompenhans et al. [25] and Naddei et al. [26] for high-order DG methods. The key ideas behind the adaptation approaches are:

- The *feature-based adaptation* is the classical approach and consists in refining where high velocity, density or pressure gradients are identified. The main disadvantage of these methods is that there is no direct relation between the adaptation criterion and the numerical errors and thus the accuracy is not easily predictable.
- A second and more sophisticated approach is known as *adjoint-based adaptation*. In this approach, a functional target is defined (e.g. drag or lift) and the adjoint problem is solved in order to obtain a spatial distribution of the functional error, which is then used for adapting the mesh. This technique was originally developed for variational formulations [27–29], and it has been also implemented successfully for Finite Volume schemes [30,31]. More recently, Wang and Mavriplis [18] implemented a non-variational formulation for the error estimates and used it to adapt a DG method. The main drawback of this approach is the high computational cost involved in solving the adjoint problem and the storage requirements needed for saving the error estimators.
- A computationally more efficient alternative is the *local error-based adaptation*, which can be based on any local error estimate. Estimations of the local discretization error have been used by Mavriplis [32,33] to develop hp -adaptation techniques for the spectral element method. Later, Casoni et al. [34] extended that approach to adapt the artificial viscosity in shock capturing discontinuous Galerkin discretizations. In the DG community, a simple and computationally inexpensive error indicator, that has already been used by Krivotonova et al. [35,36] for adaptation purposes, is the so-called non-conformity error, which quantifies the jump discontinuities. The residual-based p -adaptation is also a *local error-based adaptation* method, which uses the residual to measure how accurate is the local approximation. This method was originally developed for Finite Elements (FE) and has been successfully used in DG methods [26,28]. In the case of modal (hierarchical) DG methods, a possibility is to employ low cost error estimates that take advantage of the modal representation to drive p -adaptation procedures, such as the Variational Multiscale (VMS) indicator by Kuru and De la Llave Plata [37], or the spectral decay indicator by Persson and Peraire [38]. These have in common that they measure the relative contribution of the different modes to a quantity (kinetic energy, enthalpy, entropy, etc.). If the quantity is concentrated in the highest-order modes, refinement is performed.

On the other hand, the τ -estimation method proposed by Brandt [39], which estimates the local truncation error by injecting a fine grid solution into coarser meshes, has been used for adaptation purposes in low-order schemes [40, 24,41–44]. Rubio et al. [45] extended the τ -estimation approach to high-order methods using a continuous Chebyshev collocation method. Later, Rubio et al. applied it to DGSEM discretizations [46], and studied the quasi-*a priori* truncation error estimation, which allows estimating the truncation error without having fully converged fine solutions. Kompenhans et al. [23] applied the τ -estimation approach to perform p -adaptation using the Euler and Navier–Stokes equations, and showed that a reduction of the truncation error increases the numerical accuracy of all functionals. Furthermore, Kompenhans et al. [25] also showed that τ -based adaptation exhibits better performance than feature-based adaptation, and that anisotropic p -adaptation is more efficient than its isotropic counterpart since similar levels of accuracy can be achieved with the former using fewer degrees of freedom. The adaptation strategy consisted in converging a high order representation (reference mesh) to a specified global residual and then performing a single error estimation followed by a corresponding p -adaptation process. More recently, Rueda-Ramírez et al. [1] developed a new method for estimating the truncation error that is cheaper to evaluate than previous implementations, and showed that it produces very accurate extrapolations of the truncation error, which enables using coarser reference meshes.

The truncation error estimator bears a strong resemblance to residual-based indicators since its formulation departs from the residual equation (as shown in section 3.2.1). Moreover, it is also similar to error estimates based on hierarchi-

cal bases because the τ -estimation measures the error by comparing approximations of different polynomial orders. As the Fourier coefficients of a representation, our measured quantity, i.e. the truncation error, is known to decay spectrally in smooth solutions. Therefore, if the estimation is good, it is possible to extrapolate the behavior and assess the exact polynomial order for a desired error threshold [1]. The authors remark that, although the present work is for a nodal (collocation) DG scheme, the proposed adaptation strategy is also suited for modal (hierarchical bases) DG approaches.

The second methodology used in this work are multigrid algorithms. Multigrid methods were first proposed by Brandt [47], who discovered that the classic iterative methods (also referred to as *smoothers*) eliminate the high-frequency components of the error quickly, but fail to eliminate the low-frequency components efficiently. Therefore, he proposed to use coarser meshes to eliminate low-frequency modes. His approach is known as h-multigrid and has been used extensively in low order methods such as traditional Finite Difference and Finite Volume schemes [48–50]. Craig and Zienkiewicz [51], and Rønquist and Patera [52] were the first authors working on high-order methods that proposed the use of the polynomial order, p , to define the levels of a multigrid scheme, the former for p-finite elements and the latter for nodal spectral elements. After these initial works, the use of multilevel methods spread in the high-order community; initially as p-multigrid methods [11–15] and more recently as hp-multigrid methods [16–20]. Most of these implementations use modal hierarchical shape functions [16,14,11,17], and only a small number of publications focus on nodal-based shape functions [15,11].

Two types of multilevel methods can be found in the literature: linear and nonlinear multigrid methods. The former is *de facto* a linear solver and is usually employed to solve the system of equations obtained from an implicit time integration scheme after linearizing with a Newton or Picard iteration. In this case, the smoother is an iterative method for sparse linear systems [53]. The latter, also known as Full Approximation Scheme (FAS), consists in applying the multigrid directly to the set of nonlinear equations. In such a case, the smoother can be either a time-marching scheme (implicit or explicit), or an iterative method applied to the linearized problem. A comparison of linear and nonlinear multigrid methods for DG discretizations can be found in [16]. In our work, we make use of the nonlinear multigrid scheme since, as will be shown, it enables the estimation of the truncation error of coarse representations.

In this paper, we build on the work on p-adaptation by Kompenhans et al. [23] and on τ -estimators by Rueda-Ramírez et al. [1], and combine them with multigrid solution techniques in order to accelerate the convergence of steady-state solutions of the Navier–Stokes equations using the DGSEM. We use the multigrid scheme both as a solver and as an estimator of the truncation error of anisotropic polynomial representations. To do so, we show that the recently developed truncation error estimator by Rueda-Ramírez et al. [1] is well suited to be evaluated inside an anisotropic p-multigrid cycle with a negligible extra cost. The proposed method results in measured speed-ups of up to 816 in a proposed 2D boundary layer case and of 151 in a 3D study of the flow around a sphere, as compared to a traditional explicit solution method. The coupling of multigrid and p-adaptation also enables to propose a multi-stage adaptation process with increasing order representations which reduces the computational cost when very accurate results are required, resulting in speed-ups of 2.6 with respect to the single-stage adaptation process. To the best of our knowledge, this is the first work on DG that couples an anisotropic p-adaptation technique with multigrid.

The rest of the paper is organized as follows: in section 2, the discontinuous Galerkin spectral element method is briefly explained. Section 3 details the acceleration methods the current work builds on; namely, multigrid, p-adaptation based on τ -estimations, and their coupling. We finish section 3 describing how the coupling of multigrid and p-adaptation enables to introduce new features that speed up the solution procedure. In section 4, we study the performance of the proposed p-adaptation algorithm by means of solving 2D and 3D boundary layer test cases. Finally, the main conclusions are gathered in section 5.

2. The discontinuous Galerkin spectral element method

We consider the approximation of systems of conservation laws,

$$\mathbf{q}_t + \nabla \cdot \mathcal{F} = \mathbf{s}, \quad (1)$$

where \mathbf{q} is the vector of conserved variables, \mathcal{F} is the flux dyadic tensor which depends on \mathbf{q} and $\nabla \mathbf{q}$, and \mathbf{s} is a source term. As detailed in Appendix A, the compressible Navier–Stokes equations can be represented using equation (1). Multiplying equation (1) by a test function \mathbf{v} and integrating by parts over the domain Ω yields the weak formulation:

$$\int_{\Omega} \mathbf{q}_t \mathbf{v} d\Omega - \int_{\Omega} \mathcal{F} \cdot \nabla \mathbf{v} d\Omega + \int_{\partial\Omega} \mathcal{F} \cdot \mathbf{n} \mathbf{v} d\sigma = \int_{\Omega} \mathbf{s} \mathbf{v} d\Omega, \quad (2)$$

where \mathbf{n} is the normal unit vector on the boundary $\partial\Omega$. Let the domain Ω be approximated by a tessellation $\mathcal{T} = \{e\}$, a combination of K finite elements e of domain Ω^e and boundary $\partial\Omega^e$. Moreover, let \mathbf{q} , \mathbf{s} , \mathcal{F} and \mathbf{v} be approximated by piece-wise polynomial functions (that are continuous in each element) defined in the space of L^2 functions:

$$\mathcal{V}^N = \{\mathbf{v}^N \in L^2(\Omega) : \mathbf{v}^N|_{\Omega^e} \in \mathcal{P}^N(\Omega^e) \ \forall \ \Omega^e \in \mathcal{T}\}, \quad (3)$$

where $\mathcal{P}^N(\Omega^e)$ is the space of polynomials of degree at most N defined in the domain of the element e . We remark that the functions in \mathcal{V}^N may be discontinuous at element interfaces and that the polynomial order N may be different in each element and local coordinate direction. Equation (2) can then be rewritten for each element as:

$$\int_{\Omega^e} \mathbf{q}_f^{eN} \mathbf{v}^{eN} d\Omega^e - \int_{\Omega^e} \mathcal{F}^{eN} \cdot \nabla \mathbf{v}^{eN} d\Omega^e + \int_{\partial\Omega^e} \mathcal{F}^* (\mathbf{q}^{eN}, \mathbf{q}^{-N}, \mathbf{n}) \mathbf{v}^{eN} d\sigma^e = \int_{\Omega^e} \mathbf{s}^{eN} \mathbf{v}^{eN} d\Omega^e, \quad (4)$$

where the superindex “ e ” refers to the functions as evaluated inside the element e , i.e. $\mathbf{q}^{eN} = \mathbf{q}^N|_{\Omega^e}$; whereas the superindex “ $-$ ” refers to the value of the functions on the external side of the interface $\partial\Omega^e$. The numerical flux function, \mathcal{F}^* , allows to uniquely define the flux at the element interfaces and to weakly prescribe the boundary data as a function of the conserved variable on both sides of the boundary/interface (\mathbf{q}^{eN} and \mathbf{q}^{-N}) and the normal vector (\mathbf{n}). Multiple choices for the numerical flux functions can be found in the literature [54,55]. In the present work, we use Roe [56] as the advective Riemann Solver and Bassi–Rebay 1 [57] as the diffusive Riemann solver. We remark that the numerical flux must be computed in a specific manner when the representation is non-conforming [9].

Since \mathbf{q}^N , \mathbf{s}^N , \mathbf{v}^N and \mathcal{F}^N belong to the polynomial space \mathcal{V}^N , it is possible to express them inside every element as a linear combination of basis functions $\phi_n \in \mathcal{P}^N(\Omega^e)$:

$$\begin{aligned} \mathbf{q}|_{\Omega^e} &\approx \mathbf{q}^{eN} = \sum_n \mathbf{Q}_n^e \phi_n^e(\mathbf{x}), \quad \mathbf{s}|_{\Omega^e} \approx \mathbf{s}^{eN} = \sum_n \mathbf{S}_n^e \phi_n^e(\mathbf{x}), \\ \mathbf{v}|_{\Omega^e} &\approx \mathbf{v}^{eN} = \sum_n \mathbf{V}_n^e \phi_n^e(\mathbf{x}), \quad \mathcal{F}|_{\Omega^e} \approx \mathcal{F}^{eN} = \sum_n \mathcal{F}_n^e \phi_n^e(\mathbf{x}). \end{aligned} \quad (5)$$

Therefore, equation (4) can be expressed in a discrete form as

$$[\mathbf{M}]^e \frac{\partial \mathbf{Q}^e}{\partial t} + \mathbf{F}^e(\mathbf{Q}) = [\mathbf{M}]^e \mathbf{S}^e, \quad (6)$$

where $\mathbf{Q}^e = (\mathbf{Q}_1^e, \mathbf{Q}_2^e, \dots, \mathbf{Q}_n^e, \dots)^T$ is the local solution that contains the coefficients of the linear combination for the element e ; $\mathbf{Q} = (\mathbf{Q}^1, \mathbf{Q}^2, \dots, \mathbf{Q}^K)^T$ is the global solution that contains the information of all elements; $[\mathbf{M}]^e$ is known as the elemental mass matrix, and $\mathbf{F}^e(\cdot)$ is a nonlinear spatial discrete operator on the element level:

$$[\mathbf{M}]_{i,j}^e = \int_{\Omega^e} \phi_i^e \phi_j^e d\Omega^e, \quad (7)$$

$$\mathbf{F}^e(\mathbf{Q})_j = \sum_i \left[- \int_{\Omega^e} \mathcal{F}_i^e \cdot \phi_i^e \nabla \phi_j^e d\Omega^e \right] + \int_{\partial\Omega^e} \mathcal{F}^{*N}(\mathbf{Q}^e, \mathbf{Q}^-, \mathbf{n}) \phi_j^e d\sigma^e. \quad (8)$$

Note that the operator \mathbf{F}^e is applied on the global solution, since it is the responsible for connecting the elements of the mesh (weakly). Assembling the contributions of all elements into the global system we obtain

$$[\mathbf{M}] \frac{\partial \mathbf{Q}}{\partial t} + \mathbf{F}(\mathbf{Q}) = [\mathbf{M}] \mathbf{S}. \quad (9)$$

In the DGSEM [21], the tessellation is performed with non-overlapping hexahedral elements of order $N = (N_1, N_2, N_3)$ (independent in every local coordinate direction) and the integrals are evaluated numerically by means of a Gaussian quadrature that is also of order $N = (N_1, N_2, N_3)$. For complex geometries, it is most convenient to perform the numerical integration in a reference element and transform the results to the physical space by means of a high-order mapping of order $M = (M_1, M_2, M_3)$:

$$\mathbf{x}^e = \mathbf{x}^e(\boldsymbol{\xi}) \in \mathcal{P}^M, \quad \boldsymbol{\xi} = (\xi, \eta, \zeta) \in [-1, 1]^3. \quad (10)$$

The differential operators can be expressed in the reference element in terms of the covariant (\mathbf{a}_i) and contravariant (\mathbf{a}^i) metric tensors [21]:

$$\mathbf{a}_i = \frac{\partial \mathbf{x}^e}{\partial \xi_i}, \quad \mathbf{a}^i = \nabla \xi_i, \quad i = 1, 2, 3. \quad (11)$$

Using these mappings, the gradient and divergence operators become

$$\nabla q = \frac{1}{J} \sum_{i=1}^d \frac{\partial}{\partial \xi_i} (J \mathbf{a}^i q), \quad \nabla \cdot \mathbf{f} = \frac{1}{J} \sum_{i=1}^d \frac{\partial}{\partial \xi_i} (J \mathbf{a}^i \cdot \mathbf{f}), \quad (12)$$

where the Jacobian of the transformation can be expressed in terms of the covariant metric tensor:

$$J = \mathbf{a}_i \cdot (\mathbf{a}_j \times \mathbf{a}_k), \quad (i, j, k) \text{ cyclic.} \quad (13)$$

The covariant vectors can be readily obtained from the mapping (equation (10)). For 2D problems, the contravariant vectors can be obtained with the well-known “cross product form” [58]. However, for fully 3D problems, the contravariant vectors must be obtained using either the “conservative curl form” or the “invariant curl form” [58]. Since in this work we deal with 3D curved meshes, the “invariant curl form” is selected:

$$Ja_n^i = -\frac{1}{2} \hat{x}_i \cdot \nabla_\xi \times [\mathbf{I}^N (X_l \nabla_\xi X_m - X_m \nabla_\xi X_l)] \quad i = 1, 2, 3, \quad n = 1, 2, 3, \quad (n, m, l) \text{ cyclic,} \quad (14)$$

where \mathbf{I}^N is an interpolating operator that converts an arbitrary continuous function into a polynomial expansion (as in equation (5)).

Similarly, in the DGSEM the order of the mapping (M in equation (10)) must be $M_i \leq N_i$ for 2D, 2D-extruded and 3D p-conforming representations (subparametric or at most isoparametric mapping) to retain free-stream-preservation [58], whereas it is limited to $M_i \leq N_i/2$ for general 3D nonconforming representations [59].

Furthermore, in the DGSEM the polynomial basis functions (ϕ_n in equation (5)) are tensor product reconstructions of Lagrange interpolating polynomials on quadrature points in each of the Cartesian coordinate directions of the reference element:

$$\mathbf{q}^N = \sum_n \mathbf{Q}_n \phi_n(\mathbf{x}) = \sum_{i=0}^{N_1} \sum_{j=0}^{N_2} \sum_{k=0}^{N_3} \mathbf{Q}_{i,j,k} l_i(\xi) l_j(\eta) l_k(\zeta). \quad (15)$$

Therefore, $\mathbf{Q}_n = \mathbf{Q}_{i,j,k}$ are simply the nodal values of the solution, and $[\mathbf{M}]$ is a diagonal matrix containing the quadrature weights and the mapping terms. In the present work, we make use of the Legendre–Gauss quadrature points [21].

A final remark should be made regarding how the time step is chosen. Since in this paper we make use of explicit time integration schemes, the Courant–Friedrich–Levy (CFL) condition dictates a time step limit [60,61]:

$$\Delta t = \min(\Delta t^a, \Delta t^v), \quad (16)$$

where the advective time-step restriction is

$$\Delta t^a \leq C^a \left(\|\mathcal{S}\| \frac{N^2}{h} \right)^{-1}, \quad (17)$$

and the diffusive time-step restriction is

$$\Delta t^v \leq C^v \left(\mu \frac{N^4}{h^2} \right)^{-1}, \quad (18)$$

where C^a and C^v are constants that depend on the time integration method, $\mathcal{S} = \mathbf{v} + c$ is the characteristic velocity (with \mathbf{v} the flow velocity and c the speed of sound), μ is the fluid viscosity, and h is the local mesh size. This quantity is evaluated in every time step on the Gauss points of the domain taking into account the possibility of having anisotropic polynomial orders. The most restrictive Δt is always chosen.

3. Acceleration techniques to converge to steady-state

In this section, we describe the two methods that will be used in the present work to obtain steady-state solutions of the Navier–Stokes equations; namely, nonlinear multigrid schemes and p-adaptation methods based on truncation error estimators.

A common way of obtaining a steady-state solution for an unsteady PDE is to start from an arbitrary initial condition and integrate in time until the system converges to a steady solution. The time-stepping scheme can be either explicit or implicit. Explicit high-order Runge–Kutta methods have been traditionally preferred in high-order DG approximations because they do not require solving large complex nonlinear systems that result from implicit implementations and they facilitate the parallelization in multi-core systems [62]. These time-integration schemes perform well when spatial and temporal scales are comparable. However, because of their very strict CFL-related time-step restrictions, they are not efficient for solving steady-state problems and stiff equations, such as the ones encountered in turbulent flows, which limits the industrial application of high-order methods. For these reasons, there is a special interest in the high-order community to develop efficient implicit time-integration solvers and to improve multigrid and mesh adaptation strategies [14].

Note that multigrid methods can be (and have been) adapted to unsteady cases in a straightforward manner [14,63,64]. On the contrary, p-adaptation methods based on τ -estimators have only been applied to steady-state solutions in the context of high-order methods [25,23]. This issue will be addressed in future studies.

3.1. Nonlinear p -multigrid

Multigrid methods are techniques used to accelerate the convergence to the solution of large linear or nonlinear problems. They constitute a workaround to the fact that standard iterative solvers tend to reduce the high-frequency contents of the error fast, but fail to reduce the low-frequencies efficiently. For this reason, the iterative procedures are commonly referred to as *smoothers* in the multigrid parlance.

In h -multigrid, a sequence of progressively coarsening meshes is used where the iterative solver is employed. Every time the mesh is coarsened, some of the smooth components of the error become oscillatory relative to the mesh sampling. Therefore, further coarse-grid smoothing enhances the convergence rate. The p -multigrid scheme relies on the same notion, but low-order polynomial representations are used as coarse levels, which makes it very appropriate for high order methods. p -Multigrid methods typically use a fixed h -mesh.

For compactness, we shortly describe the Full Approximation Storage (FAS) nonlinear multigrid algorithm. Further details can be found in [39,65,16,14]. Let us consider the steady-state form ($\partial \mathbf{q} / \partial t = 0$) of our nonlinear problem (equation (9)):

$$[\mathbf{M}]^{-1} \mathbf{F}(\mathbf{Q}) = \mathbf{S}, \quad (19)$$

and define $\mathbf{A}(\mathbf{Q}) = [\mathbf{M}]^{-1} \mathbf{F}(\mathbf{Q})$, to obtain

$$\mathbf{A}(\mathbf{Q}) = \mathbf{S}, \quad (20)$$

and use the temporal discretization as the smoothing technique.

We select a third order Williamson's low-storage Runge–Kutta scheme (RK3) [66] as the smoothing time-marching scheme, so that the developed multigrid schemes can be compared to the purely explicit RK3 (see section 4.1.1). After some smoothing sweeps in a mesh with polynomial order P , the nonlinear residual equation holds

$$\mathbf{S}^P - \mathbf{A}^P(\tilde{\mathbf{Q}}^P) = \mathbf{r}^P, \quad (21)$$

where $\tilde{\mathbf{Q}}^P$ is the approximated solution and \mathbf{r}^P is known as the nonlinear residual. Remember that $P = (P_1, P_2, P_3)$ can be different in each element and coordinate direction. Using equation (20), equation (21) can be rewritten as

$$\mathbf{A}^P(\mathbf{Q}^P) - \mathbf{A}^P(\tilde{\mathbf{Q}}^P) = \mathbf{r}^P, \quad (22)$$

$$\mathbf{A}^P(\tilde{\mathbf{Q}}^P + \boldsymbol{\epsilon}_{it}^P) - \mathbf{A}^P(\tilde{\mathbf{Q}}^P) = \mathbf{r}^P, \quad (23)$$

where $\boldsymbol{\epsilon}_{it}^P$ is the iteration error on the mesh P . The standard two-level FAS p -multigrid scheme consists in transferring equation (22) to a lower polynomial representation of order $N = P - \Delta N$ (coarser grid), and using additional smoothing sweeps there. In the lower-order grid, the smoother now targets lower frequencies than the ones removed on the finer grid. Therefore, solving the residual equation on the coarse grid,

$$\mathbf{A}^N(\mathbf{Q}^N) - \mathbf{A}^N(\tilde{\mathbf{Q}}_0^N) = \mathbf{r}^N, \quad (24)$$

for \mathbf{Q}^N , leads to an improved low frequency approximation of the fine grid problem. This holds if $\tilde{\mathbf{Q}}_0^N$ and \mathbf{r}^N are transferred (interpolated or projected) from the fine grid:

$$\tilde{\mathbf{Q}}_0^N = \mathbf{I}_P^N \tilde{\mathbf{Q}}^P \quad (25)$$

$$\mathbf{r}^N = \mathbf{I}_P^N \mathbf{r}^P. \quad (26)$$

Here, \mathbf{I}_P^N is the restriction operator, an L_2 projection to the lower polynomial order. Note that no distinction is made between the solution and residual transfer operators since in this work both the solution and the residual are spanned in the same polynomial space. This is an advantage of our implementation since less storage is needed. It is also important to remark that the L_2 projection preserves the energy of the transferred quantities. This is an important difference with the transfer operators that are commonly employed in modal DG [16,14,17,11], which do not conserve energy since only the low-order coefficients are transferred for coarse-grid smoothing and the correction is then injected to the lower modes of the high order representation (the transfer matrices are simply identity matrices with rows or columns appended).

The coarse-grid nonlinear problem holds

$$\mathbf{A}^N(\mathbf{Q}^N) = \mathbf{S}^N, \quad (27)$$

where \mathbf{S}^N is an artificial source term that can be obtained combining equations (24), (25) and (26):

$$\mathbf{S}^N = \mathbf{A}^N(\mathbf{I}_P^N \tilde{\mathbf{Q}}^P) + \mathbf{I}_P^N \mathbf{r}^P, \quad (28)$$

which, according to equation (20), is the same as

$$\mathbf{S}^N = [\mathbf{M}]^{-1} \mathbf{F}^N (\mathbf{I}_p^N \tilde{\mathbf{Q}}^P) + \mathbf{I}_p^N \mathbf{r}^P. \quad (29)$$

After solving equation (27) for \mathbf{Q}^N using a smoothing procedure, we obtain a low frequency approximation of the iteration error:

$$\epsilon_{it}^N = \tilde{\mathbf{Q}}^N - \tilde{\mathbf{Q}}_0^N, \quad (30)$$

which is then used to update the solution on the fine grid:

$$\mathbf{Q}_{i+1}^P = \mathbf{Q}_i^P + \mathbf{I}_N^P \epsilon_{it}^N. \quad (31)$$

The two-level process described above can be generalized to a multilevel FAS V-Cycle and coded efficiently as a recursive procedure as depicted in Algorithm 1. Note that the superindex c now denotes the next coarser multigrid level and that the fine superindexes have been dropped for readability. The multigrid cycle has N_{MG} levels, where $level = 1$ is the coarsest (lowest polynomial order) and $level = N_{MG}$ is the finest (highest polynomial order).

Algorithm 1 FAS: V-cycle.

Recursive Procedure: FASVCycle($\tilde{\mathbf{Q}}, \mathbf{r}, level, N_{MG}$)

```

if  $level < N_{MG}$  then  $\mathbf{S} = \mathbf{A}(\tilde{\mathbf{Q}}) + \mathbf{r}$                                 ▷ Find coarse-grid source term (eq (28))
 $\tilde{\mathbf{Q}}_0 \leftarrow \tilde{\mathbf{Q}}$                                                          ▷ Store fine grid solution
 $\tilde{\mathbf{Q}} \leftarrow \text{Smooth}(\tilde{\mathbf{Q}}, \beta_1)$                                            ▷ Pre-smooth  $\beta_1$  times (RK3)
if  $level > 1$  then                                                    ▷ If not on the coarsest level, correct the solution using multigrid
     $\tilde{\mathbf{Q}}^c \leftarrow \mathbf{I}_f^c \tilde{\mathbf{Q}}$                                               ▷ Restrict solution to coarse grid (eq (25))
     $\mathbf{r}^c \leftarrow \mathbf{I}_f^c (\mathbf{S} - \mathbf{A}(\tilde{\mathbf{Q}}))$                                        ▷ Restrict residual to coarse grid (eq (26))
    CALL FASVCycle(  $\tilde{\mathbf{Q}}^c, \mathbf{r}^c, level - 1, N_{MG}$  )                       ▷ Recursive calling
     $\tilde{\mathbf{Q}} = \tilde{\mathbf{Q}} + \mathbf{I}_c^c \epsilon_{it}^c$                                            ▷ Correct solution using coarse-grid approximation (eq (31))
end if
 $\tilde{\mathbf{Q}} \leftarrow \text{Smooth}(\tilde{\mathbf{Q}}, \beta_2)$                                            ▷ Post-smooth  $\beta_2$  times (RK3)
if  $level < N_{MG}$  then  $\epsilon_{it} \leftarrow \tilde{\mathbf{Q}}_0 - \tilde{\mathbf{Q}}$                              ▷ Compute iteration error (eq (30))

```

In this work, we use $\Delta N = P - N = 1$ as the polynomial order reduction in every coarsening. Therefore, the number of multigrid levels corresponds to the maximum polynomial order of the mesh.

3.1.1. Multigrid cycling strategy

The typical cycling strategy used for h- and hp-multigrid implementations is to perform repeated V-cycles [17,11,12,20,52,16,13,19,18] (Fig. 1(a)). Some authors [17,65] make use of V or W saw-tooth cycles (without post-smoothing). This technique is well-suited for modal discretizations since the solution correction (equation (30)) is injected in the low-order coefficients of the fine-grid representation after coarse-grid smoothing. However, in the nodal discretizations of DGSEM (used in this paper), the coarse-grid smoothed solution can excite high-frequency modes of the fine-grid representation after the interpolation to the fine grid. In consequence, we find that post-smoothing is required.

The V-cycling strategy can be very sensitive to the initial condition. To get an appropriate initial condition in the high-order representation, the mainstream alternative is to employ a Full Multigrid (FMG) cycle (see Fig. 1(b)) at the beginning of the simulation. In this cycling strategy, some iterations are taken with the lowest polynomial order (coarsest multigrid level) and the approximated solution is projected to the next (finer) approximation. There, a number of V-cycles are taken before increasing the polynomial order again. The last step is repeated until the finest multigrid level is reached. The number of iterations in every level can be fixed [17,16] or can be tuned using a residual-based approach [11,15]. In this work, we use a residual-based approach where multiple V-cycle repetitions are taken at each level until a predefined residual is reached; when that happens, the approximation level is raised. Such a cycling strategy can be easily implemented using a recursive routine, as shown in Algorithm 2.

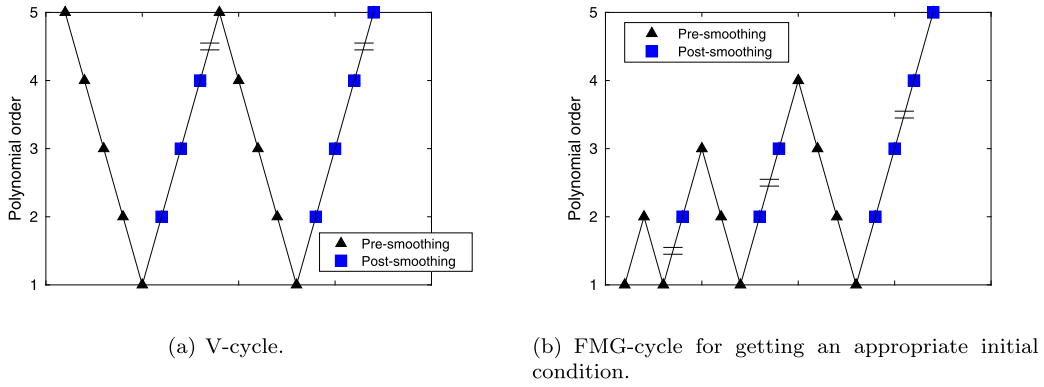


Fig. 1. FMG- and V-cycling strategies. Equal signs represent the continuation of the V-cycling process until reaching the desired residual.

Algorithm 2 FAS: FMG-cycle.

Recursive Procedure: FASFMGCycle($\tilde{\mathbf{Q}}$, level, tol)

if level > 1 **then**

$\tilde{\mathbf{Q}}^c \leftarrow \mathbf{I}_f^c \tilde{\mathbf{Q}}$

CALL FASFMGCycle($\tilde{\mathbf{Q}}^c$, level – 1, tol)

$\tilde{\mathbf{Q}} \leftarrow \mathbf{I}_f^f \tilde{\mathbf{Q}}^c$

repeat

CALL FASVCycle($\tilde{\mathbf{Q}}$, \mathbf{r} , level, level)

$\mathbf{r} \leftarrow \mathbf{S} - \mathbf{A}(\tilde{\mathbf{Q}})$

until $\|\mathbf{r}\| < \text{tol}$

else

repeat

$\tilde{\mathbf{Q}} \leftarrow \text{Smooth}(\tilde{\mathbf{Q}}, \beta)$

$\mathbf{r} \leftarrow \mathbf{S} - \mathbf{A}(\tilde{\mathbf{Q}})$

until $\|\mathbf{r}\| < \text{tol}$

end if

▷ Go to coarsest level

▷ Load coarse solution

▷ Perform V-cycle here

▷ Smooth β times (RK3)

3.1.2. Designing the smoothing

In general, the number of pre-smoothing sweeps (β_1 in Algorithm 1) must be high enough to ensure that the high-frequency modes of the error have been smoothed out, so the L_2 restriction does not introduce noise into lower multigrid levels. Likewise, the number of post-smoothing sweeps (β_2) must be high enough to guarantee that mid-frequency modes of the error do not develop into higher-order representations. These mid-frequency modes of the error can be excited by the L_2 prolongation of the solution that was smoothed in a lower multigrid level.

A common practice is to set a fixed number of pre- and post-smoothing sweeps [17,67,15,16,11]. Nonetheless, when very high polynomial orders and anisotropic non-conforming representations are used, some stages of the simulation can be very sensitive to insufficient smoothing (e.g. at the beginning of the simulation or after an adaptation stage). With that in mind, we propose two residual-based strategies for tuning the number of smoothing sweeps:

1. **Pre-smoothing:** After every β_1^0 sweeps (fixed number), the residual in the next (coarser) representation is checked. If $\|\mathbf{r}^P\|_\infty < \eta \|\mathbf{r}^N\|_\infty$, the pre-smoothing is stopped; otherwise, β_1^0 additional sweeps are performed. This strategy is a modification of the residual-based approach that some authors employ in FMG cycles for checking if the coarse level smoothing is enough [11,15]. For the simulations of this paper $\eta \leq 1.1$ showed to work fine for meshes with both uniform polynomial orders and also p-anisotropic non-conforming meshes. For this reason, all the simulations that are shown henceforth employ $\eta = 1.1$.
2. **Post-smoothing:** The norm of the residual after the post-smoothing must be at least as low as it was after the pre-smoothing, $\|\mathbf{r}_{\text{post}}^N\|_\infty \leq \|\mathbf{r}_{\text{pre}}^N\|_\infty$. This condition is checked every β_2^0 sweeps and the post-smoothing loop is exited when fulfilled. This way, we guarantee that most of the high-frequency errors that could be excited during coarse smoothing are eliminated.

3.2. p-Adaptation process

Mesh adaptation procedures aim to reduce the number of degrees of freedom of a problem retaining a comparable accuracy. Within those, p-adaptation methods work by increasing the polynomial order of the elements in regions of interest and decreasing it where low order representations are accurate enough. In the present work, we perform anisotropic

p-adaptation based on estimations of the truncation error. To do so, we need a methodology for estimating the error of anisotropic polynomial order combinations, which is summarized in next section and detailed in [1].

3.2.1. Truncation error estimation

The *non-isolated* truncation error of a discretization of order N (τ^N) is defined as the difference between the discrete partial differential operator (\mathcal{R}^N) and the exact partial differential operator (\mathcal{R}) applied to the exact solution, $\bar{\mathbf{q}}$:

$$\tau^N = \mathcal{R}^N(\mathbf{I}^N \bar{\mathbf{q}}) - \mathcal{R}(\bar{\mathbf{q}}), \quad (32)$$

where \mathbf{I}^N is a discretizing operator. For steady-state ($\partial \mathbf{q} / \partial t = 0$), the exact partial differential operator can be derived from equation (1) as

$$\mathcal{R}(\bar{\mathbf{q}}) = \mathbf{s} - \nabla \cdot \mathcal{F} = \bar{\mathbf{q}}_t = 0, \quad (33)$$

and the discrete partial differential operator can be derived point-wise from equation (9) as

$$\mathcal{R}^N(\mathbf{I}^N \bar{\mathbf{q}}) = [\mathbf{M}]\mathbf{S} - \mathbf{F}(\mathbf{I}^N \bar{\mathbf{q}}), \quad (34)$$

where \mathcal{R}^N contains the sampled values of \mathcal{R} in all the nodes of the domain and \mathbf{I}^N is a sampling operator. The *non-isolated* truncation error can then be simplified to

$$\tau^N = \mathcal{R}^N(\mathbf{I}^N \bar{\mathbf{q}}) = [\mathbf{M}]\mathbf{S} - \mathbf{F}(\mathbf{I}^N \bar{\mathbf{q}}). \quad (35)$$

In addition to the *non-isolated* truncation error, Rubio et al. [46] defined the *isolated* truncation error as

$$\hat{\tau}^N = \hat{\mathcal{R}}^N(\mathbf{I}^N \bar{\mathbf{q}}) = [\mathbf{M}]\mathbf{S} - \hat{\mathbf{F}}(\mathbf{I}^N \bar{\mathbf{q}}), \quad (36)$$

where $\hat{\mathcal{R}}^N(\cdot)$ is the *isolated* discrete partial differential operator, which is derived without substituting the flux, \mathcal{F} , by the numerical flux, \mathcal{F}^* , in equation (4), thus eliminating the influence of neighboring elements and boundaries in the truncation error of each element (see [46,1]). Rubio et al. [46] showed that the *isolated* truncation error in an element depends solely on its own polynomial order, whereas the *non-isolated* truncation error in the same element depends on the polynomial order of the element and its neighbors. In consequence, it was suggested that the *isolated* truncation error may be a better sensor for local p-adaptation than the *non-isolated* truncation error since it is not contaminated by neighbors' errors. Moreover, Rueda-Ramírez et al. [1] showed that an accurate estimation of the *isolated* truncation error imposes fewer conditions, and therefore can be computationally cheaper, than of the *non-isolated* truncation error. For these reasons, in this work we retain the isolated truncation as the driver of the proposed p-adaptation procedure.

The aim of this work is the development of a method for solving the Navier–Stokes equations in complex geometries, where the exact solution, $\bar{\mathbf{q}}$, is usually not at hand. Therefore, we utilize the τ -estimation method, which approximates the truncation error using an approximate solution on a high order grid, P , instead of the exact one. Furthermore, we are interested in a low cost approximation which suits the multigrid procedure. In consequence, we use the *quasi a-priori* approach without correction [23], which makes use of a non-converged solution, $\tilde{\mathbf{Q}}^P$:

$$\tau_P^N = [\mathbf{M}^N]\mathbf{S}^N - \mathbf{F}^N(\mathbf{I}_P^N \tilde{\mathbf{Q}}^P). \quad (37)$$

Here, the estimate of the *isolated* truncation error can be obtained by simply replacing \mathbf{F}^N by $\hat{\mathbf{F}}^N$ in equation (37). In the rest of this work, the expressions containing the symbol τ are valid for both the *non-isolated* and the *isolated* truncation error unless the contrary explicitly stated. Kompenhans et al. [23] showed that $\tilde{\mathbf{Q}}^P$ must be converged down to a residual $\tau_{\max}/10$, in order for equation (37) to yield accurate estimations of τ^N in regions where $\tau^N > \tau_{\max}$. In addition, Rueda-Ramírez et al. [1] showed that for p-anisotropic representations, the truncation error of a polynomial order combination, $N = (N_1, N_2, N_3)$, can be obtained as the sum of individual directional components:

$$\tau_{N_1 N_2 N_3} \approx \tau_1^{N_1 N_2 N_3} + \tau_2^{N_1 N_2 N_3} + \tau_3^{N_1 N_2 N_3} \approx \tau_{P_1 P_2 P_3}^{N_1 P_2 P_3} + \tau_{P_1 P_2 P_3}^{P_1 N_2 P_3} + \tau_{P_1 P_2 P_3}^{P_1 P_2 N_3}. \quad (38)$$

Each of the directional components, $\tau_i^{N_1 N_2 N_3}$, can be estimated using the τ -estimation method (equation (37)), but coarsening only in the corresponding coordinate direction, i . For example, the first component in discrete form can be obtained as

$$\tau_{P_1 P_2 P_3}^{N_1 P_2 P_3} = [\mathbf{M}^{N_1 P_2 P_3}]\mathbf{S}^{N_1 P_2 P_3} - \mathbf{F}^{N_1 P_2 P_3}(\mathbf{I}_{P_1 P_2 P_3}^{N_1 P_2 P_3} \tilde{\mathbf{Q}}^{P_1 P_2 P_3}), \quad (39)$$

where the high-order reference solution, $\tilde{\mathbf{Q}}^{P_1 P_2 P_3}$ is inserted into a lower order representation of order (N_1, P_2, P_3) . These components have spectral convergence with respect to the polynomial order in the corresponding local coordinate direction, N_i [1]. This allows obtaining accurate extrapolations of τ by extrapolating the values of τ_i with a *linear-log* regression and summing the individual contributions [1].

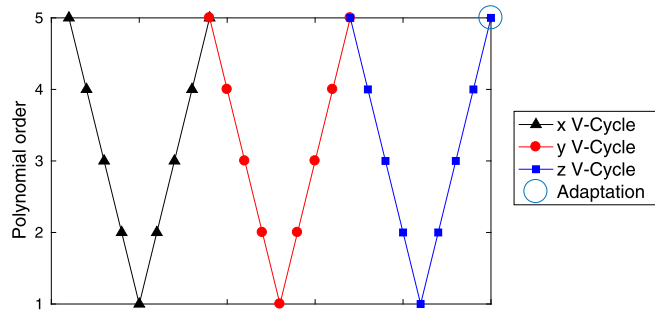


Fig. 2. Adaptation process: Anisotropic 3V FAS cycle and subsequent adaptation.

3.3. Coupling anisotropic τ -estimation-based adaptation with multigrid

In this section, we present a new technique for obtaining steady-state solutions based on coupling anisotropic p-adaptation methods and multigrid. As pointed out firstly by Brandt [39], and then recently used by Syrakos et al. [43] in the context of h-refinement techniques, the concept of truncation error arises naturally in FAS multigrid methods. In fact, the second term of our *non-isolated* truncation error estimator (equation (37)) is contained in the coarse grid source term of the multigrid scheme (equation (29)). Consequently, computing τ_p^N inside the multigrid cycle only involves a few additional operations. In the case of the *isolated* truncation error, an additional inexpensive step is required to evaluate the operator $\hat{\mathbf{F}}^N$.

Two main differences with previous τ -estimators can be identified. First, instead of interpolating the finest solution directly to every coarser representation, the solution is interpolated level by level in multigrid methods; and second, the smoothing procedure modifies the finest solution before it is transferred to lower orders. In that regard, preliminary tests showed no significant difference between the multigrid τ -estimations and the conventional ones.

Since in p-multigrid techniques the coarsening is usually performed in all coordinate directions simultaneously, i.e. the polynomial order of every local coordinate direction is decreased (isotropic multigrid), only certain combinations of low-order polynomial orders are evaluated. This makes it impossible to generate the full anisotropic truncation error map [1] (that is needed for performing anisotropic p-adaptation) with the conventional τ -estimation procedure. For this reason, in section 3.3.1 we propose a p-anisotropic multigrid procedure and explain how the anisotropic decoupled truncation error estimator by Rueda-Ramírez et al. [1] can be evaluated using such a multigrid scheme for generating the full truncation error map. In section 3.3.2, we describe how the new polynomial orders are computed based on the proposed estimations; and finally, we present a multi-stage p-adaptation process in section 3.3.4.

3.3.1. Anisotropic multigrid

The classical approach to implement a p-multigrid method is to perform coarsening in all coordinate directions simultaneously. This strategy will be referred to as isotropic multigrid in following sections. In this paper, we propose the use of an anisotropic multigrid method in which the coarsening is done in each coordinate direction at a time, in order to estimate the truncation error. For instance, in a 3D p-anisotropic multigrid case, when coarsening in ξ , the coarse grid problem is derived from equation (27) as

$$\mathbf{A}^{N_1 P_2 P_3}(\mathbf{Q}^{N_1 P_2 P_3}) = \mathbf{S}^{N_1 P_2 P_3}, \quad (40)$$

where the source term is obtained from equation (29):

$$\mathbf{S}^{N_1 P_2 P_3} = [\mathbf{M}]^{-1} \mathbf{F}^{N_1 P_2 P_3} (\mathbf{I}_{P_1 P_2 P_3}^{N_1 P_2 P_3} \mathbf{Q}^{P_1 P_2 P_3}) + \mathbf{I}_{P_1 P_2 P_3}^{N_1 P_2 P_3} \mathbf{r}^{P_1 P_2 P_3}. \quad (41)$$

The authors remark that both the isotropic and the anisotropic p-multigrid methods can be applied to p-isotropic and p-anisotropic representations indistinctively (see section 3.3.3 for more details on the generation of the coarser meshes). If the anisotropic p-multigrid method is used to solve a p-isotropic representation, the number of multigrid levels is the same in all coordinate directions, $N_{MG,i} = N_{MG,j}$. However, if it is used to solve a p-anisotropic representation, the number of multigrid levels can be different in every coordinate direction, $N_{MG,i} \neq N_{MG,j}$.

Note that this method is perfectly suited to generate the truncation error map using the decoupled truncation error estimator proposed by Rueda-Ramírez et al. [1] (equation (38)). Fig. 2 depicts the so-called anisotropic 3V FAS cycle. In every V-cycle, the coarsening is performed in one of the coordinate directions of the reference element and the directional component of the truncation error is estimated. Afterwards, the p-adaptation process that is detailed in section 3.3.2 takes place. It is noteworthy that the reference coordinate frame of an element inside a general 3D mesh is commonly not aligned with its neighbors'. This can pose a problem for the *non-isolated* truncation error estimation, but not for the *isolated* truncation error that neglects the contribution of the neighboring elements (a thorough analysis can be found in [1]).

Notice that instead of evaluating every possible combination of $N = (N_1, N_2, N_3)$ for $N_i < P_i$ (which can be a large number for 3D cases), as in [46,23], the full truncation error map is constructed from a completely decoupled approach. One important advantage of doing so is that all the storage needed for the decoupled error estimators is already allocated in the anisotropic multigrid routines and only a few inexpensive additional computations are required. Hence, the multigrid process works indeed as both solver and τ -estimator.

3.3.2. Computing the new polynomial orders

Given a truncation error threshold, τ_{max} , that needs to be achieved in a specific case, and a maximum polynomial order allowed, N_{max} , the proposed adaptation process can be summarized in six steps:

1. A high-order representation, $P = (P_1, P_2, P_3)$, is converged down to a residual $\tau_{max}/10$ using the multigrid method described in section 3.1.
2. An anisotropic multigrid procedure (section 3.3.1) is used to estimate the decoupled truncation error contribution of every coordinate direction. For instance, when coarsening in the coordinate direction (1), the contribution is:

$$\tau_1^{N_1 N_2 N_3} \approx \tau_{P_1 P_2 P_3}^{N_1 P_2 P_3} = [\mathbf{M}^{N_1 P_2 P_3}] \mathbf{S}^{N_1 P_2 P_3} - \mathbf{F}^{N_1 P_2 P_3} (\mathbf{I}_{P_1 P_2 P_3} \tilde{\mathbf{Q}}^{P_1 P_2 P_3}). \quad (42)$$

3. The *inner* truncation error map (for $N_i < P_i$) is generated using equation (38).
4. If τ_{max} can be achieved using one of these combinations, it is selected and the simulation continues.
5. If τ_{max} cannot be achieved in the *inner* truncation error map, an extrapolation procedure based on linear-log regression is conducted in each of the three coordinate directions of the decoupled truncation error (τ_i), and then the full truncation error map is generated for $P_i \leq N_i \leq N_{max,i}$.
6. If τ_{max} can be achieved using one of these combinations, it is selected. If not, $N_1 = N_2 = N_3 = N_{max}$ is selected.

In steps 4 and 6, there can be multiple combinations (N_1, N_2, N_3) that achieve $\tau < \tau_{max}$. In that case, the combination with the lowest number of degrees of freedom is selected. Notice that the two main differences with the method of Kompenhans et al. [23] are: (i) the way in which the truncation error is estimated for $N_i < P_i$ (step 3) and later for $N_i \geq P_i$ (step 5); and (ii) that if the truncation error is not achieved, the element is fully enriched in all coordinate directions, instead of in only one.

3.3.3. Uniform coarsening versus high-order coarsening

We propose two ways of obtaining the polynomial orders of the coarser representations. Let us now define the diadic tensor \mathcal{N} , which contains the polynomial orders of all the elements in a mesh:

$$\mathcal{N} = (N^1, N^2, \dots, N^e, \dots, N^K), \quad (43)$$

where K is the number of elements of the mesh, e is the element index, and $(N^e = N_1^e, N_2^e, N_3^e)$.

After the p-adaptation procedure is done, the mesh consists of elements with non-uniform anisotropic polynomial orders. Taking into account that $\Delta N = 1$ (section 3.1), the number of multigrid levels is $N_{MG} = \max(\mathcal{N}) - N_{coarse} + 1$ for the isotropic multigrid and $N_{MG,i} = \max(\mathcal{N}_i) - N_{coarse} + 1$ for the anisotropic multigrid (the latter is a function of the maximum polynomial order per coordinate direction). Let us define two ways of performing the coarsening inside a multigrid cycle:

- **Uniform coarsening:** The coarsening is performed in all elements simultaneously:

$$(N_i^e)_{level} = (N_i^e)_{level+1} - \Delta N, \quad (44)$$

except in the elements where the minimum polynomial order allowed has been reached:

$$\text{if } ((N_i^e)_{level} < N_{coarse}) \text{ then } (N_i^e)_{level} = N_{coarse}. \quad (45)$$

- **High-order coarsening:** Since the maximum polynomial order in every multigrid level can be known beforehand:

$$(N_i)_{level}^{max} = \max(\mathcal{N}_i) - \Delta N (N_{MG,i} - level), \quad (46)$$

we can coarsen only the elements that do not fulfill this condition:

$$\text{if } ((N_i^e)_{level+1} > (N_i)_{level}^{max}) \text{ then } (N_i^e)_{level} = (N_i)_{level}^{max}. \quad (47)$$

In this way only the high-order elements are coarsened. In this paper, we use $N_{coarse} = \Delta N = 1$. Therefore, equation (46) reduces to

$$(N_i)_{level}^{max} = level. \quad (48)$$

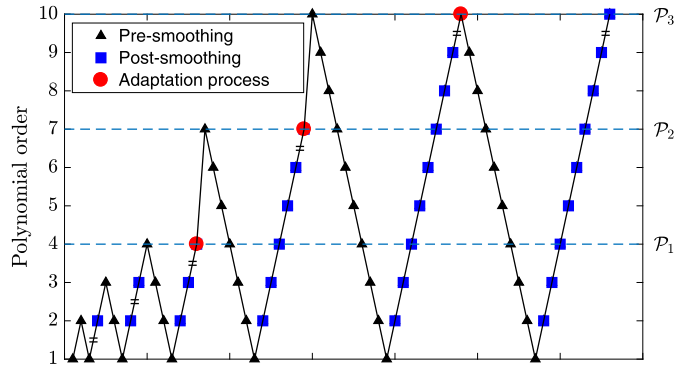


Fig. 3. Proposed FMG cycle with multiple adaptation stages. Equal signs represent the continuation of the V-cycling process until reaching the desired residual (see section 3.1.1).

Notice that equations (44) to (47) are valid for isotropic and anisotropic multigrid procedures. In the former, $N_{MG,i}$ must be simply replaced by N_{MG} , $\max(\mathcal{N}_i)$ by $\max(\mathcal{N})$, and the operations are performed in all coordinate directions. In the latter, the operations are only performed in the coordinate direction in which the coarsening is done. Furthermore, both coarsening methods are equivalent for isotropic polynomial representations.

One could argue that the *uniform coarsening* involves less computational cost than the *high-order coarsening* since coarse representations have fewer degrees of freedom. Nevertheless, the latter has two main advantages:

- Several preliminary tests showed that *uniform coarsening* could be unstable for highly anisotropic meshes in 2D and 3D.
- In 3D meshes (that are not 2D extrusions), p-nonconforming representations require the mapping order to be $M \leq N/2$ (see section 2). This means that the minimum polynomial order of the mesh must be $\min(\mathcal{N}) \geq 2$. If the *uniform coarsening* is used, the mapping restriction forces the coarsest multigrid level to have a polynomial order $N_{coarse} \geq 2$. However, if the *high-order coarsening* is used, the coarsest polynomial order can be as low as $N_{coarse} \geq 1$ since the two coarsest levels are always p-conforming. The additional coarse multigrid level helps to eliminate the low frequency components of the error.

For these reasons, in this paper we use only *high-order coarsening*.

3.3.4. Multi-stage adaptation process

The proposed *multi-stage* adaptation strategy takes advantage of an FMG-cycle and performs multiple adaptation processes at different polynomial orders (\mathcal{P}_i), as depicted in Fig. 3. In an adaptation stage at level \mathcal{P}_i (red circular markers), a τ -estimation procedure is performed using a 3V anisotropic multigrid cycle (Fig. 2), and subsequently, the polynomial orders are adjusted accordingly, but never to a polynomial order that is higher than \mathcal{P}_{i+1} . In such a case, \mathcal{P}_{i+1} is selected. This differs from the previous adaptation strategies based on τ -estimation in that, traditionally, the whole domain had to be solved in a considerably high-order mesh before performing the single-stage adaptation process. The main advantage of using this methodology is that the zones of the domain that only require a low order representation are identified early in the simulation and are not enriched. This reduces the overall computational costs.

After a truncation error estimation at the level \mathcal{P}_i , the algorithm checks if the maximum *required* polynomial order is lower or equal to the polynomial order of the next stage, \mathcal{P}_{i+1} . In such a case, the performed adaptation step is marked as the last one and the simulation continues without any further adaptation processes.

3.4. Final remarks

The isotropic versions of the p-multigrid strategy and the τ -based p-adaptation procedure can be extended to any nodal or modal DG scheme on triangular or tetrahedral elements, where the isotropic multigrid can be readily used as an isotropic τ -estimator. However, the anisotropic versions of these methods require tensor-product basis functions. To the author's knowledge, there are two kinds of mapping strategies that retain tensor-product bases in triangular elements. The most known one employs collapsed-edge quadrilaterals, as in [68,69]. Another option divides one physical triangular edge into two edges [70, Fig. 3]. The anisotropic truncation error estimator, p-multigrid procedures and adaptation strategies that are explored in this paper can be implemented in triangles/tetras/pyramids using any of these mappings.

4. Numerical results

In this section, we test the accuracy and performance of the proposed p-adaptation algorithms in 2D (section 4.1) and 3D (section 4.2) cases. For the reasons exposed in section 3.2.1, we will use the *isolated* truncation error for the p-adaptation

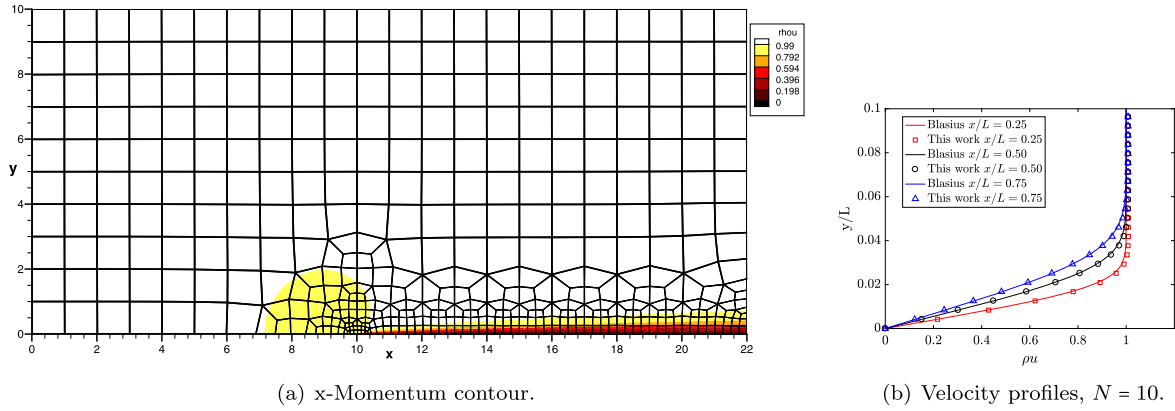


Fig. 4. Flat plate at $Re_\infty = 6000$. x-Momentum contour and velocity profiles at different positions, as compared with the exact solution by Blasius.

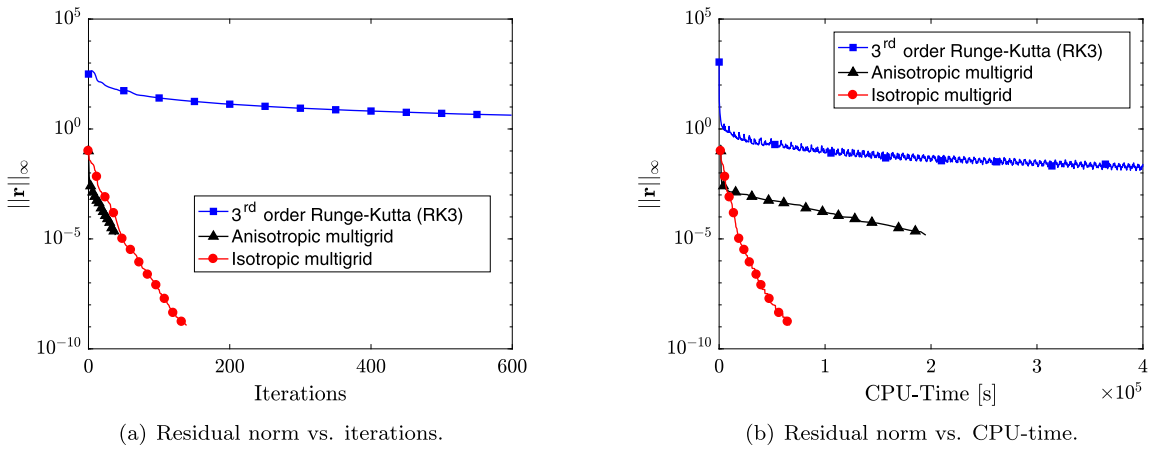


Fig. 5. Comparison of performance of a classic RK3 method and the isotropic/anisotropic FAS p-multigrid method (with RK3 as a smoother) for solving the subsonic boundary layer test case ($N = 10$).

algorithms. All the results presented in this section were obtained using an 8-core 2.6 GHz Intel Xeon E5-2670 and 32 GB RAM, and shared memory parallelization (OpenMP + guided schedule) for computing the spatial terms, as explained in [62]. Note that the parallel implementation has a near-optimal scalability for the considered discretization schemes and the selected OpenMP schedule, as can be seen in Appendix B.

4.1. 2D Flow over a flat plate

For this boundary layer test case, the mesh is constructed using 458 quadrilateral elements, and the simulations are computed with a Reynolds number of $Re_\infty = 6000$ (based on the reference length $L = 12$) and a Mach number of $M_\infty = 0.2$. Fig. 4 shows the mesh and the distribution of the momentum in the x direction, ρu .

On the boundary at $x = 0$, a uniform inflow boundary condition was imposed. On the boundary $y = 0$, $x < 10$, a free-slip boundary condition was prescribed, whereas for $y = 0$, $x \geq 10$, a no-slip adiabatic wall boundary condition emulates the effect of the flat plate. On the remaining boundaries we use a subsonic outflow boundary condition where only the far-field pressure is specified.

4.1.1. Multigrid method

First, a uniform mesh of order $N_1 = N_2 = 10$ was simulated using different solution procedures. Fig. 5 shows the infinity norm of the residual (equation (21)) as a function of the iterations and the simulation time for the classic 3rd order Runge–Kutta scheme (RK3 – in blue), an isotropic FAS p-multigrid procedure (in red), and an anisotropic FAS p-multigrid procedure (in black), both of the latter using RK3 as a smoother. All the results were obtained using $\beta_1^0 = 100$ pre-smoothing sweeps, $\beta_2^0 = 100$ post-smoothing sweeps, 400 smoothing sweeps on the coarsest multigrid level (common strategy for getting a good low-frequency representation), the smoothing tuning explained in section 3.1.2, and an FMG cycling strategy for obtaining an appropriate initial condition. As stated in section 3.1.1, a residual-based strategy is used to control when the polynomial order is increased in the FMG cycle (Algorithm 2). A fixed residual of $tol = 10^{-1}$ must be obtained before

Table 1

Computation times and speed-up for the different methods for achieving a relative drag error of at least 1.8×10^{-4} after converging until $\|\mathbf{r}\|_\infty < 10^{-9}$.

Method	CPU-time[s]	Time [%]	Speed-up
RK3	4.78×10^6	100.00%	1.00
RK3 + p-adaptation	1.02×10^5	2.14%	46.72
FAS	6.69×10^4	1.40%	71.51
FAS + p-adaptation	5.86×10^3	0.12%	815.76

the polynomial order is raised to the next FMG level. This threshold was selected because it showed good performance in preliminary tests.

It can be seen that the convergence rate of the multigrid strategies is much higher than the one of the completely explicit RK3 time integration, both in number of iterations and in CPU-time, even when the multigrid methods use the same RK3 as a smoother. Fig. 5(a) shows that the iteration performance of the RK3 method is much lower than the one exhibited by multigrid methods. Nevertheless, this is not a fair comparison since every iteration of a FAS p-multigrid method involves several iterations of the RK3. Therefore, Fig. 5(b) is a more reasonable comparison between the RK3 and the FAS p-multigrid methods. Note that the isotropic and anisotropic multigrid methods have a similar convergence rate with respect to the number of iterations, being the latter slightly better. However, when comparing the simulation times, it is remarkable that the isotropic FAS multigrid procedure is more efficient than the anisotropic one. The reason is that in the isotropic multigrid the lower multigrid levels have fewer degrees of freedom than in its anisotropic counterpart because the coarsening is done in all coordinate directions. For this reason, in next sections the anisotropic FAS will only be used as the anisotropic $\hat{\tau}$ -estimator (although during the estimation it is also used as a smoother) and the isotropic FAS will be mainly used as the solver.

4.1.2. Single-stage adaptation

In this section, we study the computational cost involved in solving the boundary layer test case for different accuracy levels. To do that, we compare the results obtained using uniform adaptation with the ones obtained using the single-stage adaptation algorithm of section 3.3.2. In every case, the two main solvers considered in this paper were analyzed (the RK3 and the FAS solver). The single-stage adaptation process was performed for $N_{max} = 10$ and $N_{max} = 5$, where a reference mesh of $P_1 = P_2 = 4$ was used. Notice that the use of such a coarse mesh as a reference mesh is now possible because of the extrapolation capacities of the new estimation algorithm [1]. After adapting the mesh, the polynomial order jump across faces is limited to

$$|N_i^+ - N_i^-| \leq 1, \quad (49)$$

where the symbols $+$ and $-$ indicate the polynomial order in the coordinate direction i of an element and its neighbor, respectively (the relative rotation between neighboring elements is taken into account). This condition provides robustness to the adapted mesh and is comparable with the *two-to-one rule* that is usually employed in h-adaptation methods [71,72]. Since the anisotropic truncation error estimator (equation (38)) has been shown to generate more accurate extrapolations of the truncation error map than conventional $\hat{\tau}$ -estimators [1], the single-stage p-adaptation method (that is used in all cases) employs a 3V anisotropic V-cycle for estimating the *isolated* truncation error, even when the time-marching solver is RK3.

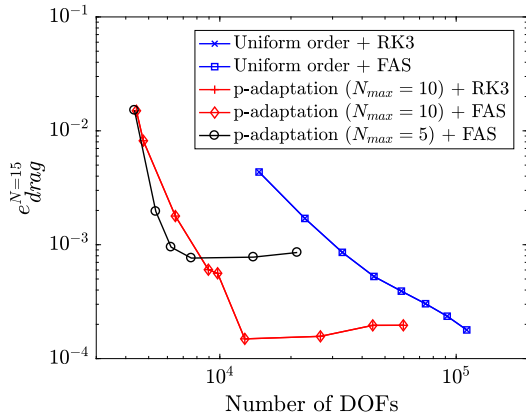
A higher-order solution of order $N_1 = N_2 = 15$ was used to estimate the relative error in the drag coefficient:

$$e_{drag}^{N=15} = \frac{|C_d - C_d^{N=15}|}{C_d^{N=15}}, \quad (50)$$

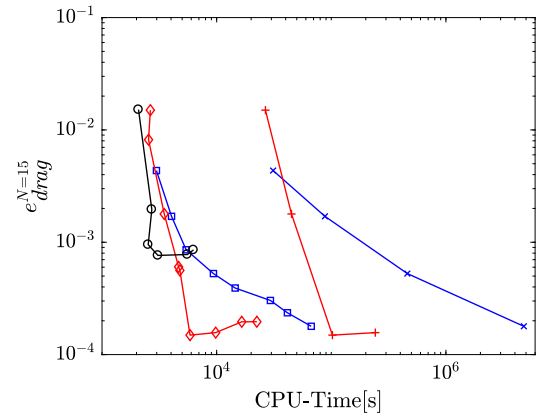
where $C_d^{N=15} = 0.211$, a value that is comparable to results in the literature [73] for a flat plate at $Re_\infty = 6000$.

The results obtained with the different methods are illustrated in Fig. 6. As shown in Fig. 6(a), the p-adapted meshes require a much fewer number of degrees of freedom for achieving a specific error than the uniformly adapted meshes. Note that the minimum relative error, that is achieved for low values of $\hat{\tau}_{max}$, tends to the relative error that corresponds to a mesh with uniform N_{max} , in the same way as the minimum $\|\hat{\tau}\|_\infty$ is a function of N_{max} (see [1]). After meeting this plateau, no further improvement in the functional error is expected. This plateau is not necessarily obtained when all elements have N_{max} , as can be seen in Fig. 7.

As can be seen in Fig. 6(b), the p-adaptation procedures are especially efficient when high accuracy is needed. Using a low N_{max} can lead to faster simulations, but the stagnation point is met sooner. It can also be observed that the most efficient procedure is the one that uses both FAS multigrid and p-adaptation. In fact, for the analyzed test case, this method achieves a better accuracy after a two hours of simulation than the classical approach (uniform order + RK3) after several days of computations. Table 1 shows the CPU-time comparison of different solution procedures for reaching a drag error of at least 1.8×10^{-4} . The speed-up is as high as 815.76.

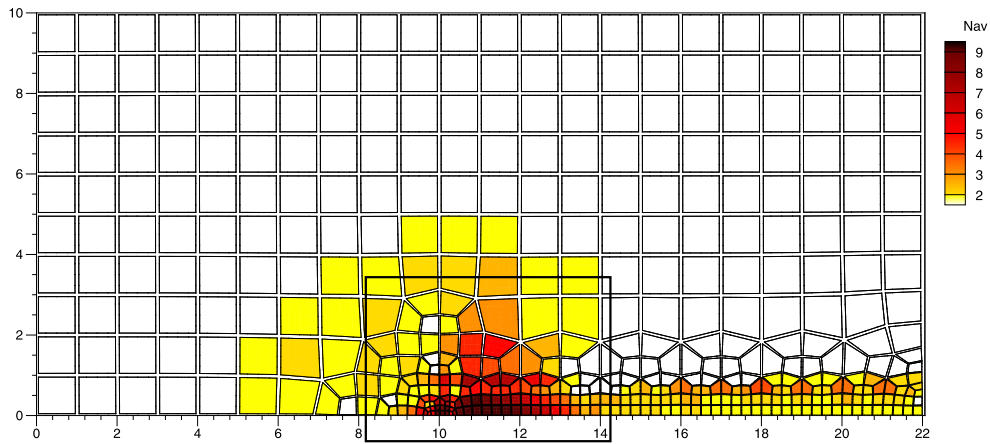
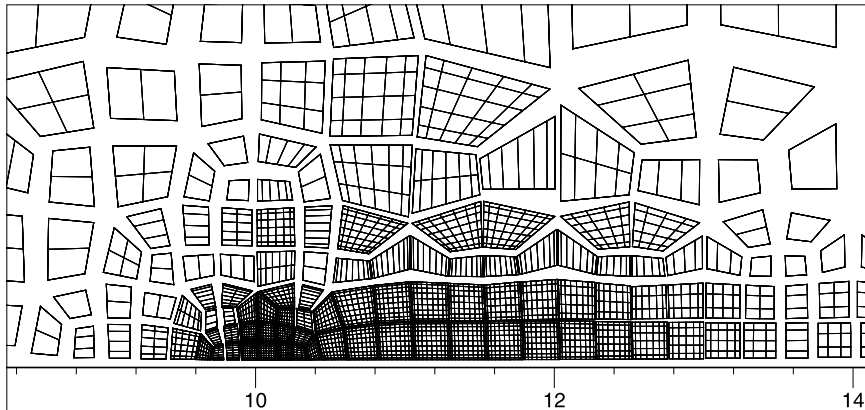


(a) Drag error vs. number of DOFs.



(b) Drag error vs. CPU-Time.

Fig. 6. Relative error in the drag coefficient calculation for different methods. The reference drag $C_d^{P=15}$ was calculated on a uniformly refined mesh with $P = 15$. The blue lines represent uniform refinement, the red lines represent the $\hat{\tau}$ -based p-adaptation procedure with $N_{max} = 10$, and the black lines with $N_{max} = 5$. Overlapping curves in (a).

(a) Average polynomial order (N_{av}).

(b) Detail of the Gauss-points.

Fig. 7. Contour indicating the final average polynomial orders after the adaptation procedure (a) and a detail of the Gauss-Points that shows the anisotropic nature of the p-adaptation method (b) for a threshold of $\hat{\tau}_{max} = 10^{-3}$, which produces a relative drag error of $e_{drag}^{N=15} = 1.49 \times 10^{-4}$. White boxes represent $N_1 = N_2 = 1.. N_{av} = (N_1 + N_2)/2$.

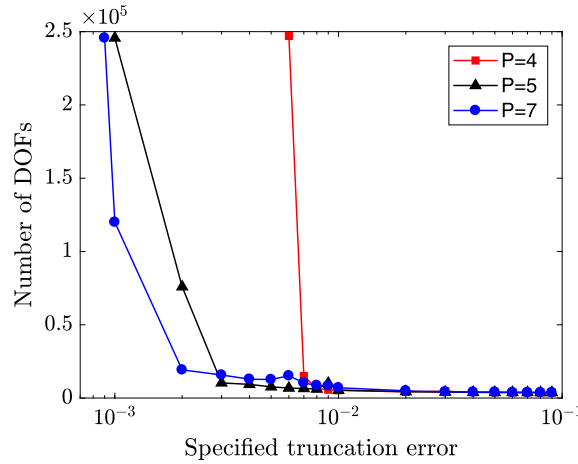


Fig. 8. Number of degrees of freedom obtained after adapting the mesh with different thresholds ($\hat{\tau}_{max}$) and different reference meshes (P) for $N_{max} = 20$.

Fig. 7 shows the final polynomial orders as computed by the proposed method for $\hat{\tau}_{max} = 10^{-3}$ (equivalent to a drag error of $e_{drag}^{N=15} = 1.49 \times 10^{-4}$). It can be seen that intensive polynomial enrichment is performed on the leading edge of the flat plate around the singularity and on the regions where the boundary layer grows, as expected. Further polynomial enrichment can be observed in regions where the mesh size changes.

4.1.3. Multi-stage adaptation

In section 3.3.4, we proposed a multi-stage p-adaptation procedure based on a full multigrid scheme with increasing polynomial orders and explored some of its theoretical advantages. Now, we apply this scheme to the boundary layer test case and analyze when it may be advantageous.

Moreover, in the last section we showed how the accuracy of the solution can be increased by choosing a higher N_{max} . Nonetheless, a higher N_{max} represents a larger truncation error map. The calculations needed for generating the larger map are not computationally intensive [1]. However, as we increase the area of the map where we have to extrapolate the values, the uncertainty of the estimations also increases.

Fig. 8 shows the number of degrees of freedom of the mesh after a single-stage adaptation procedure ($9 \times 10^{-4} \leq \hat{\tau}_{max} < 10^{-1}$ and $N_{max} = 20$) for reference meshes of different order. As can be seen, the number of DOFs increases drastically when the specified truncation error is reduced below a certain value. This behavior occurs sooner for low-order reference meshes, where some elements are over-enriched to N_{max} . In fact, the polynomial order of the reference mesh is related to the maximum polynomial order it can accurately extrapolate the truncation error to. This relation is highly dependent on the PDE being approximated and the h-size of the mesh. Preliminary tests showed that, for the cases presented in this paper, a reference mesh of order P can extrapolate accurately up to $2P$.

For high values of N_{max} , a multi-stage p-adaptation procedure becomes very useful. As was explained in section 3.3.4, instead of starting with a high-order reference mesh (which can be very expensive), a coarse reference mesh of order $P = \mathcal{P}_1$ is chosen to estimate the truncation error. With the estimation, the regions where a low-order approximation is enough are identified. Afterwards, the p-adaptation algorithm sets the polynomial orders of the mesh according to the $\hat{\tau}$ -estimation and limits the over-enrichment in more complex flow regions to \mathcal{P}_2 . In the second adaptation process at $P = \mathcal{P}_2$, and in subsequent adaptation stages, the polynomial orders of the mesh are corrected with a more accurate error estimation at hand.

In order to illustrate how this method can reduce the computational cost of highly accurate simulations, we present a comparison of the convergence of the single-stage and the multi-stage p-adaptation procedures for $\hat{\tau}_{max} = 4 \times 10^{-3}$, and $N_{max} = 20$ (Fig. 9(a)) and $N_{max} = 30$ (Fig. 9(b)). The reference meshes of the multi-stage algorithm were selected at $\mathcal{P}_1 = 4$, $\mathcal{P}_2 = 8$ and $\mathcal{P}_3 = 16$. The measured speed-up is 1.69 for $N_{max} = 20$ and 1.72 for $N_{max} = 30$ with respect to the single-stage adaptation.

4.2. 3D flow around a sphere

For this test case, the mesh is constructed with 1904 hexahedral elements, and the simulations are computed with a Reynolds number of $Re_\infty = 200$ and a Mach number of $M_\infty = 0.2$. The curvilinear hexahedral mesh has a mapping order $M = 3$ and was created using the HOPR package [74]. Fig. 10 shows the mesh and the distribution of the conserved variable ρu around the sphere.

In order to assess the properties of the representations obtained after performing $\hat{\tau}$ -based adaptation, we use a relative drag error that is computed against a high-order solution of order $N = 12$ in the same mesh:

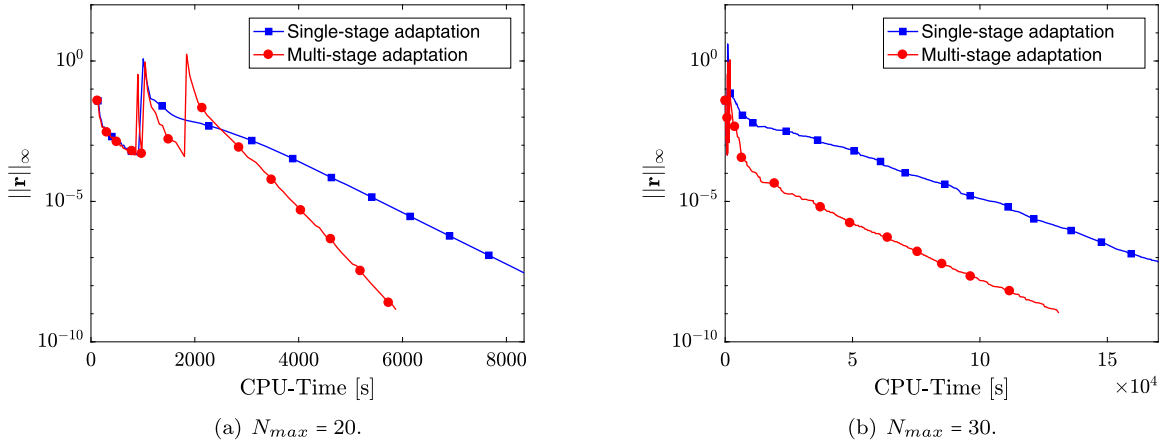


Fig. 9. Comparison of a single-stage and a multi-stage adaptation process for solving the boundary layer test case with a truncation error threshold of $\hat{\tau}_{\max} = 4 \times 10^{-3}$: $N_{\max} = 20$ (a), and $N_{\max} = 30$ (b).

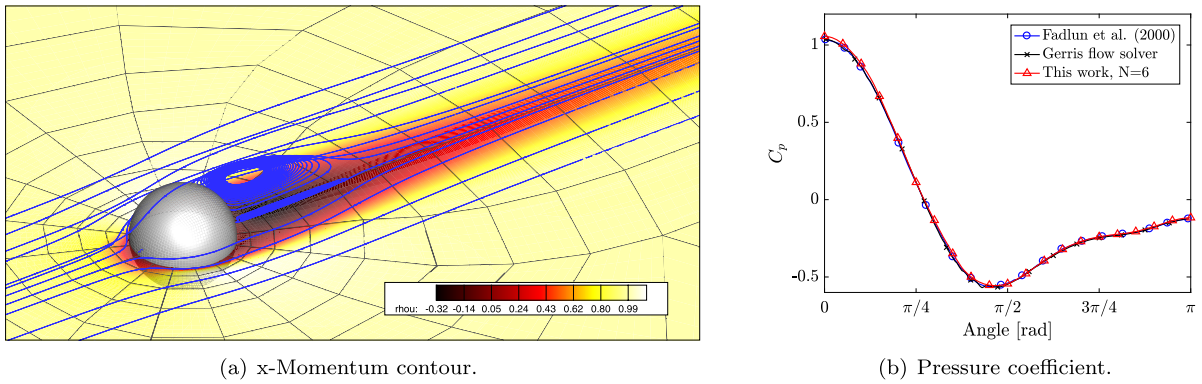


Fig. 10. Sphere at $Re_{\infty} = 200$. x-Momentum contour and pressure coefficient (C_p) as compared with data by Fadlun et al. [77] and the Gerris flow solver [78].

Table 2
Drag coefficient for sphere at $Re_{\infty} = 200$.

Author	Value
Campregher et al. [75]	0.815
Fornberg [76]	0.7683
Fadlun et al. [77]	0.7567
This work	0.7771

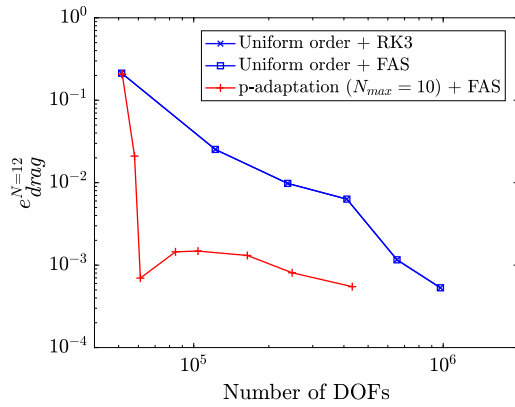
$$e_{\text{drag}}^{N=12} = \frac{|C_d - C_d^{N=12}|}{C_d^{N=12}}. \quad (51)$$

Table 2 shows a comparison between the reference drag coefficient obtained in this work and in other studies.

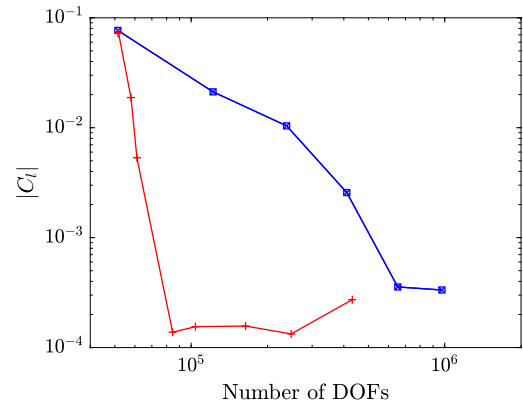
4.2.1. 3D considerations

Since in the general 3D p-nonconforming DGSEM the mapping order in every coordinate direction is limited by the solution order as $M_i \leq N_i/2$ (as indicated in section 2), and considering that the p-adapted meshes are in general p-nonconforming, the minimum polynomial order after p-adaptation is set to $N_{\min} = 2$. Additionally, taking into account the observations made in section 3.3.3, we use high-order coarsening and $N_{\text{coarse}} = 1$ for the p-multigrid method before and after p-adaptation.

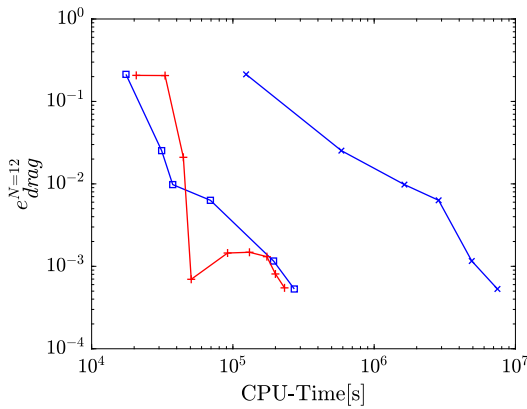
Moreover, in order to represent the curved boundary on the sphere as exactly as possible, after the p-adaptation, a *conforming algorithm* changes the polynomial orders of all elements on that surface, so that there is no polynomial order jump across their faces. This allows using a mapping of order $M_i \leq N_i$ there.



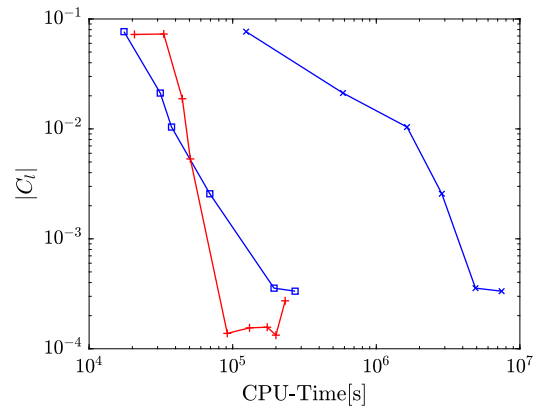
(a) Drag error vs. number of DOFs.



(b) Absolute lift vs. number of DOFs.



(c) Drag error vs. CPU-Time.



(d) Absolute lift vs. CPU-Time.

Fig. 11. Relative error in the drag and lift coefficients for different methods on the sphere. The blue lines represent uniform refinement, and the red lines represent the $\hat{\tau}$ -based p-adaptation procedure with $N_{max} = 7$.

Finally, let us remark that in 3D, the condition of equation (49) (polynomial jump across faces of 1) can cause a steep increase in the number of degrees of freedom because the polynomial enriching is transmitted in three coordinate directions. Therefore, for this test case the polynomial order jump across faces after p-adaptation is softened to

$$N_i^+ \geq \left\lfloor \frac{2}{3} N_i^- \right\rfloor, \quad (52)$$

where $\lfloor \cdot \rfloor$ is the integer part floor function.

This condition showed to provide enough robustness to the p-adapted representations and lowered the number of degrees of freedom of the adapted meshes. The conforming algorithm that is used on the sphere boundary and the algorithm that controls the polynomial order jump everywhere must be executed iteratively, until no further changes are needed, to ensure that the final mesh has all the desired properties.

4.2.2. Single-stage adaptation

The single-stage adaptation process is performed for $N_{max} = 7$, where a reference mesh of order $P_1 = P_2 = P_3 = 5$ is used. Different values of the specified truncation error threshold were tested in the range $10^{-1} \leq \hat{\tau}_{max} \leq 10^{-4}$.

The isotropic and conforming reference mesh is iterated down to a residual of $\|\mathbf{r}\|_\infty \leq \hat{\tau}_{max}/10$ using a p-multigrid algorithm with $\beta_1^0 = \beta_2^0 = 100$ pre- and post-smoothing sweeps, and 400 smoothing sweeps on the coarsest multigrid level. After the p-adaptation, the pre- and post-smoothing sweeps are $\beta_1^0 = \beta_2^0 = 50$, and the number of smoothing sweeps on the coarsest multigrid level is 200. This combination exhibited the best performance. The smoothing tuning detailed in section 3.1.2 is used and an FMG cycling strategy is employed for obtaining an appropriate initial condition with a residual of $\|\mathbf{r}\|_\infty \leq 1.0$.

The relative drag error and the absolute lift of the adapted meshes are assessed. Fig. 11 shows a comparison between the errors obtained using the $\hat{\tau}$ -based adaptation procedure and the ones using uniform p-refinement. As can be observed

Table 3Computation times and speed-up for the different methods after converging until $\|\mathbf{r}\|_\infty < 10^{-9}$.

Method	Drag coefficient ($e_{drag} \leq 5.31 \times 10^{-4}$)			Lift coefficient ($ C_l \leq 3.34 \times 10^{-4}$)		
	CPU-time[s]	Time [%]	Speed-up	CPU-time[s]	Time [%]	Speed-up
RK3	7.46×10^6	100.00%	1.00	7.46×10^6	100.00%	1.00
FAS	2.72×10^5	3.65%	27.41	2.72×10^5	3.65%	27.41
FAS + p-adaptation	4.91×10^4	0.68%	151.94	5.80×10^4	1.06%	128.55

Table 4Summary of performance for single- and multi-stage simulations with $\hat{\epsilon}_{max} = 10^{-4}$.

Adaptation strategy	DOFs(1)	DOFs(2)	CPU-Time(s)	Speed-up	e_{drag}	$ C_l $
Single-Stage: $P = 4$	1.07×10^6	–	4.53×10^5	1.00	2.27×10^{-5}	1.39×10^{-5}
Single-Stage: $P = 5$	7.90×10^5	–	2.68×10^5	1.69	3.57×10^{-5}	1.90×10^{-4}
Multi-Stage: $\mathcal{P}_1 = 4, \mathcal{P}_2 = 8$	6.20×10^5	3.85×10^5	1.75×10^5	2.59	4.50×10^{-5}	2.12×10^{-5}

in Figs. 11(a) and 11(b), the number of degrees of freedom is greatly reduced for the same accuracy when using the $\hat{\epsilon}$ -based p-adaptation. As could be expected, the maximum error in both coefficients is related to the one obtained with a uniform mesh of $N_1 = N_2 = N_3 = N_{min} = 2$. Similarly, as the *isolated* truncation error threshold $\hat{\epsilon}_{max}$ is decreased, the polynomial orders of the mesh tend to the maximum specified polynomial order, $N_{max} = 7$. Consequently, the lift coefficient also tends to $C_l^{N=7}$. Depending on specific factors of the analyzed case, such as small asymmetries, discrepancies in the error estimation, etc., the absolute lift might fall below $|C_l^{N_{max}}|$ for some intermediate representations. However, the expected value for $\hat{\epsilon} \rightarrow 0$ is $C_l^{N_{max}}$.

It is interesting to notice that, for high truncation error thresholds, the $\hat{\epsilon}$ -based adaptation does not provide an advantage in CPU-time (Figs. 11(c) and 11(d)). This is due to the cost of obtaining a semi-converged solution on the reference mesh of $P = 5$, for cases where the final polynomial order post-adaptation is $N < 5$. Additionally, let us remark that the rate of convergence in CPU-time deteriorates after the p-adaptation. This is because the p-anisotropic nonconforming representations are *more difficult* to solve. Further investigation on multigrid, or other solution methods, could improve the speed-ups here observed.

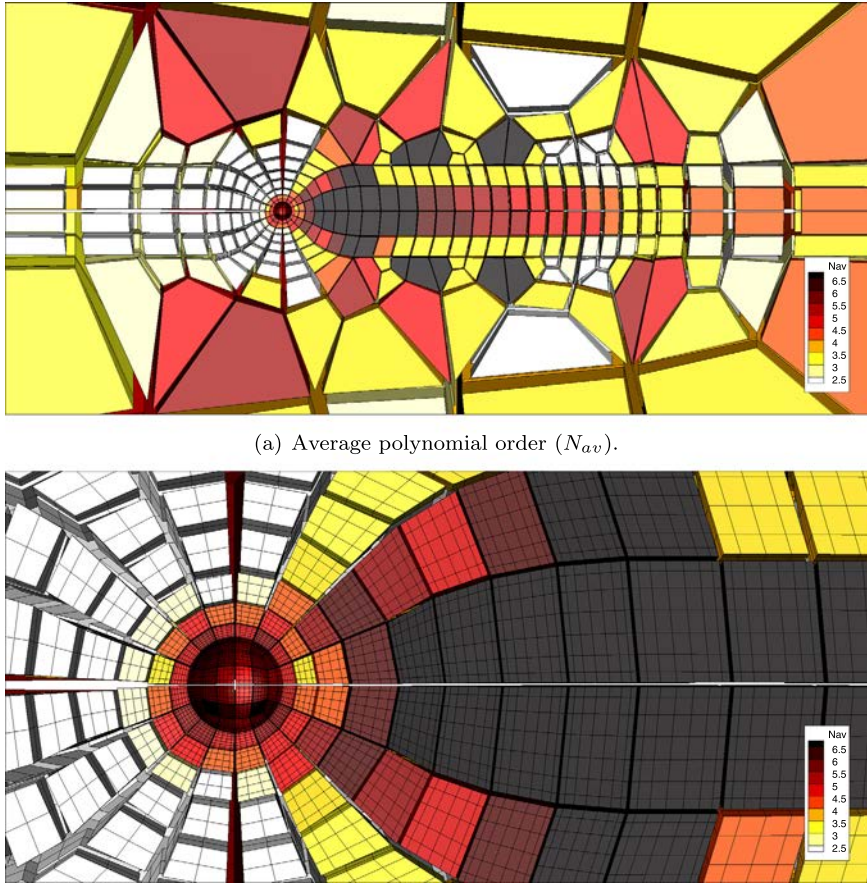
Using the data provided by Figs. 11(c) and 11(d), it is possible to compute the speed-up as a function of the drag or lift errors. Table 3 shows the computation times and speed-ups achieved for the lowest error obtained in lift and drag ($e_{drag}^{N=12}$ and $|C_l|$). The maximum speed-up is 151.94 for this 3D challenging case.

Fig. 12 illustrates the polynomial order distribution after p-adaptation for $\hat{\epsilon}_{max} = 4 \times 10^{-4}$, which corresponds to a drag error of $e_{drag} = 8.08 \times 10^{-4}$ and an absolute lift of $|C_l| = 1.33 \times 10^{-4}$. It can be seen that intensive polynomial enrichment is performed on the recirculation bubble, the wake, and on the boundary layer, as expected. Further polynomial enrichment can be observed in regions where the mesh size changes drastically. In particular, we observe that the polynomial enrichment is higher on the region that connects the wake with the recirculation bubble than on the boundary layer. Indeed, the base mesh showed large elements downstream of the sphere, which required local refinement. This over-enrichment of the recirculation bubble increases the number of degrees of freedom and may explain why the drag and lift are not as efficiently controlled here as in the flat plate case (the speed-up is not as high). Namely, the zone that is expected to contribute the most to the lift and drag is the boundary layer, which is enriched to a lesser extent. The slight asymmetries are due to small discrepancies in the error estimation and the fact that the polynomial order is a discrete function (a slight difference in the error estimation can cause important differences in polynomial orders). In addition, the relative rotation between neighboring elements can also affect the truncation error estimation in the anisotropic coarsening on a small scale. Although this effect is more important for the *non-isolated* truncation error estimation, it is also weakly present for the *isolated* truncation error. For more details on the effects of neighboring elements, we refer to [1].

4.2.3. Multi-stage adaptation

The multi-stage adaptation procedure introduced in section 3.3.4 becomes useful when the maximum allowable polynomial order after adaptation (N_{max}) is increased and the specified *isolated* truncation threshold ($\hat{\epsilon}_{max}$) is low. In this section, we use the multi-stage p-adaptation procedure on the sphere test case and set the maximum polynomial order after adaptation to $N_{max} = 11$, the truncation error threshold to $\hat{\epsilon}_{max} = 10^{-4}$, and the adaptation stages to $\mathcal{P}_1 = 4$ and $\mathcal{P}_2 = 8$. Fig. 13 shows a comparison of performance (in CPU-Time) between the multi-stage p-adaptation procedure and two single-stage procedures with $P = 4$ and $P = 5$. The maximum polynomial order after adaptation in the single stage cases is also $N_{max} = 11$.

As can be observed, the convergence rate (with respect to CPU-time) of the multi-stage p-adapted mesh is higher than for single-stage p-adapted meshes since the former has fewer degrees of freedom. For this reason, the multi-stage p-adapted mesh is able to achieve a speed-up of 2.69 in comparison to the single-stage method to converge to a near-round-off residual $\|\mathbf{r}\|_\infty = 10^{-9}$. Table 4 shows a summary of results for the simulations of this section.



(a) Average polynomial order (N_{av}).

Fig. 12. Contours indicating the final polynomial orders after p-adaptation for the sphere test case: Average polynomial orders (N_{av}) (a) and a detail of the Gauss-Points that shows the anisotropic nature of the p-adaptation method (b) for a threshold of $\hat{\epsilon}_{max} = 4 \times 10^{-4}$. White boxes represent $N_1 = N_2 = N_3 = 2$. $N_{av} = (N_1 + N_2 + N_3)/3$.

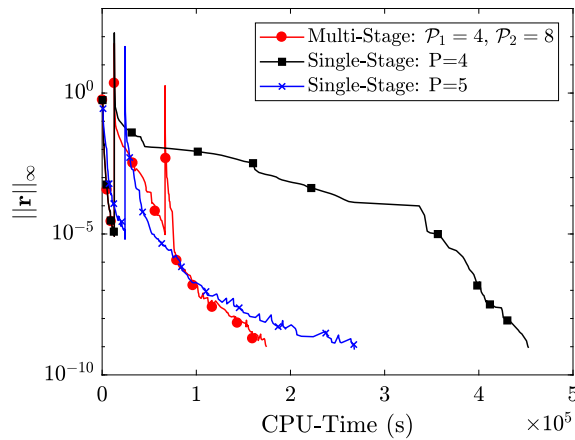


Fig. 13. Comparison of single-stage ($P = 4$ and $P = 5$) and multi-stage adaptation ($\mathcal{P}_1 = 4, \mathcal{P}_2 = 8$) processes for the sphere. $N_{max} = 11$, $\hat{\epsilon}_{max} = 10^{-4}$.

The number of degrees of freedom for the single-stage $P = 4$ is the highest, since in that case many elements are enriched to the maximum $N_1 = N_2 = N_3 = 11$ due to problems in the error estimation (as explained in section 4.1.3). In the single-stage $P = 5$ this behavior is also observed, but to a lesser extent. On the other hand, in the multi-stage case the number of degrees of freedom in the first stage is limited by the condition $N_{max,1} = 8$, and the distribution of polynomial orders is then corrected in the second stage, where the number of degrees of freedom decreases, even though the maximum

polynomial order is $N_{\max,2} = 11$. It is remarkable that the multi-stage adapted mesh can achieve comparable drag and lift errors with about one third of the degrees of freedom and a speed-up of 2.59 with respect to the single-stage $P = 4$.

5. Conclusions

In this paper, we have developed a coupled solver using truncation error estimators, anisotropic p-adaptation and multi-grid. The most important conclusions of this work are:

1. A novel anisotropic p-adaptation multigrid algorithm is presented which uses the multigrid method both as a solver and as a truncation error estimator.
2. The coupling of single-stage p-adaptation strategies and multigrid methods resulted in a speed-up of 816 for a 2D boundary layer case and of 152 for the 3D sphere case.
3. The technique for evaluating the truncation error by Rueda-Ramírez et al. [1] can be performed directly inside an anisotropic multigrid procedure needing only a few additional operations.
4. Isotropic multigrid methods show better performance than anisotropic multigrid methods. The reason is that the successive coarse grids are cheaper to compute when the polynomial order is reduced in all coordinate directions.
5. A multi-stage p-adaptation technique based on coupling τ -estimations and multigrid was developed. Experiments show that multi-stage is advantageous for highly accurate simulations compared with single-stage adaptation procedures. The multi-stage procedure showed to be a promising alternative for 3D simulations, since coarser reference meshes can be used: the elements that do not need to be enriched are identified early and their polynomials are frozen in a low value. The achieved speed-ups with this method were as high as 2.59 with respect to the single-stage adaptation.

Acknowledgements

The authors would like to thank David Kopriva for his friendly advice and cooperation. This project has received funding from the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie grant agreement No 675008 for the SSeMID project (A.M. Rueda-Ramírez, E. Valero). This work has been partially supported by Ministerio de Economía y Competitividad under the research grant TRA2015-67679-C2-2-R (J. Manzanero, E. Ferrer, G. Rubio).

The authors acknowledge the computer resources and technical assistance provided by the *Centro de Supercomputación y Visualización de Madrid* (CeSViMa).

Appendix A. The Navier–Stokes equations

The compressible Navier–Stokes equations in conservative form can be written in non-dimensional form as

$$\mathbf{q}_t + \nabla \cdot (\mathcal{F}^a - \mathcal{F}^v) = \mathbf{s}, \quad (53)$$

where the conserved variables are $\mathbf{q} = (\rho, \rho u, \rho v, \rho w, \rho e)^T$, \mathbf{s} is an external source term, and \mathcal{F}^a and \mathcal{F}^v are called the advective and diffusive flux tensors, respectively, which depend on \mathbf{q} . Expanding the fluxes in Cartesian coordinates leads to the expression,

$$\mathbf{q}_t + \mathbf{f}_x^a + \mathbf{g}_y^a + \mathbf{h}_z^a - \frac{1}{\text{Re}_\infty} (\mathbf{f}_x^v + \mathbf{g}_y^v + \mathbf{h}_z^v) = \mathbf{s}. \quad (54)$$

Here, $\text{Re}_\infty = V_\infty L_\infty \rho_\infty / \mu_\infty$ is the Reynolds number in the far-field. The advective fluxes are then defined as

$$\mathbf{f}^a = \begin{bmatrix} \rho u \\ p + \rho u^2 \\ \rho uv \\ \rho uw \\ u(\rho e + p) \end{bmatrix}, \mathbf{g}^a = \begin{bmatrix} \rho v \\ \rho uv \\ p + \rho v^2 \\ \rho vw \\ v(\rho e + p) \end{bmatrix}, \mathbf{h}^a = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ p + \rho w^2 \\ w(\rho e + p) \end{bmatrix}, \quad (55)$$

where the pressure p is computed using the calorically perfect gas approximation. On the other hand, the diffusive fluxes are defined as

$$\mathbf{f}^v = \begin{bmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} + \frac{\kappa}{(\gamma-1)\text{Pr}_\infty \text{M}_\infty^2} T_x \end{bmatrix}, \quad (56)$$

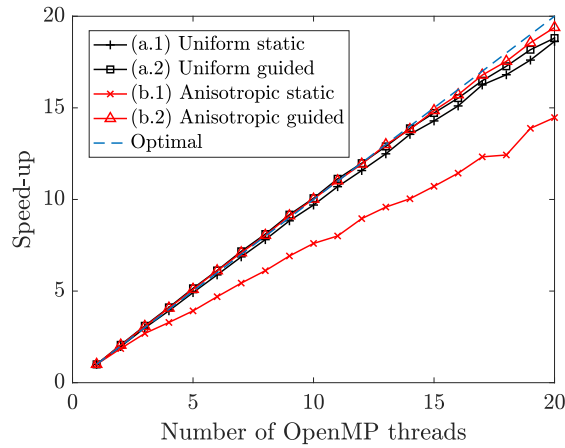


Fig. 14. Strong scalability test.

$$\mathbf{g}^v = \begin{bmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{yx} + v\tau_{yy} + w\tau_{yz} + \frac{\kappa}{(\gamma-1)\text{Pr}_\infty M_\infty^2} T_y \end{bmatrix}, \quad (57)$$

$$\mathbf{h}^v = \begin{bmatrix} 0 \\ \tau_{zx} \\ \tau_{zy} \\ \tau_{zz} \\ u\tau_{zx} + v\tau_{zy} + w\tau_{zz} + \frac{\kappa}{(\gamma-1)\text{Pr}_\infty M_\infty^2} T_z \end{bmatrix}, \quad (58)$$

where T is the temperature, T_i its spatial derivatives, γ is the heat capacity ratio, and κ is the thermal diffusivity. The nondimensional far-field parameters are the Prandtl number, $\text{Pr}_\infty = c_p \mu_\infty / \kappa_\infty$; and the Mach number, $M_\infty = \|\mathbf{v}\|/c$. The stress tensor components are computed using the Stokes hypothesis,

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right), \quad i \neq j \quad (59)$$

$$\tau_{ii} = 2\mu \frac{\partial v_i}{\partial x_i} + \lambda \nabla \cdot \mathbf{v}, \quad (60)$$

with μ the fluid's viscosity, $\lambda = -\frac{2}{3}\mu$ the bulk viscosity coefficient, and \mathbf{v} the flow velocity. For the simulations in this paper we chose the typical parameters for air: $\text{Pr} = 0.72$, $\gamma = 1.4$, while μ and κ are calculated using Sutherland's law.

Appendix B. Scalability test

In this work, we use a shared-memory parallelization (OpenMP) that consists in distributing the most expensive tasks (i.e. the volume and surface integral computations) to different processing units. In this section, a strong scalability test is presented for the sphere test case ($\text{Re}_\infty = 200$, section 4.2). Two common OpenMP schedules were tested: static and guided. The static schedule consists in dividing the loops of the elements and faces into equal-sized chunks (or as equal as possible) for the different available threads (i.e. $\text{chunk_size} = \text{num_of_elements}/\text{num_of_threads}$ and $\text{chunk_size} = \text{num_of_faces}/\text{num_of_threads}$, respectively). On the other hand, the guided schedule gives a smaller chunk of loop iterations to each thread. When a thread finishes, it retrieves another chunk to compute. The chunk size starts large and decreases as the loop computation is completed. The computation time needed for taking 100 iterations (RK3) was measured using a 2-socket Intel CPU with 2×20 cores at 2.2 GHz and 529 GB RAM. Two representations with roughly the same number of DOFs ($\pm 5\%$) were used:

- (a) A uniform order ($N_1 = N_2 = N_3 = 5$) representation (411264 DOFs): $e_{\text{drag}}^{N=15} = 6.32 \times 10^{-3}$, $|C_l| = 2.57 \times 10^{-3}$.
- (b) The p-anisotropic representation that was obtained for $\hat{\tau} = 10^{-4}$, which provided an improved accuracy (comparable to the $N = N_{\text{max}} = 7$ representation) for a similar number of degrees of freedom (431667 DOFs): $e_{\text{drag}}^{N=15} = 5.48 \times 10^{-4}$, $|C_l| = 2.73 \times 10^{-4}$.

Each of the simulations was run five times and the computation time was averaged. The results are shown in Fig. 14.

As can be seen in Fig. 14, the code has a near-optimal scalability when using OpenMP + guided schedule. The anisotropic p-adapted representation performs suboptimally when the OpenMP schedule is static. However, in OpenMP the guided schedule provides an efficient load balancing.

Regarding the spatial discretization, Roe is used as the inviscid Riemann Solver and BR1 as the viscous Riemann Solver, as in the rest of the paper. The authors remark that the BR1 discretization has been shown to be non-compact [79], which can deteriorate performance in parallel MPI implementations (more communication is needed) but has a negligible effect when considering a shared-memory parallelization.

References

- [1] A.M. Rueda-Ramírez, G. Rubio, E. Ferrer, E. Valero, Truncation error estimation in the p-anisotropic discontinuous Galerkin spectral element method, *J. Sci. Comput.* (2018), <https://doi.org/10.1007/s10915-018-0772-0>, arXiv:1806.08439.
- [2] E. Ferrer, An interior penalty stabilised incompressible discontinuous Galerkin–Fourier solver for implicit large eddy simulations, *J. Comput. Phys.* 348 (2017) 754–775, <https://doi.org/10.1016/j.jcp.2017.07.049>.
- [3] J. Manzanero, E. Ferrer, G. Rubio, E. Valero, On the role of numerical dissipation in stabilising under-resolved turbulent simulations using discontinuous Galerkin methods, *J. Comput. Phys.* (2018), submitted for publication.
- [4] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order CFD methods: current status and perspective, *Int. J. Numer. Methods Fluids* 72 (8) (2013) 811–845, <https://doi.org/10.1002/fld.3767>.
- [5] F. Frayse, C. Redondo, G. Rubio, E. Valero, Upwind methods for the Baer–Nunziato equations and higher-order reconstruction using artificial viscosity, *J. Comput. Phys.* 326 (2016) 805–827, <https://doi.org/10.1016/j.jcp.2016.09.017>.
- [6] O.M.F. Browne, G. Rubio, E. Ferrer, E. Valero, Sensitivity analysis to unsteady perturbations of complex flows: a discrete approach, *Int. J. Numer. Methods Fluids* 76 (12) (2014) 1088–1110, <https://doi.org/10.1002/fld.3962>.
- [7] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion systems, *SIAM J. Numer. Anal.* 35 (6) (1998) 2440–2463, <https://doi.org/10.1137/S0036142997316712>.
- [8] B. Rivière, *Spectral Methods for Solving Elliptic and Parabolic Equations Theory and Implementation*, SIAM, 2008.
- [9] D.A. Kopriva, S.L. Woodruff, M.Y. Hussaini, Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method, *Int. J. Numer. Methods Eng.* 53 (1) (2002) 105–122, <https://doi.org/10.1002/nme.394>.
- [10] E. Ferrer, R.H. Willden, A high order discontinuous Galerkin–Fourier incompressible 3D Navier–Stokes solver with rotating sliding meshes, *J. Comput. Phys.* 231 (21) (2012) 7037–7056, <https://doi.org/10.1016/j.jcp.2012.04.039>.
- [11] K.J. Fidkowski, T.A. Oliver, J. Lu, D.L. Darmofal, p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, *J. Comput. Phys.* 207 (1) (2005) 92–113, <https://doi.org/10.1016/j.jcp.2005.01.005>.
- [12] H. Luo, J.D. Baum, R. Löhner, A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, *J. Comput. Phys.* 211 (2) (2006) 767–783, <https://doi.org/10.1016/j.jcp.2005.06.019>.
- [13] H. Luo, J.D. Baum, R. Löhner, A fast, p-multigrid discontinuous Galerkin method for compressible flows at all speeds, *AIAA J.* 46 (3) (2008) 635–652.
- [14] L. Wang, D.J. Mavriplis, Implicit solution of the unsteady Euler equations for high-order accurate discontinuous Galerkin discretizations, *J. Comput. Phys.* 225 (2) (2007) 1994–2015, <https://doi.org/10.1016/j.jcp.2007.03.002>.
- [15] P.T.F. Bassi, A. Ghidoni, S. Rebay, High-order accurate p-multigrid discontinuous Galerkin solution of the Euler Equations, *Int. J. Numer. Methods Fluids* 60 (2009) 847–865, <https://doi.org/10.1002/fld.1917>.
- [16] C.R. Nastase, D.J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, *J. Comput. Phys.* 213 (2006) 330–357, <https://doi.org/10.1016/j.jcp.2005.08.022>.
- [17] K. Shahbazi, D.J. Mavriplis, N.K. Burgess, Multigrid algorithms for high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations, *J. Comput. Phys.* 228 (21) (2009) 7917–7940, <https://doi.org/10.1016/j.jcp.2009.07.013>.
- [18] L. Wang, D. Mavriplis, Adjoint-based h–p adaptive discontinuous Galerkin methods for the compressible Euler equations, *J. Comput. Phys.* 228 (20) (2009) 7643–7661.
- [19] W.F. Mitchell, C.S. Division, The hp-multigrid method applied to hp-adaptive refinement of triangular grids, *Numer. Linear Algebra Appl.* 17 (1) (2010) 211–228, <https://doi.org/10.1002/nla.700>.
- [20] L. Botti, A. Colombo, F. Bassi, h-Multigrid agglomeration based solution strategies for discontinuous Galerkin discretizations of incompressible flow problems, *J. Comput. Phys.* 347 (2017) 382–415, arXiv:1703.03592.
- [21] D. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*, Springer Science & Business Media, 2009.
- [22] K. Black, A conservative spectral element method for the approximation of compressible fluid flow, *Kybernetika* 35 (1) (1999) 133–146.
- [23] M. Kompenhans, G. Rubio, E. Ferrer, E. Valero, Adaptation strategies for high order discontinuous Galerkin methods based on Tau-estimation, *J. Comput. Phys.* 306 (2016) 216–236, <https://doi.org/10.1016/j.jcp.2015.11.032>.
- [24] F. Frayse, E. Valero, J. Ponsín, Comparison of mesh adaptation using the adjoint methodology and truncation error estimates, *AIAA J.* 50 (9) (2012) 1920–1932, <https://doi.org/10.2514/1.j051450>.
- [25] M. Kompenhans, G. Rubio, E. Ferrer, E. Valero, Comparisons of p-adaptation and discretisation-errors for high order discontinuous Galerkin methods, *Comput. Fluids* 139 (2016) 36–46, <https://doi.org/10.1016/j.compfluid.2016.03.026>.
- [26] F. Naddei, M. De La Llave Plata, V. Couaillier, F. Coquel, A comparison of refinement indicators for the p-adaptive simulation of steady and unsteady flows with discontinuous Galerkin methods, *J. Comput. Phys.* 376 (2019) 508–533, <https://doi.org/10.1016/j.jcp.2018.09.045>.
- [27] D. Estep, A posteriori error bounds and global error control for approximation of ordinary differential equations, *SIAM J. Numer. Anal.* 32 (1) (1995) 1–48.
- [28] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, *J. Comput. Phys.* 183 (2) (2002) 508–532, <https://doi.org/10.1006/jcph.2002.7206>.
- [29] R. Hartmann, Error estimation and adjoint-based adaptation in aerodynamics, in: *European Conference on Computational Fluid Dynamics*, 2006, pp. 1–14.
- [30] D.A. Venditti, D.L. Darmofal, Grid adaptation for functional outputs: application to two-dimensional inviscid flows, *J. Comput. Phys.* 176 (1) (2002) 40–69, <https://doi.org/10.1006/jcph.2001.6967>.
- [31] N.A. Pierce, M.B. Giles, Adjoint and defect error bounding and correction for functional estimates, *J. Comput. Phys.* 200 (2) (2004) 769–794, <https://doi.org/10.1016/j.jcp.2004.05.001>.
- [32] C. Mavriplis, Nonconforming Discretizations and a Posteriori Error Estimators for Adaptive Spectral Element Techniques, Ph.D. thesis, Massachusetts Institute of Technology, 1989, <http://hdl.handle.net/1721.1/14526>.

- [33] C. Mavriplis, Adaptive mesh strategies for the spectral element method, *Comput. Methods Appl. Mech. Eng.* 116 (1–4) (1994) 77–86, <http://www.sciencedirect.com/science/article/pii/S0045782594800103>.
- [34] Eva Casoni y, Antonio Huerta, Shock capturing for discontinuous Galerkin methods, <http://www.tesisenred.net/handle/10803/51571>, 2011.
- [35] L. Krivodonova, J.E. Flaherty, Error estimation for discontinuous Galerkin solutions of multidimensional hyperbolic problems, *Adv. Comput. Math.* 19 (2003) 57–71, <https://doi.org/10.1023/A:1022894504834>.
- [36] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugnon, J.E. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, *Appl. Numer. Math.* 48 (3–4) (2004) 323–338, <https://doi.org/10.1016/j.apnum.2003.11.002>, <http://linkinghub.elsevier.com/retrieve/pii/S0168927403001831>.
- [37] G. Kuru, M. De la Llave Plata, An adaptive variational multiscale discontinuous Galerkin method for large eddy simulation, in: 54th AIAA Aerospace Sciences Meeting, 2016, 0584.
- [38] P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, in: 44th AIAA Aerospace Sciences Meeting and Exhibit, 2006.
- [39] A. Brandt, O.E. Livne, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, revised edition, SIAM, 2011.
- [40] M.J. Berger, Adaptive finite difference methods in fluid dynamics, in: *Computational Fluid Dynamics*, Von Karman Inst. for Fluid Dynamics, 1987, pp. 8–34.
- [41] F. Frayssé, E. Valero, G. Rubio, Quasi-a priori truncation error estimation and higher order extrapolation for non-linear partial differential equations, *J. Comput. Phys.* 253 (2013) 389–404, <https://doi.org/10.1016/j.jcp.2013.07.018>.
- [42] F. Frayssé, G. Rubio, J. De Vicente, E. Valero, Quasi-a priori mesh adaptation and extrapolation to higher order using τ -estimation, *Aerosp. Sci. Technol.* 38 (2014) 76–87, <https://doi.org/10.1016/j.ast.2014.07.017>.
- [43] A. Syrakos, G. Efthimiou, J.G. Bartzis, A. Goulas, Numerical experiments on the efficiency of local grid refinement based on truncation error estimates, *J. Comput. Phys.* 231 (20) (2012) 6725–6753, <https://doi.org/10.1016/j.jcp.2012.06.023>, arXiv:1508.02345.
- [44] A. Syrakos, A. Goulas, Finite volume adaptive solutions using SIMPLE as smoother, *Int. J. Numer. Methods Fluids* 52 (11) (2006) 1215–1245, <https://doi.org/10.1002/fld.1228>.
- [45] G. Rubio, F. Frayssé, J. De Vicente, E. Valero, The estimation of truncation error by τ -estimation for Chebyshev spectral collocation method, *J. Sci. Comput.* 57 (1) (2013) 146–173, <https://doi.org/10.1007/s10915-013-9698-8>.
- [46] G. Rubio, F. Frayssé, D.A. Kopriva, E. Valero, Quasi-a priori truncation error estimation in the DGSEM, *J. Sci. Comput.* 64 (2) (2015) 425–455, <https://doi.org/10.1007/s10915-014-9938-6>.
- [47] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (138) (1977) 333–390, <https://doi.org/10.1090/S0025-5718-1977-0431719-X>.
- [48] M. Hortmann, M. Perić, G. Scheuerer, Finite volume multigrid prediction of laminar natural convection: bench-mark solutions, *Int. J. Numer. Methods Fluids* 11 (2) (1990) 189–207, <https://doi.org/10.1002/fld.1650110206>.
- [49] H.-J. Leister, M. Perić, Numerical simulation of a 3D Czochralski-melt flow by a finite volume multigrid-algorithm, *J. Cryst. Growth* 123 (1992) 567–574, [https://doi.org/10.1016/0022-0248\(92\)90618-S](https://doi.org/10.1016/0022-0248(92)90618-S).
- [50] H.K. Versteeg, W. Malalasekera, *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*, Pearson Education, 2007.
- [51] A.W. Craig, O.C. Zienkiewicz, A multigrid algorithm using a hierarchical finite element basis, in: *Multigrid Methods for Integral and Differential Equations*, Clarendon Press, Oxford, 1985, pp. 301–312.
- [52] E.M. Rønquist, A.T. Patera, Spectral element multigrid, I: formulation and numerical results, *J. Sci. Comput.* 2 (4) (1987) 389–406, <https://doi.org/10.1007/BF01061297>.
- [53] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
- [54] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*, Springer Science & Business Media, 2013.
- [55] J. Manzanero, A.M. Rueda-Ramírez, G. Rubio, E. Ferrer, The Bassi Rebay 1 scheme is a special case of the symmetric interior penalty formulation for discontinuous Galerkin discretisations with Gauss–Lobatto points, *J. Comput. Phys.* 363 (2018) 1–10, <https://doi.org/10.1016/j.jcp.2018.02.035>.
- [56] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, *J. Comput. Phys.* 43 (2) (1981) 357–372, [https://doi.org/10.1016/0021-9991\(81\)90128-5](https://doi.org/10.1016/0021-9991(81)90128-5), arXiv:1011.1669v3.
- [57] F. Bassi, S. Rebay, A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations, *J. Comput. Phys.* 131 (1997) 267–279, <https://doi.org/10.1006/jcph.1996.5572>, <http://isn-csm.mit.edu/literature/1997-jcp-bassi.pdf>.
- [58] D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes, *J. Sci. Comput.* 26 (3) (2006) 301–327, <https://doi.org/10.1007/s10915-005-9070-8>.
- [59] D.A. Kopriva, F. Hindenlang, T. Boleman, G.J. Gassner, Free-stream preservation for curved geometrically non-conforming discontinuous Galerkin spectral elements, arXiv:1809.05206.
- [60] G. Karniadakis, S. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, Oxford University Press, 2013.
- [61] J.S. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, Springer Science & Business Media, 2007.
- [62] F. Hindenlang, G.J. Gassner, C. Altmann, A. Beck, M. Staudenmaier, C.D. Munz, Explicit discontinuous Galerkin methods for unsteady problems, *Comput. Fluids* 61 (2012) 86–93, <https://doi.org/10.1016/j.compfluid.2012.03.006>.
- [63] A. Arnone, Integration of Navier–Stokes equations using dual time stepping and a multigrid method, *AIAA J.* 33 (6) (1995) 985–990, <https://doi.org/10.2514/3.12518>.
- [64] P. Birken, G. Gassner, M. Haas, C.-d. Munz, in: *European Congress on Computational Methods and Applied Sciences and Engineering, ECCOMAS 2012*, 2012.
- [65] D.J. Mavriplis, An assessment of linear versus nonlinear multigrid methods for unstructured mesh solvers, *J. Comput. Phys.* 175 (1) (2002) 302–325, <https://doi.org/10.1006/jcph.2001.6948>.
- [66] J.H. Williamson, Low-storage Runge–Kutta schemes, *J. Comput. Phys.* 35 (1) (1980) 48–56.
- [67] L. Haupt, J. Stiller, W.E. Nagel, A fast spectral element solver combining static condensation and multigrid techniques, *J. Comput. Phys.* 255 (2013) 384–395, <https://doi.org/10.1016/j.jcp.2013.07.035>.
- [68] S. Sherwin, G. Karniadakis, A triangular spectral element method; applications to the incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 123 (1–4) (1995) 189–229, [https://doi.org/10.1016/0045-7825\(94\)00745-9](https://doi.org/10.1016/0045-7825(94)00745-9).
- [69] J. Romero, A. Jameson, Extension of the flux reconstruction method to triangular elements using collapsed-edge quadrilaterals, in: 54th AIAA Aerospace Sciences Meeting, January, 2016, pp. 1–17.
- [70] W. Li, J. Pan, Y.-x. Ren, The discontinuous Galerkin spectral element methods for compressible flows on two-dimensional mixed grids, *J. Comput. Phys.* 364 (2018) 314–346, <https://doi.org/10.1016/j.jcp.2018.03.001>.
- [71] T. Liszka, W. Tworzydło, J. Bass, S. Sharma, T. Westermann, B. Yavari, ProPHLEX – an hp-adaptive finite element kernel for solving coupled systems of partial differential equations in computational mechanics, *Comput. Methods Appl. Mech. Eng.* 150 (1–4) (1997) 251–271, [https://doi.org/10.1016/S0045-7825\(97\)00088-1](https://doi.org/10.1016/S0045-7825(97)00088-1).
- [72] N.K. Burgess, An adaptive discontinuous Galerkin solver for aerodynamic flows, January 2011.
- [73] T.B. Lynn, D.W. Bechert, D.A. Gerich, Direct drag measurements in a turbulent flat-plate boundary layer with turbulence manipulators, *Exp. Fluids* 19 (6) (1995) 405–416, <https://doi.org/10.1007/BF00190258>.

- [74] F. Hindenlang, T. Bolemann, C.-D. Munz, Mesh curving techniques for high order discontinuous Galerkin simulations, in: *IDIHOM: Industrialization of High-Order Methods-A Top-Down Approach*, Springer, 2015, pp. 133–152.
- [75] R. Campregher, J. Militzer, R. Campregher, S.S. Mansur, A. da Silveira Neto, Computations of the flow past a still sphere at moderate reynolds numbers using an immersed boundary method, *J. Braz. Soc. Mech. Sci. Eng.* 31 (4) (2009) 344–352, <https://doi.org/10.1590/S1678-58782009000400009>.
- [76] B. Fornberg, Steady viscous flow past a sphere at high Reynolds numbers, *J. Fluid Mech.* 190 (1988) 471–489, <https://doi.org/10.1017/S0022112088001417>.
- [77] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, *J. Comput. Phys.* 161 (1) (2000) 35–60, <https://doi.org/10.1006/jcph.2000.6484>, arXiv:1501.0228.
- [78] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, *J. Comput. Phys.* 190 (2) (2003) 572–600, [https://doi.org/10.1016/S0021-9991\(03\)00298-5](https://doi.org/10.1016/S0021-9991(03)00298-5).
- [79] R.M. Kirby, G.E. Karniadakis, Selecting the numerical flux in discontinuous Galerkin methods for diffusion problems 22 and 23, *J. Sci. Comput.* 22 (2005) 385–411, <https://doi.org/10.1007/s10915-004-4145-5>.