Hybridizable Discontinuous Galerkin with degree adaptivity for the incompressible Navier-Stokes equations[☆]

Giorgio Giorgiani[a], Sonia Fernández-Méndez[a], Antonio Huerta[a,*]

[a] *Laboratori de Calcul Numeric (LaCaN). Departament de Matematica Aplicada III, E.T.S. de Ingenieros de Caminos, Canales y Puertos, Universitat Politècnica de Catalunya – BarcelonaTech, Jordi Girona 1, 08034 Barcelona, Spain.*

Abstract

A degree adaptive Hybridizable Discontinuous Galerkin (HDG) method for the solution of the incompressible Navier-Stokes equations is presented. The key ingredient is an accurate and computationally inexpensive a posteriori error estimator based on the super-convergence properties of HDG. The error estimator drives the local modification of approximation degree in the elements and faces of the mesh, aimed at obtaining a uniform error distribution below a user-given tolerance in a given are of interest. Three 2D numerical examples are presented. High efficiency of the proposed error estimator is found, and an important reduction of the computational effort is shown with respect to non-adaptive computations, both for steady state and transient simulations.

Keywords: Hybrid methods, Discontinuous Galerkin, Navier-Stokes

[☆]This work has been partially supported by the Spanish Ministry of Science and Competitiveness, grants number: CICYT-DPI2011-27778-C02-02 and DPI2011-23058, the European Comission, grant number: PITN-2009-238548, and the Catalan Government, grant number: 2009SGR875.

[*]Corresponding author

Email addresses: giorgio.giorgiani@upc.edu (Giorgio Giorgiani), sonia.fernandez@upc.edu (Sonia Fernández-Méndez), antonio.huerta@upc.edu (Antonio Huerta)

URL: www.lacan.upc.edu (Antonio Huerta)

equations, CFD, p-adaptivity, high-order, Hybridizable Discontinuous Galerkin

1. Introduction

The importance of adaptive simulations in the field of computational fluid dynamics (CFD) has been pointed out by various authors in the last years [1–9]. Non-uniform discretizations, adapted to local flow features, are necessary to capture strong variations in the solution, shocks, sharp fronts or boundary layers, while keeping a coarser mesh where it is possible. Moreover, in transient problems, the discretization must be regularly updated, to adapt to the flow features evolving in time. In this context, the efficiency of the error estimator, or indicator, and the automatic adaptive algorithm is crucial.

On one hand, despite the existing vast literature on adaptive algorithms, see for instance [1, 3–5, 7], degree adaptive techniques have not received the attention they deserve, probably due to its cumbersome implementation in the context of continuous approximations. Nevertheless, degree adaptive algorithms, usually in combination with mesh size adaptation, have proved to be clearly suited for CFD computations, specially in the context of discontinuous approximations [3, 10]. This work focuses exclusively on degree adaptivity, given that mesh size adaptivity has been extensively studied. A proper element size in the whole domain is assumed, such that the solution has exponential convergence for increasing degree.

On the other hand, the ability and efficiency of high-order approximations has been discussed and proven in recent papers [11–13]. In particular, the interest in Discontinuous Galerkin (DG) methods has increased over the past years, because they have proved their suitability to construct robust stabilized high-order numerical schemes on arbitrary unstructured and non-

conforming grids for a variety of physical phenomena, see, for instance, [14–20] for the solution of incompressible flows. More precisely, the Hybridizable Discontinuous Galerkin (HDG) method [20–29] stands out among all DG methods for implicit schemes, thanks to its stability features, its reduced number of degrees of freedom, and its superconvergence properties.

After the initial success of degree adaptive HDG in wave problems, see [27], the approach is further extended and tested for incompressible flows. The element-by-element discontinuous approximation, common to all DG methods, enables a straightforward implementation of variable degree computations. In addition, an inexpensive element-by-element post-process provides an HDG superconvergent solution that can be used to define an asymptotically exact error estimator for the velocity field. The error estimate drives an automatic update of the approximation degree in each element, which is aimed at obtaining a uniform error distribution with a user defined tolerance.

Despite the simplicity and low cost of the error estimator, high efficiency is exhibited for analytical examples. Moreover, although adaptivity relies on an error estimate for just the velocity field, high accuracy is attained for all variables, with sharp resolution of key features of the flow and accurate evaluation of fluid-dynamic forces. In particular, high degrees are automatically located along boundary layers, reducing the need for highly distorted elements in the computational mesh. Numerical tests show an important reduction in computational cost, compared to uniform degree computations, for both steady and unsteady computations.

The paper is organized as follows. Section 2 recalls the basics on the HDG method for the solution of the incompressible Navier-Stokes equations with variable degree discretizations. Section 3 is devoted to the error estimation and the degree adaptive algorithm. Numerical examples in section

4 demonstrate the high efficiency of the error estimator, and the excellent performance of the adaptive algorithm in terms of computational effort for 2D steady state and transient simulations.

2. Variable degree HDG for incompressible Navier-Stokes

2.1. Navier-Stokes over a broken domain

Let $\Omega \subset \mathbb{R}^d$ be an open bounded domain with boundary $\partial\Omega$ split in the Dirichlet, $\partial\Omega_D$, and Neumann, $\partial\Omega_N$, boundaries, and let T be the final instant of interest. Recall the unsteady Navier-Stokes equations

$$\begin{cases} \partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\cdot(-p\mathbf{I} + \nu\boldsymbol{\nabla}\boldsymbol{u}) = \boldsymbol{f} & \text{in } \Omega\times]0,T[, \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} = 0 & \text{in } \Omega\times]0,T[, \\ \boldsymbol{u} = \boldsymbol{g} & \text{on } \partial\Omega_D\times]0,T[, \\ (-p\mathbf{I} + \nu\boldsymbol{\nabla}\boldsymbol{u})\cdot\boldsymbol{n} = \boldsymbol{t} & \text{on } \partial\Omega_N\times]0,T[, \\ \boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0 & \text{in } \Omega. \end{cases} \tag{1}$$

Where $\boldsymbol{u}$ and p are the velocity and the kinematic pressure in the fluid, ν is the kinematic viscosity, $\boldsymbol{f}$ is a body force, $\boldsymbol{n}$ is the unitary outward normal vector, $\mathbf{I}$ is the identity matrix, $\boldsymbol{g}$ is the prescribed velocity on the Dirichlet boundary $\partial\Omega_D$, $\boldsymbol{t}$ are the prescribed pseudo-tractions imposed on the Neumann boundary $\partial\Omega_N$, and $\boldsymbol{u}_0$ is the initial velocity field (assumed solenoidal: $\boldsymbol{\nabla}\cdot\boldsymbol{u}_0 = 0$).

Note that, these Neumann boundary conditions do not correspond to regular stresses but to pseudo-stresses, see [30] for a discussion on pseudo-stress boundary conditions or, more specifically, [20] for a survey on the implementation of different kind of boundary conditions for the Navier-Stokes equations in HDG. This is standard in velocity–pressure formulation.

For discontinuous Galerkin approaches the domain Ω is partitioned in $\mathtt{n_{el}}$ disjoint elements Ω_i with boundaries $\partial\Omega_i$, such that

$$\overline{\Omega} = \bigcup_{i=1}^{\mathtt{n_{el}}} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \text{ for } i \neq j,$$

and the union of all $\mathtt{n_{fc}}$ faces (sides for 2D) is denoted as

$$\Gamma := \bigcup_{i=1}^{\mathtt{n_{el}}} \partial\Omega_i.$$

The discontinuous setting induces a new problem equivalent to (1). It is written as a system of first order partial differential equations (mixed form) with some element-by-element equations and some global ones, namely, for $i = 1, \ldots, \mathtt{n_{el}}$

$$\left.\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} &= \boldsymbol{0} \\ \partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\cdot(-p\mathbf{I} + \nu\boldsymbol{L}) &= \boldsymbol{f} \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \end{aligned}\right\} \qquad \text{in } \Omega_i \times]0,T[, \tag{2a}$$

$$\boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0 \qquad \text{in } \Omega_i, \tag{2b}$$

and

$$[\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma\backslash\partial\Omega \times]0,T[, \tag{2c}$$

$$[\![(-p\mathbf{I} + \nu\boldsymbol{L})\cdot\boldsymbol{n}]\!] = \boldsymbol{0} \qquad \text{on } \Gamma\backslash\partial\Omega \times]0,T[, \tag{2d}$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega_D \times]0,T[, \tag{2e}$$

$$(-p\mathbf{I} + \nu\boldsymbol{L})\cdot\boldsymbol{n} = \boldsymbol{t} \qquad \text{on } \partial\Omega_N \times]0,T[. \tag{2f}$$

Where a new variable $\boldsymbol{L}$ for the velocity gradient tensor is introduced after splitting the second order momentum conservation equation in two first order equations. The *jump* $[\![\cdot]\!]$ operator is defined at each internal face of Γ, i.e. on $\Gamma\backslash\partial\Omega$, using values from the elements to the left and right of the face (say, Ω_i and Ω_j), namely

$$[\![\circledcirc]\!] = \circledcirc_i + \circledcirc_j,$$

and always involving the normal vector $\boldsymbol{n}$, see [31] for details. Thus, equation (2c) imposes the continuity of velocity and equation (2d) imposes the continuity of the normal component of the pseudo-stress across interior faces.

2.2. The HDG local problem

A major feature of HDG is that, in general, unknowns are restricted to the skeleton of the mesh, that is, the union of all faces denoted by Γ. Here, the velocity field, $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$, on the mesh skeleton Γ is this unknown. However, as described next, satisfying the incompressibility equation requires the introduction of one scalar unknown per element, irrespective of the polynomial order used in the approximation. There are many choices for this scalar unknown; here, the mean pressure on the element boundary is chosen. The introduction of the new variable $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$ on the mesh skeleton Γ is crucial to define two types of problems: a local problem for each element and a global one for all faces.

In fact, the local element-by-element problem corresponds to the Navier-Stokes equations on each element, see equations (2a), with imposed Dirichlet boundary conditions. The imposed Dirichlet conditions on the element boundary are precisely the velocities $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$ for $\boldsymbol{x} \in \Gamma$.

It is well known that an incompressible Navier-Stokes problem on a bounded domain with non-homogeneous velocity prescribed everywhere on the boundary requires a solvability condition and implies that pressure is know up to a constant. More precisely, the continuity equation implies a zero flux compatibility condition. Thus, the newly introduced variable $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$ on the mesh skeleton Γ must verify

$$\left\langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, 1 \right\rangle_{\partial\Omega_i} = 0, \quad \text{for } i = 1, \ldots, \mathtt{n_{el}}. \tag{3}$$

Now the local element-by-element Navier-Stokes equations can be solved

to determine $(\boldsymbol{u}, \boldsymbol{L}, p)$ in terms of the imposed $\hat{\boldsymbol{u}}(\boldsymbol{x}, t)$ on the mesh skeleton Γ. Thus, for $i = 1, \ldots, \mathtt{n_{el}}$ the local HDG problem is solved, namely

$$\left.\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla}\boldsymbol{u} &= \boldsymbol{0} \\ \partial_t \boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\cdot(-p\mathbf{I} + \nu\boldsymbol{L}) &= \boldsymbol{f} \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \end{aligned}\right\} \qquad \text{in } \Omega_i \times]0, T[, \tag{4a}$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \qquad \text{in } \Omega_i, \tag{4b}$$

$$\boldsymbol{u} = \hat{\boldsymbol{u}} \qquad \text{on } \partial\Omega_i \times]0, T[, \tag{4c}$$

$$\big\langle p, 1 \big\rangle_{\partial\Omega_i} = \rho_i. \tag{4d}$$

As noted earlier, (4) is a Dirichlet problem and, consequently, pressure is determined up to a constant. This constant is determined by prescribing some value for the pressure. Typical choices are to impose pressure at one point or to prescribe the mean value in the domain. In HDG, as noted earlier and since the unknowns are restricted to the mesh skeleton, the usual choice is to prescribe the mean pressure on the element boundary, see (4d), where $\big\langle \cdot, \cdot \big\rangle_B$ denotes the $\mathcal{L}^2$ scalar product of the traces over any $B \subset \Gamma$.

At this point it is important to notice that given the values of the velocities on Γ, $\hat{\boldsymbol{u}}(\cdot, t) \in [\mathcal{L}^2(\Gamma)]^d$ for any instant $t \in [0, T]$, the same Dirichlet boundary condition is imposed to the left and right element of a given face. Consequently, the velocity continuity, recall equation (2c), is ensured by (4c). Obviously, on the Dirichlet boundary $\hat{\boldsymbol{u}} = \boldsymbol{g}$, replicating (2e).

Observe that in each element the original unknowns, $(\boldsymbol{L}, \boldsymbol{u}, p)$, can be determined in terms of the two extra unknowns: the velocity on Γ, $\hat{\boldsymbol{u}}$, and the vector of average pressures on the boundaries for each element, $\{\rho\}_{i=1}^{\mathtt{n_{el}}} \in \mathbb{R}^{\mathtt{n_{el}}}$.

An approximation is obtained after the corresponding discretization, see [20]. Two types of finite dimensional spaces must be defined one for functions

in the elements interior and another for trace functions, namely

$$
\begin{aligned}
\mathcal{V}_t^h &:= \left\{ v : v(\cdot, t) \in \mathcal{V}^h \text{ for any } t \in [0, T] \right\}, \text{ with} \\
\mathcal{V}^h &:= \left\{ v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{k_{\Omega_i}}(\Omega_i) \text{ for } i = 1, \ldots, \mathtt{n_{el}} \right\}, \text{ and} \\
\Lambda_t^h &:= \left\{ \hat{v} : \hat{v}(\cdot, t) \in \Lambda^h \text{ for any } t \in [0, T] \right\}, \text{ with} \\
\Lambda^h &:= \left\{ \hat{v} \in \mathcal{L}^2(\Gamma) : \hat{v}|_{\Gamma_i} \in \mathcal{P}^{k_{\Gamma_i}}(\Gamma_i) \text{ for } i = 1, \ldots, \mathtt{n_{fc}} \right\},
\end{aligned}
$$

where $\mathcal{P}^k$ denotes the space of polynomials of degree less or equal to k, while k_{Ω_i} and k_{Γ_i} are the polynomial degrees in element Ω_i and face Γ_i respectively, see Remark 1 for a discussion on the degree on the skeleton for non-uniform approximation and Remark 2 for details on the polynomial interpolation. To simplify the presentation, in an abuse of notation, the same notation is used for the numerical approximation, belonging to the finite dimensional spaces, and the exact solution, that is $(u, \boldsymbol{L}, p)$.

The weak problem for each element corresponding to (4) becomes: given $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}} \in \mathbb{R}^{\mathtt{n_{el}}}$ and $\hat{\boldsymbol{u}} \in [\Lambda_t^h]^d$ satisfying (3), find an approximation $(\boldsymbol{L}, \boldsymbol{u}, p) \in [\mathcal{V}_t^h]^{d\times d} \times [\mathcal{V}_t^h]^d \times \mathcal{V}_t^h$ such that

$$
\begin{aligned}
\left(\boldsymbol{G}, \boldsymbol{L}\right)_{\Omega_i} + \left(\boldsymbol{\nabla}\cdot\boldsymbol{G}, \boldsymbol{u}\right)_{\Omega_i} - \left\langle \boldsymbol{G}\cdot\boldsymbol{n}, \hat{\boldsymbol{u}} \right\rangle_{\partial\Omega_i} &= 0, \\
\left(\boldsymbol{v}, \partial_t \boldsymbol{u}\right)_{\Omega_i} + \left(\boldsymbol{v}, (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u}\right)_{\Omega_i} + \left(\boldsymbol{\nabla}\boldsymbol{v}, -p\mathbf{I} + \nu\boldsymbol{L}\right)_{\Omega_i} & \\
-\left\langle \boldsymbol{v}, (-p\mathbf{I} + \nu\boldsymbol{L})\cdot\boldsymbol{n} \right\rangle_{\partial\Omega_i} - \left\langle \boldsymbol{v}, \tau(\hat{\boldsymbol{u}} - \boldsymbol{u}) \right\rangle_{\partial\Omega_i} &= \left(\boldsymbol{v}, \boldsymbol{f}\right)_{\Omega_i}, \\
-\left(\boldsymbol{\nabla} q, \boldsymbol{u}\right)_{\Omega_i} + \left\langle q, \hat{\boldsymbol{u}}\cdot\boldsymbol{n} \right\rangle_{\partial\Omega_i} &= 0, \\
\left\langle p, 1 \right\rangle_{\partial\Omega_i} &= \rho_i,
\end{aligned}
\tag{5}
$$

for all $(\boldsymbol{G}, \boldsymbol{v}, q) \in [\mathcal{V}^h]^{d\times d} \times [\mathcal{V}^h]^d \times \mathcal{V}^h$, for $i = 1, \ldots, \mathtt{n_{el}}$, and with the initial condition defined in (4b). As usual, $\left(\cdot, \cdot\right)_{\Omega_i}$ denotes the $\mathcal{L}^2$ scalar product in the element Ω_i.

In this weak problem it is important to note two details. First, the Dirichlet boundary conditions, (4c), are imposed weakly; and second, the

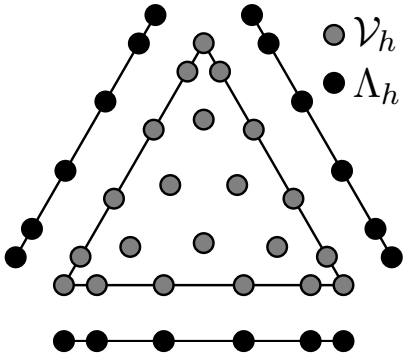

Figure 1: Nodes representing the spaces $\mathcal{V}^h$ and Λ^h.

trace of the normal stress has been replaced in all boundary integrals by the following numerical trace

$$\widehat{(-p\mathbf{I}+\nu\boldsymbol{L})}\cdot\boldsymbol{n} := (-p\mathbf{I}+\nu\boldsymbol{L})\cdot\boldsymbol{n}+\tau(\hat{\boldsymbol{u}}-\boldsymbol{u}), \tag{6}$$

where τ is a stability parameter, see Remark 3 for a more detailed discussion.

Remark 1 (Non-uniform degree). *Polynomial degree can vary from element to element and, consequently, from face to face. The interpolation degree k_{Γ_f} for a face shared by two elements, $\Gamma_f = \partial\Omega_i \cap \partial\Omega_j$, is set as the maximum of the degree in Ω_i and Ω_j, that is $k_{\Gamma_f} = \max\{k_{\Omega_i}, k_{\Omega_j}\}$, and $k_{\Gamma_f} = k_{\Omega_i}$ when $\Gamma_f \subset \partial\Omega_i \cap \partial\Omega$. This procedure ensures that, for any element Ω_i, the degree on all its faces is at least k_{Ω_i}, providing the desired accuracy in the element, see also [28, 29].*

Remark 2 (Polynomial interpolation). *Standard nodal basis are considered here, and Fekete node distributions are used to minimize ill-conditioning, see [32]. As an illustrative example, Figure 1 shows the nodes corresponding to the spaces $\mathcal{V}^h$ and Λ^h for a triangle with polynomial degree $k = 5$. However, other approximations can be considered.*

2.3. The HDG global problem

The local problems (4), or (5), allow to compute the solution in the whole domain in terms of the trace of the velocity on the mesh skeleton, $\hat{\boldsymbol{u}}$, and the mean pressure for each element, $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}}$. Thus, these two variables can now be understood as the actual unknowns of the problem. They are determined using the global equations in (2), namely from (2c) to (2f), and the solvability condition (3). In fact, as already discussed, equations (2c) and (2e) are already imposed. Thus (2d) and (2f) are the remaining global conditions which must be imposed. These two equations (in weak form) and the solvability condition (3) determine the HDG *global problem*. Namely , find approximations $\hat{\boldsymbol{u}} \in [\Lambda_t^h(\boldsymbol{g})]^d$ and $\{\rho_i\}_{i=1}^{\mathtt{n_{el}}} \in \mathbb{R}^{\mathtt{n_{el}}}$ such that

$$\sum_{i=1}^{\mathtt{n_{el}}} \left\langle \hat{\boldsymbol{v}}, (-p\mathbf{I} + \nu \boldsymbol{L}) \cdot \boldsymbol{n} + \tau(\hat{\boldsymbol{u}} - \boldsymbol{u}) \right\rangle_{\partial\Omega_i} = \left\langle \hat{\boldsymbol{v}}, \boldsymbol{t} \right\rangle_{\partial\Omega_N}, \tag{7a}$$

$$\left\langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, 1 \right\rangle_{\partial\Omega_i} = 0, \text{ for } i = 1, \ldots, \mathtt{n_{el}}, \tag{7b}$$

for all $\hat{\boldsymbol{v}} \in [\Lambda^h(\mathbf{0})]^d$. Here $(\boldsymbol{L}, \boldsymbol{u}, p) \in [\mathcal{V}_t^h]^{d\times d} \times [\mathcal{V}_t^h]^d \times \mathcal{V}_t^h$ are solution of the local problems (5) and the trace spaces associated to the Dirichlet boundary are defined by $[\Lambda^h(\square)]^d = \{\hat{\boldsymbol{v}} \in [\Lambda^h]^d : \hat{\boldsymbol{v}} = \mathbb{P}_\partial \square \text{ on } \partial\Omega_D\}$, with $\mathbb{P}_\partial$ the $\mathcal{L}^2$ projection on $\partial\Omega_D$.

Note that equation (2d) imposes continuity of the of the normal component of the pseudo-traction on each element face, which induces (7a) after using (6). Thus equation (7a) weakly imposes the continuity of the normal pseudo-stress.

Remark 3 (Stabilization tensor). *The stabilization parameter was introduced in* (6)*. Its influence is crucial and has been discussed and analyzed for a large number of problems by Cockburn and co-workers see, for instance, [20–25]. Choosing the correct stabilization parameter should provide sufficient*

stabilization to the solution and, at the same time, should ensure superconvergence of the post-processed solution, see also the discussion in [12].

Some useful criteria are provided in [20, 23, 24]. While in the diffusion limit (no convection) the single face *τ guarantees both stability and superconvergence with simple numerical considerations (see [12]), in presence of convection this choice is not available. In fact, the* single face *approach is not capable to stabilize the convective effect, see [23] for a detailed discussion.*

The dimensional analysis presented in [20] induces to use a constant positive stabilization parameter $\tau \approx \nu/\ell + \|\boldsymbol{u}\|$, where ℓ is a representative length scale and $\|\boldsymbol{u}\|$ a characteristic velocity. The first term accounts for the diffusive effect and the second accounts for the convective effect. As proved in [20], this choice ensure stability and superconvergence of the post-processed solution. In the following numerical tests, since $\nu/\ell \ll |\boldsymbol{u}|$, the parameter τ is set to $\tau \approx |\boldsymbol{u}|$.

Finally it is important to remark that a more general stabilization can be implemented by introducing a stabilization tensor $\mathbf{S}$, which will modify equation (6) as follows

$$\widehat{(-p\mathbf{I} + \nu\boldsymbol{L})} \cdot \boldsymbol{n} := (-p\mathbf{I} + \nu\boldsymbol{L}) \cdot \boldsymbol{n} + \mathbf{S} \cdot (\hat{\boldsymbol{u}} - \boldsymbol{u}).$$

Nevertheless, this richer stabilization does not seem necessary in the examples discussed here, as well as in the examples presented in the litterature.

Remark 4 (Mixed interpolation requirements). *Note that, in contrast to other mixed or hybrid methods, HDG allows for equal interpolation in the space of the velocity, $[\mathcal{V}^h]^d$, and the space of its gradient, $[\mathcal{V}^h]^{d\times d}$, as well as for the trace variable, $[\Lambda^h]^d$. This is due to the particular form of the numerical fluxes and the stabilization parameter. They ensure solvability and stability, see [22], without the need of an enriched space for the gradient*

variable, or a reduced space for the trace variable.

Remark 5 (Nonlinear DAE system). *The HDG discrete problem is defined by (5) and (7). It is a system of Differential Algebraic Equations (DAE) of index 1, that can be efficiently discretized in time with an implicit time integrator, such as backward Euler, a Backward Differentiation Formula, or a diagonally implicit Runge-Kutta method, see [20]. Time discretization of (5) and (7) leads to a non-linear system of equations that can be solved with an iterative scheme. Here, the non-linear system has been linearized using the Newton-Raphson method. That is, every non-linear convective term in (5), which can be expressed as a trilinear form $c(\boldsymbol{u};\boldsymbol{v},\boldsymbol{u})$, is linearized using the first-order approximation $c(\boldsymbol{u}^r;\boldsymbol{v},\boldsymbol{u}^r) \approx c(\boldsymbol{u}^{r-1};\boldsymbol{v},\boldsymbol{u}^r) + c(\boldsymbol{u}^r;\boldsymbol{v},\boldsymbol{u}^{r-1}) - c(\boldsymbol{u}^{r-1};\boldsymbol{v},\boldsymbol{u}^{r-1})$, where here r is the iteration count. Obviously, this implies that the global solution, equation (7), must be iterated.*

In any case, a linear system of equations is solved for each iteration of the non-linear solver. In this linear system, the equations corresponding to (5) can be solved element-by-element to express the solution at each element Ω_i in terms of the trace variable, $\hat{\boldsymbol{u}}$, and the mean of the pressure in the element boundary, ρ_i. Then, these expressions are replaced in (7) yielding a global system of equations that only involves $\hat{\boldsymbol{u}}$ and $\{\rho\}_{i=1}^{n_{el}}$, with an important reduction in number of DOF. Further details on the efficient solution of the non-linear DAE can be found in [20].

Steady computations are also considered here and they follow the same solution procedure. That is, the non-linear system corresponding to (5) and (7), after neglecting the time derivatives, is linearized with the Newton-Raphson method. Then, the resulting linear system to be solved in each iteration is reduced to a system involving only $\hat{\boldsymbol{u}}$ and $\{\rho\}_{i=1}^{n_{el}}$.

Remark 6 (Imposing boundary tractions). *In some problems, viz. fluid*

structure interaction, the actual tractions on the Neumann boundary must be imposed. This requires to compute Cauchy stresses and not pseudo-traction. The equations (2) are rewritten for each element $i = 1, \ldots, \mathtt{n_{el}}$

$$\left.\begin{aligned} \boldsymbol{L} - \boldsymbol{\nabla} \boldsymbol{u} &= \boldsymbol{0} \\ \partial_t \boldsymbol{u} + (\boldsymbol{u} \cdot \boldsymbol{\nabla}) \boldsymbol{u} - \boldsymbol{\nabla} \cdot (-p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2) &= \boldsymbol{f} \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} &= 0 \end{aligned}\right\} \quad \text{in } \Omega_i \times]0, T[,$$

$$\boldsymbol{u}(\boldsymbol{x}, 0) = \boldsymbol{u}_0 \qquad \text{in } \Omega_i,$$

and

$$\left.\begin{aligned} [\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] &= \boldsymbol{0} \\ [\![(-p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2) \cdot \boldsymbol{n}]\!] &= \boldsymbol{0} \end{aligned}\right\} \quad \text{on } \Gamma \backslash \partial\Omega \times]0, T[,$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \partial\Omega_D \times]0, T[,$$

$$(-p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2) \cdot \boldsymbol{n} = \boldsymbol{t} \qquad \text{on } \partial\Omega_N \times]0, T[.$$

The corresponding HDG local problem becomes:

$$\begin{aligned} \big(\boldsymbol{G}, \boldsymbol{L}\big)_{\Omega_i} + \big(\boldsymbol{\nabla} \cdot \boldsymbol{G}, \boldsymbol{u}\big)_{\Omega_i} - \big\langle \boldsymbol{G} \cdot \boldsymbol{n}, \hat{\boldsymbol{u}} \big\rangle_{\partial\Omega_i} &= 0, \\ \big(\boldsymbol{v}, \partial_t \boldsymbol{u}\big)_{\Omega_i} + \big(\boldsymbol{v}, (\boldsymbol{u} \cdot \boldsymbol{\nabla}) \boldsymbol{u}\big)_{\Omega_i} + \big(\boldsymbol{\nabla} \boldsymbol{v}, -p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2\big)_{\Omega_i} & \\ - \big\langle \boldsymbol{v}, (-p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2) \cdot \boldsymbol{n} \big\rangle_{\partial\Omega_i} - \big\langle \boldsymbol{v}, \tau (\hat{\boldsymbol{u}} - \boldsymbol{u}) \big\rangle_{\partial\Omega_i} &= \big(\boldsymbol{v}, \boldsymbol{f}\big)_{\Omega_i}, \\ - \big(\boldsymbol{\nabla} q, \boldsymbol{u}\big)_{\Omega_i} + \big\langle q, \hat{\boldsymbol{u}} \cdot \boldsymbol{n} \big\rangle_{\partial\Omega_i} &= 0, \\ \big\langle p, 1 \big\rangle_{\partial\Omega_i} &= \rho_i, \end{aligned}$$

for $i = 1, \ldots, \mathtt{n_{el}}$*, and the global one is*

$$\sum_{i=1}^{\mathtt{n_{el}}} \big\langle \hat{\boldsymbol{v}}, (-p \mathbf{I} + \nu (\boldsymbol{L} + \boldsymbol{L}^T)/2) \cdot \boldsymbol{n} + \tau (\hat{\boldsymbol{u}} - \boldsymbol{u}) \big\rangle_{\partial\Omega_i} = \big\langle \hat{\boldsymbol{v}}, \boldsymbol{t} \big\rangle_{\partial\Omega_N},$$

$$\big\langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, 1 \big\rangle_{\partial\Omega_i} = 0, \ \textit{for } i = 1, \ldots, \mathtt{n_{el}},$$

where now $\boldsymbol{t}$ *are actual imposed boundary tractions.*

Remark 7 (HDG post-processed solution). *The solution of the HDG problem, given by equations (5) and (7), provides a numerical solution* $(\boldsymbol{L}, \boldsymbol{u}) \in [\mathcal{V}_t^h]^{d\times d} \times [\mathcal{V}_t^h]^d$ *with optimal numerical convergence in both variables. Then, a new problem can be solved element-by-element to compute a superconvergent approximation of the velocity* $\boldsymbol{u}^*$*, namely, for* $i = 1, \ldots, \mathtt{n_{el}}$*, solve*

$$\begin{cases} \boldsymbol{\nabla}\cdot\boldsymbol{\nabla}\boldsymbol{u}^* = \boldsymbol{\nabla}\cdot\boldsymbol{L} & \text{in } \Omega_i, \\ \boldsymbol{n}\cdot\boldsymbol{\nabla}\boldsymbol{u}^* = \boldsymbol{n}\cdot\boldsymbol{L} & \text{on } \partial\Omega_i, \\ \left(\boldsymbol{u}^*, 1\right)_{\Omega_i} = \left(\boldsymbol{u}, 1\right)_{\Omega_i}. \end{cases}$$

This induces a weak problem in a richer finite dimensional space, that is, find $\boldsymbol{u}^* \in [\mathcal{V}_t^{h*}]^d$ *such that*

$$\left(\boldsymbol{\nabla}\boldsymbol{v}, \boldsymbol{\nabla}\boldsymbol{u}^*\right)_{\Omega_i} = \left(\boldsymbol{\nabla}\boldsymbol{v}, \boldsymbol{L}\right)_{\Omega_i} \text{ and } \left(\boldsymbol{u}^*, 1\right)_{\Omega_i} = \left(\boldsymbol{u}, 1\right)_{\Omega_i},$$

for all $\boldsymbol{v} \in [\mathcal{V}^{h*}]^d$ *and* $i = 1, \ldots, \mathtt{n_{el}}$*, where* $\mathcal{V}^{h*}$ *must be a bigger space than* $\mathcal{V}^h$*. In fact, with one degree more in the element-by-element polynomial approximation, i.e.* $\mathcal{V}^{h*} = \left\{v \in \mathcal{L}^2(\Omega) : v|_{\Omega_i} \in \mathcal{P}^{k_{\Omega_i}+1}(\Omega_i), \text{ for } i = 1, \ldots, \mathtt{n_{el}}\right\}$*,* $\boldsymbol{u}^*$ *converges asymptotically at a rate* $k+2$ *in the* $\mathcal{L}^2$ *norm, see [25, 26] for uniform degree* k*. In case of non-uniform degree, superconvergence of the HDG post-process solution is proved in [28, 29] under the conditions in Remark 1. Note that the post-process solution is not required to be computed at each time step, but only when an improved solution is needed. Moreover, the computational overhead is small and decreases with the approximation degree, see [12].*

In the following section, the superconvergent solution $\boldsymbol{u}^*$ is used to compute a reliable and inexpensive error estimator for the HDG velocity approximation $\boldsymbol{u}$.

3. Error estimation and degree adaptive algorithm

The use of *a posteriori* computable error estimates to drive an automatic adaptive process is nowadays a common practice to control the accuracy of the computation and to ensure the quality of the FE solution [33]. Two different approaches can be used for assessing the error: error indicators or error estimators, see [34] for a general discussion. Error indicators are cheap to evaluate but, in general, are designed for a specific problem and do not provide error bounds. Error estimators are more accurate and general, can be used in linear or nonlinear problems, see [35], and can even produce bounds of the exact solution [36–39]. However, a posteriori error estimators have a non trivial computational overhead when recovery techniques are used [40], or when equilibrated fluxes must be computed [41], and also in the more efficient flux-free implementations [42]. Here an inexpensive, reliable and computable error estimator for the velocity field is proposed. It is based on the superconvergent HDG post-processed solution.

More precisely, the $\mathcal{L}_2$ error in the velocity field $\boldsymbol{u}$ is estimated in an element Ω_i as

$$E_i^2 = \frac{1}{|\Omega_i|} \int_{\Omega_i} (\boldsymbol{u}^* - \boldsymbol{u})^2 \, d\Omega, \tag{8}$$

where $\boldsymbol{u}$ is the solution of the HDG problem, see equations (5) and (7), and $\boldsymbol{u}^*$ is the improved velocity, see Remark 7. In contrast with [10], here the elemental error is normalized by the element measure [43]. This normalization is crucial for non-uniform meshes. Note that the evaluation of the improved velocity $\boldsymbol{u}^*$ requires an inexpensive element-by-element computation, see [12] and Table 1 in examples Section 4.2. Moreover, the post-process, which implies evaluating $\boldsymbol{u}^*$, is only done when the error is evaluated, not necessarily at each time step. Given the superconvergence properties of the improved solution $\boldsymbol{u}^*$, the error estimate E_i is an asymptotically exact estimate.

Given the error estimate (8), an automatic degree adaptive process is proposed. A tolerance ε for the velocity elemental error in a region of interest $\Omega_{\mathtt{int}} \subset \Omega$ is assumed. The adaptive process aims to get a map of elemental degrees $\{k_{\Omega_i}\}_{i=1}^{\mathtt{n_{el}}}$ such that the error estimate of the HDG solution satisfies

$$E_i \leq \varepsilon \quad \forall\, \Omega_i \subset \Omega_{\mathtt{int}}. \tag{9}$$

In each iteration of the adaptive process, given distribution of the error, a map of degree increments Δk_{Ω_i} is evaluated for each element in the whole computational domain Ω. Inspired from [3], the degree variation in each element is computed as

$$\Delta k_{\Omega_i} = \left\lceil \log_b(E_i/\epsilon_i) \right\rceil, \text{ for } i = 1, \ldots, \mathtt{n_{el}}. \tag{10}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. The elemental tolerance ϵ_i usually corresponds to the tolerance ε in (9), but can also be set differently element-by-element, for instance, to intensify the refinement in the region of interest Ω_{int}, see [27].

Similarly to the *adaptation aggressiveness* parameter proposed in [6], the logarithm base b controls the behavior of the adaptive scheme. Namely, for a fixed elemental error ratio E_i/ϵ_i, increasing b has the obvious effect of decreasing Δk_{Ω_i}. Thus, small values of b yield drastic variations in the polynomial degree for the elements. This may reduce the number of iterations until convergence but it can also produce an oscillatory degree map $\{k_{\Omega_i}\}_{i=1}^{\mathtt{n_{el}}}$ around the optimal one. On the contrary, large values of b yield slow degree variations. The number of iterations to determine the optimal degree map is consequently increased but it converges to the optimal map without undesired oscillations. The optimal value of the parameter b depends on the tolerance ε required in the computation and on the specific problem at hand. Here,

reasonable results are obtained with $10 \leq b \leq 100$. These bounds correspond to an increment $\Delta k_{\Omega_i} = 1$ for $E_i/\epsilon_i = 10$ and $E_i/\epsilon_i = 100$ respectively.

In addition, upper and lower bounds for the approximation degree are usually defined, namely $k^- \leq k_{\Omega_i} \leq k^+$. Here, the lower bound k^- is used to guarantee a correct representation of the geometry with isoparametric elements. On the contrary, the upper bound k^+, which limits the maximum polynomial degree in the mesh, is set for practical reasons. It is worth noting that h-refinement would be necessary if this upper bound for the degree is reached and prevents convergence to a solution with the desired accuracy. However, as shown in the numerical tests, in most of the cases the *a priori* information on the solution allows to distribute the element sizes in such a way that h-adaptivity is not needed.

The adaptive procedure is applied to both steady and unsteady problems. In both cases, the adaption process starts with a uniform degree mesh such that $k_{\Omega_i} = k^-$, for $i = 1, \ldots, \mathtt{n_{el}}$.

If a steady-state solution is sought, each iteration of the adaptive process consists on computing the steady-state solution of (5) and (7) neglecting the time derivatives, evaluating the error estimate (8), updating the degree map using (10) within the bounds $k^- \leq k_{\Omega_i} \leq k^+$, and projecting the solution onto the new computational mesh. The iterative adaptation scheme stops when the prescribed precision ε is attained in the domain of interest, i.e. (9). Failure to converge to the desired accuracy is assumed when the percentage of elements changing their degree in two successive iterations of the adaptive process is lower than a given value, for instance 1%, and the target criterion (9) is not accomplished.

To reach convergence of the non-linear problem in the first steady computation, a relaxation technique may be necessary. Here, in the numerical

examples, for the computation of the solution in the initial mesh, the transient problem is advanced in time with large time step until the steady solution is reached. However, for subsequent computations, for each iteration of the adaptive process, a good approximation of the solution is available from the previous computation, and can be used as initial guess for the steady non-linear problem. Thus, in most cases, the non-linear steady problem can be directly solved, reducing the computational cost of each iteration of the adaptive process to the solution of one non-linear system of equations.

For the computation of unsteady solutions, error estimation and consequent degree adaptation is repeatedly performed after a fixed number of time steps N. That is, every N time steps, the post-process to obtain the superconvergent velocity $\boldsymbol{u}^*$ is done, the error estimate (8) is evaluated, the degree map is updated using (10) within the bounds $k^- \leq k_{\Omega_i} \leq k^+$, the solution is projected onto the new computational mesh, and the time integration is continued. The flow charts of the adaptive strategy, for steady and unsteady solutions, are shown in Figure 2.

Numerical experiments in next section show that, even though the adaptive process is based on an error estimate for the velocity field, accurate approximations of the fluid-dynamic forces are also obtained.

4. Numerical tests

The performance of the error estimation and adaptive technique is tested in three numerical examples. First, a Wang flow with known analytical solution is considered, hence estimated errors and exact errors can be compared. No analytical solution is available for the last two examples, thus a reference solution is considered to evaluate the error in each case. Reference solutions are overkilled approximations computed with high-order HDG on an

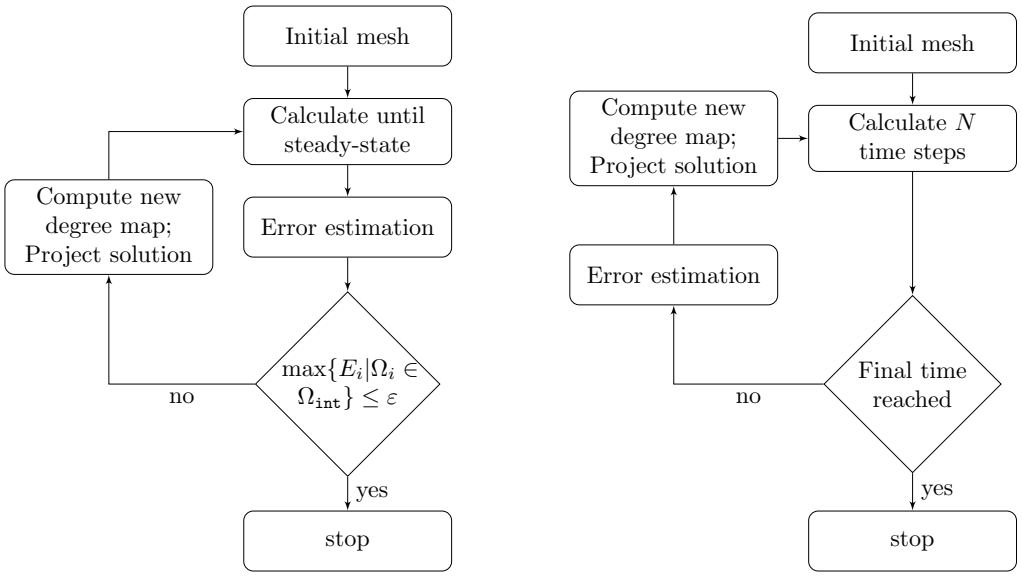

Figure 2: Adaptive strategy for steady-state solutions (left) and for transient solutions (right).

h-refined mesh.

4.1. Wang flow

This synthetic solution, proposed [44], is solved in $\Omega =]-0.5, 0.5[\times]0, 1[$ with a source term $f = 0$ and imposing an analytical solution for the velocity as

$$\boldsymbol{u} = \begin{pmatrix} 2ay - b\lambda \exp(-\lambda y)\cos(\lambda x) \\ b\exp(-\lambda y)\sin(\lambda x)\lambda \end{pmatrix}.$$

The coefficients are set to $a = 1$, $b = 1$ and $\lambda = 10$, which leads to large variations of the velocity and pressure profile at the bottom of the computational domain, similarly to a boundary layer behavior. Figure 3 shows the streamlines (left), the pressure profile at $x = 0$ (right) and the velocity profile at the bottom boundary (bottom).

No Neumann boundary conditions are set in this problem, thus the global problem for pressure is closed setting to zero its average value in the domain,

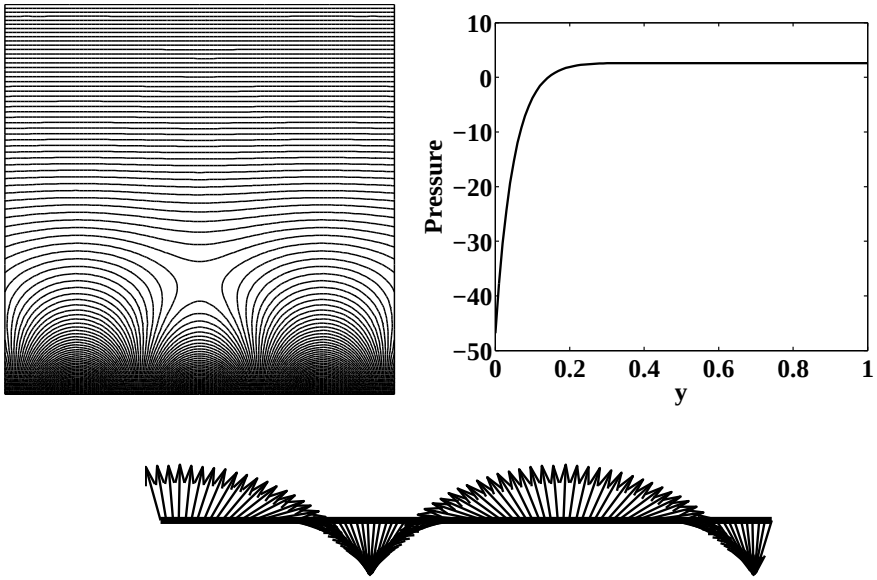

Figure 3: Wang flow problem: streamlines (left), pressure profile at $x = 0$ (right) and velocity profile in the bottom boundary (bottom).

i.e. $\big(p, 1\big)_\Omega = 0$ is added to system (7).

To evaluate the accuracy of the error estimator, the estimated and exact errors are compared for a structured triangular mesh with element size $h = 0.1$ and uniform degree k, for $k = 1 \ldots 8$. Figure 4 left shows the comparison between the estimated and exact maximum $\mathcal{L}^2$ elemental error, exhibiting an excellent agreement. The right figure depicts the estimator efficiency, defined as

$$\eta = \max_{i=1,\ldots,\mathtt{n_{el}}} \{E_i | \Omega_i \subset \Omega\} / \max_{i=1,\ldots,\mathtt{n_{el}}} \{\tilde{E}_i | \Omega_i \subset \Omega\} - 1,$$

where $\tilde{E}_i$ is the elemental error computed with the analytical solution. The maximum elemental error is always estimated with an accuracy greater than 95%, demonstrating the excellent performance of the proposed estimator.

Figure 5 shows the evolution of the maximum elemental $\mathcal{L}^2$ error for an adaptive computation with tolerance $\varepsilon = 10^{-8}$, uniform elemental tolerance

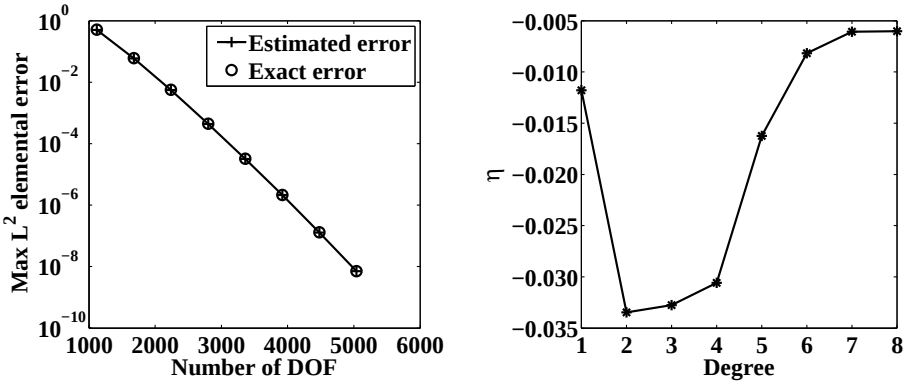

Figure 4: Wang flow problem: uniform degree computations, $k = 1 \ldots 8$. Comparison of the estimated and exact maximum $\mathcal{L}^2$ elemental error (left) and error estimator efficiency (right).

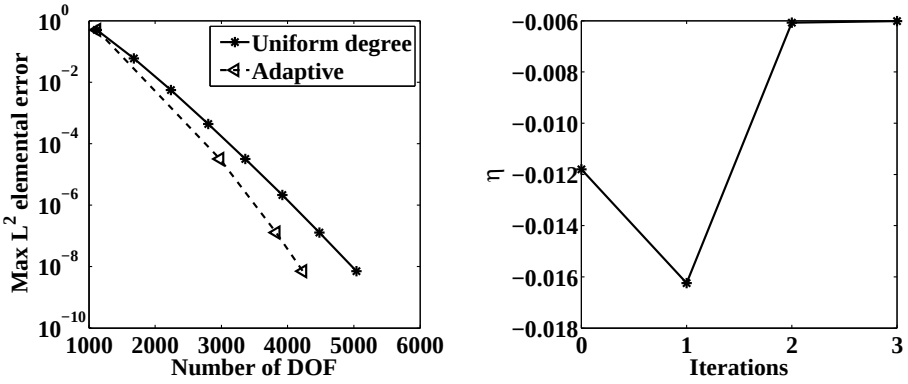

Figure 5: Wang flow problem: comparison of uniform degree computations, for $k = 1 \ldots 8$, and a degree adaptive computation (left), and error estimator efficiency in the three iterations of the adaptive computation (right)

$\epsilon_i = \varepsilon$ for $i = 1, \ldots, \mathtt{n_{el}}$, and adaption aggressiveness parameter $b = 100$. Starting from a mesh with uniform degree $k = 1$, the desired accuracy is reached with three iterations of the adaptive process.

In the left figure, the convergence to the exact solution obtained with the degree adaptive computation is compared with the convergence of uniform degree computations, for degree $k = 1 \ldots 8$. The maximum elemental error is plotted versus the number of coupled DOF, that is, the size of the linear system of the HDG discretization. Even in this simple example, the adaptive technique reduces the computational cost respect to the uniform degree computation, involving a smaller number of DOF for a given accuracy. The right figure shows the efficiency of the estimator in each iteration of the adaptive process, which is always greater than 98%.

Figure 6 shows the map of elemental degrees, $\{k_{\Omega_i}\}_{i=1}^{\mathtt{n_{el}}}$, for each iteration. As expected, higher approximation degrees are concentrated to the bottom of the domain, to properly capture the sharp gradient in the solution. The estimated and exact elemental error maps are also shown for each iteration, revealing again an excellent performance of the estimator.

It is worth mentioning that, in all computations, if the simulation is continued after the convergence criteria is satisfied, the maximum error either stays under the tolerance, or has small oscillations around the tolerance value in a small fraction of the elements in the mesh. This confirms the robustness of the convergence criteria.

4.2. Evaluation of the NACA 0012 aerodynamic characteristics

The second numerical test concerns the evaluation of the aerodynamic characteristics of a NACA 0012 airfoil in laminar flow regime and steady state. The analytical expression considered for the parametrization of the

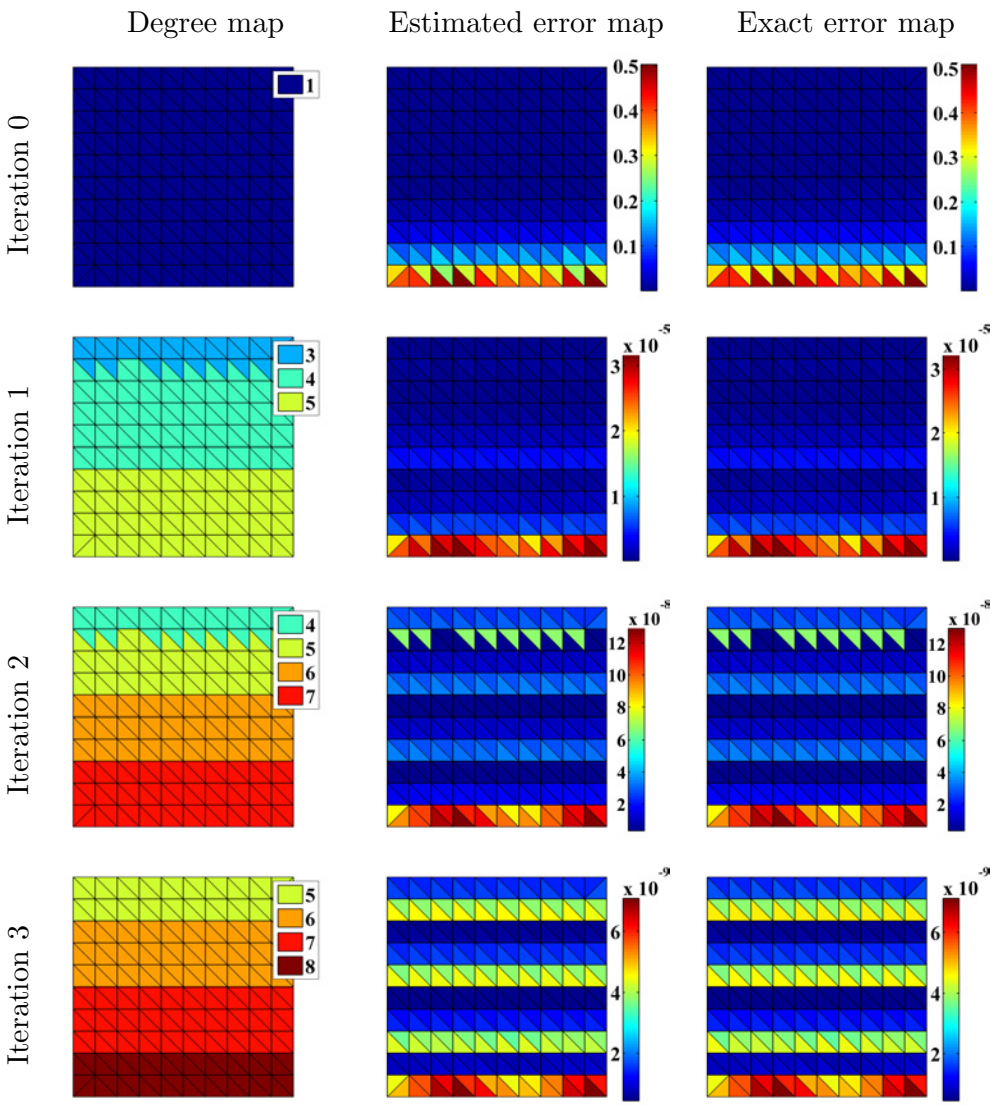

Figure 6: Wang flow problem: degree map, estimated error map and exact error map in each iteration of the adaptive simulation.

upper part of the symmetric NACA airfoil is

$$y = \frac{\ell}{0.2}(0.2969\sqrt{x} - 0.1260x - 0.3516x^2 + 0.2843x^3 - 0.1036x^4), \qquad (11)$$

for $x \in [0, 1]$, where $\ell = 0.12$ is the maximum thickness of the airfoil. The formula provides a zero thickness airfoil at the trailing edge, which is usually preferred for numerical simulations.

Similarly to [45], the geometry is approximated with a conformal mapping. Here the Kármán-Trefftz (K-T) transformation is used, which allows to obtain a profile with a corner in the trailing edge. The K-T transformation in the complex plane is

$$\zeta \to z = n\frac{(1 + 1/\zeta)^n + (1 - 1/\zeta)^n}{(1 + 1/\zeta)^n - (1 - 1/\zeta)^n},$$

and maps the circumference of center (ξ_0, η_0), passing through the origin of the complex plane, to an approximation of the airfoil profile. The coefficient $\eta_0 = 0$ yields a symmetric profile, while the coefficients $n = 1.9506$ and $\xi_0 = 0.0749$ are chosen to minimize the error between the real NACA profile given by (11) and the K-T profile. The mapping is used to obtain the computational mesh shown in Figure 7, where each element layer corresponds to a circumference of increasing radius. As in [46], the mesh is refined around the airfoil to capture the boundary layer, and further refined at the leading and trailing edges. This avoids too large elements in the area of the domain where the strongest gradients of the solution appear, that could lead to a non-convergent degree adaptive process requiring *hp*-refinement. In this example a geometrical representation with constant degree $k = 2$ is used in all computations. However, it is worth mentioning that in practical applications a proper description of the geometry is crucial to attain the best accuracy for a given computational mesh [47, 48].

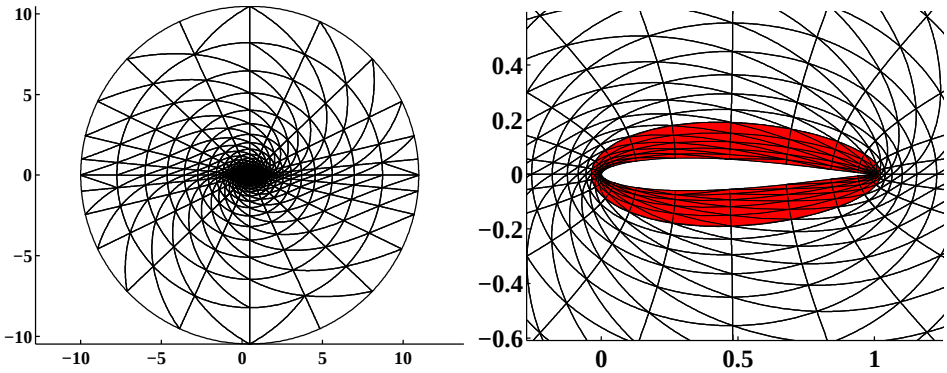

Figure 7: NACA 0012 airfoil: computational mesh (right) and zoom around the airfoil (left). The area of interest is also highlighted.

In each iteration of the adaptive process the degree is updated for all elements in the computational mesh. A uniform error tolerance $\epsilon_i = \varepsilon$ for $i = 1, \ldots, \mathtt{n_{el}}$ is set, aiming at obtaining a uniform error distribution in the whole domain, and minimizing possible pollution effects. Nevertheless, the stopping criterion for the adaptive procedure is based on checking the error only at the elements in the area of interest $\Omega_{\mathtt{int}}$, see Equation (9). In this section, the region of interest $\Omega_{\mathtt{int}}$ is defined as the area composed by 5 element layers surrounding the airfoil, see Figure 7. As will be shown in the numerical tests, this convergence criteria for the adaptive procedure provides an accurate approximation of the aerodynamic forces on the airfoil. Obviously, a goal oriented analysis would be needed, hence involving the solution of an adjoint problem, to relate the elemental errors to the accuracy in the aerodynamic characteristics, see for example [8, 9]. This analysis is out of the scope of the present work.

Results on the mesh depicted in Figure 7 are discussed next. Reference values of the aerodynamic coefficients at steady state are obtained using a mesh with element size $h/2$ and uniform degree $k = 6$. Increasing the degree or reducing the mesh size does not improve the results of these reference

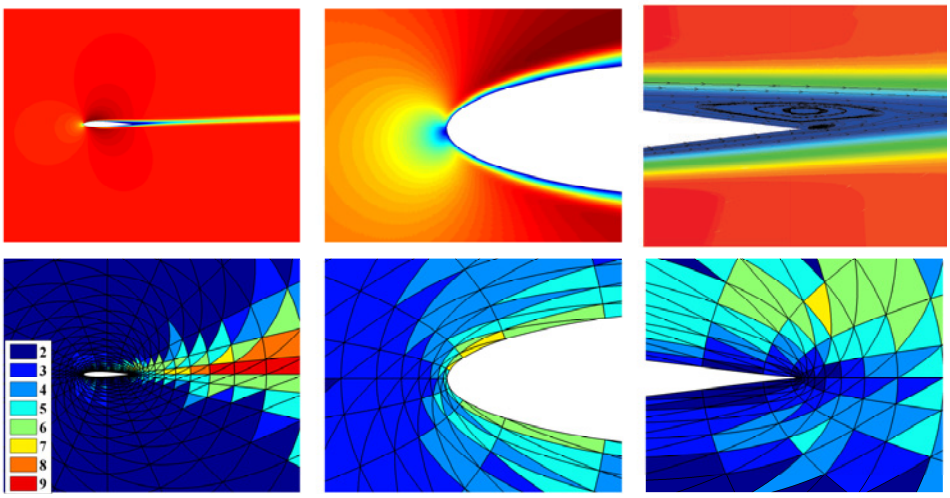

Figure 8: NACA 0012 airfoil: solution of the adaptive computation at Re=5 000 and $= 2$, after convergence in 5 iterations with tolerance 10^{-4}. Velocity module (top) and degree map (bottom) around the airfoil and details around the leading edge (center) and the trailing edge (right).

mesh for the outputs desired.

4.2.1. Re=5 000

First, a simulation at Re=5 000 with angle of attack $= 2$ is performed. The initial adaptive iteration is with a uniform degree $k = 2$, the tolerance is prescribed at $= 10^{-4}$, adaptation parameter is $b = 10$ and a maximum degree $k^{+} = 10$ is de ned. Convergence of the adaptive process is achieved in 5 iterations. Figure 8 shows the solution and the map of elemental degrees in di erent parts of the domain. As expected, higher order elements are automatically placed in the boundary layer and in the wake. However, in the majority of the computational mesh low-order elements are used, achieving a great reduction in computational cost compared to a uniform high-order mesh. Thus, the boundary layer at the leading edge and the recirculation bubbles at the trailing edge are correctly captured. The re nement also

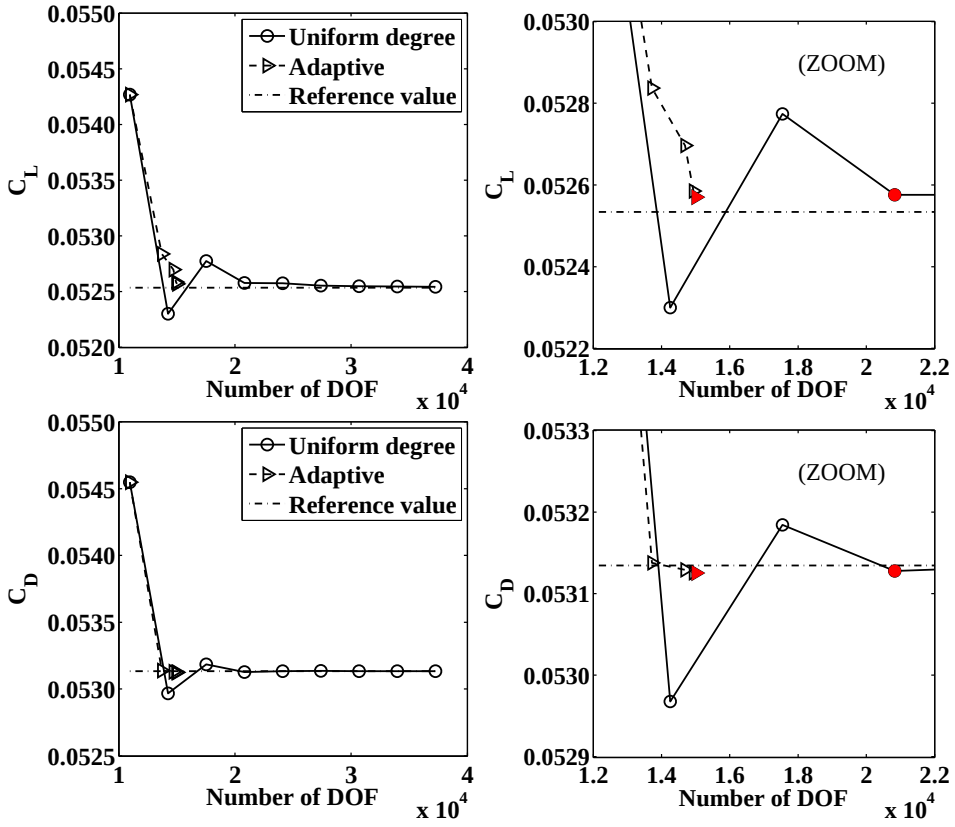

Figure 9: NACA 0012 airfoil: convergence for adaptive and uniform degree simulations of the aerodynamic coefficients C_L (top) and C_D (bottom) at Re=5 000 and $\alpha = 2°$. The zoomed plots on the left shown in red the converged solution for a prescribed tolerance.

propagates in front of the leading edge of the airfoil, where the flow starts to notice the presence of the object. Note that, thanks to the high-order elements that are automatically placed after the leading edge along the airfoil, there is no need for highly stretched elements to resolve the boundary layer.

Figure 9 shows the convergence to the reference values of the lift, C_L, and drag, C_D, coefficients both for the uniform degree computations, degrees from $k = 2$ to $k = 10$, and the adaptive scheme. Fast convergence to the reference solution is obtained with the adaptive simulation, reaching an error smaller than 0.07% in the C_L and than 0.02% in the C_D. Similar accuracy is

Table 1: NACA 0012 airfoil at Re=5 000 and $\alpha = 2°$: runtime at each iteration for solving and estimating the error.

iteration	0	1	2	3	4
runtime solve [s]	91	21	26	22	23
runtime err. est.[s]	2	4	4	4	4

obtained with a uniform degree $k = 5$, but with about 40% more DOF than the adaptive computation.

To evaluate the overhead of computing the error with the proposed estimator, Table 1 s hows runtimes for solving the HDG steady problem and for estimating the error at each iteration. All computations are carried out with Matlab running on a Xeon E5640, 2.66 Ghz/12MB cachewith 72 GB of RAM. Is it worth noting that, for convergence of the non-linear solver with the initial mesh, the transient problem is advanced in time with a large time step until the steady-state solution in reached. However, as commented in Section 3, for subsequent computations, in each iteration of the adaptive process, the steady problem corresponding to equations (7) and (5) neglecting time derivatives, is directly solved. The initial guess for the non-linear solver is set as the solution of the previous computation, and it is good enough to provide convergence of the non-linear solver, without the need of a relaxation strategy. The complexity of the computation in each iteration of the adaptive process is then reduced to solving a single non-linear problem, with clearly smaller runtime than the initial computation.

In any case, in all iterations of the adaptive process the runtime of the error estimation is significantly smaller than the cost of solving the HDG non-linear problem.

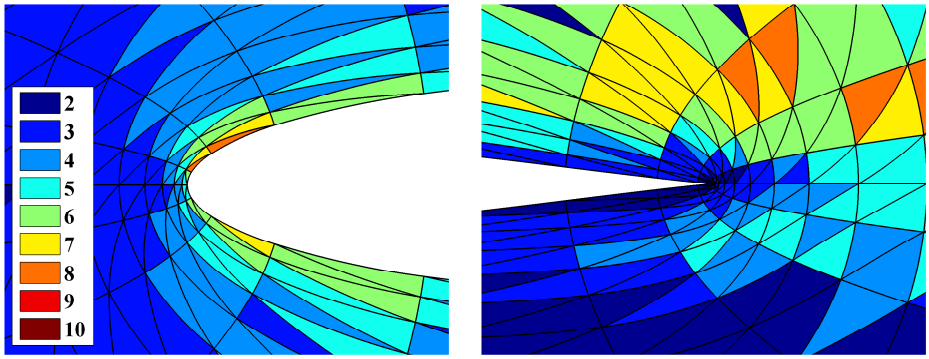

Figure 10: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: map of converged degrees at the leading edge (left) and at the trailing edge (right).

4.2.2. Re=10 000

The second steady simulation uses the same computational mesh with Re=10 000 and 2° angle of attack. The same parameters and tolerance are used to set up the adaptive process. Convergence is achieved with 6 iterations. A detail of the final degree map is shown in Figure 10. Note that, comparing to the results with Re=5 000 depicted in Figure 8, higher order elements are automatically placed at the leading edge and at the trailing edge to capture the thinner boundary layer. For instance, the presence of elements of degree $k = 8$ in the leading edge, that are not present in the simulation with Re=5 000. Figure 11 shows a comparison on how the recirculation bubbles are captured by the starting uniform $k = 2$ mesh and by the converged adaptive simulation. This underlines how the adaptive technique correctly captures features of the flow that are not properly resolved with the original discretization.

This conclusion can be extended to other flow features such as, pressure and velocity gradient. In fact, Figure 12 shows the pressure and the skin friction coefficients around the airfoil, in very good agreement with the reference solution. For this reason, it is possible to notice that, even though the error

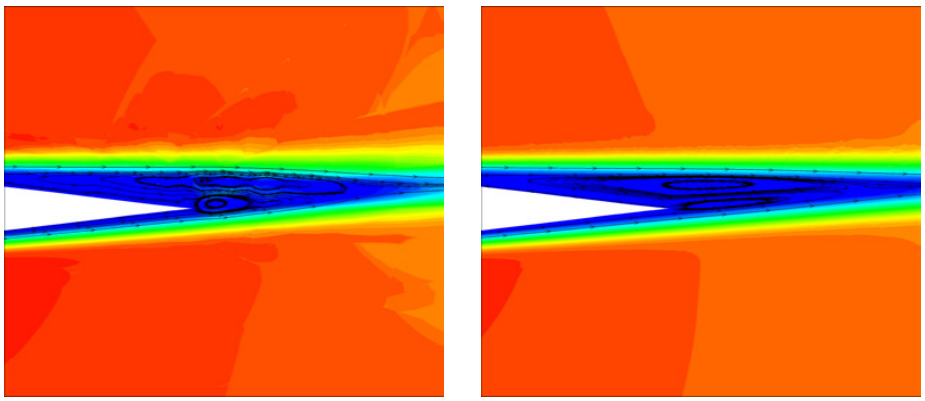

Figure 11: NACA 0012 airfoil: recirculation bubbles in the simulation at Re=10 000 as captured with a uniform $k = 2$ computation (left) and the adaptive computation (right).

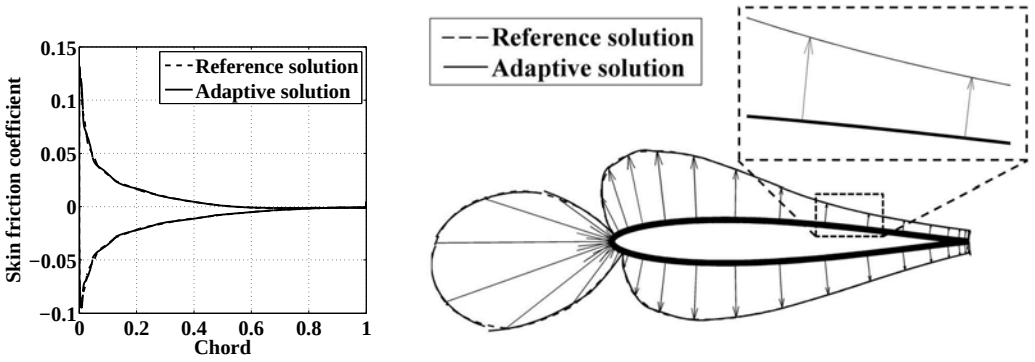

Figure 12: NACA 0012 airfoil at Re=10 000 and $\alpha = 2°$: skin friction coefficient (left) and pressure coefficient (right).

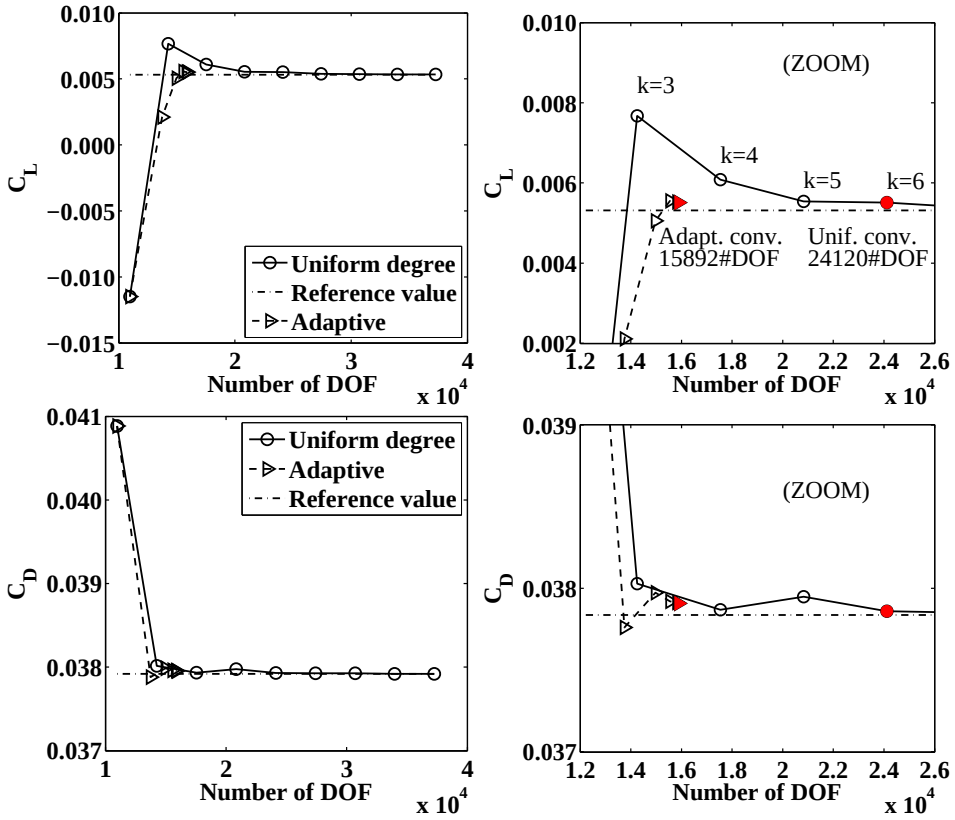

Figure 13: NACA 0012 airfoil: convergence for adaptive and uniform degree simulations of the aerodynamic coefficients C_L (top) and C_D (bottom) at Re=10 000 and $\alpha = 2°$. The zoomed plots on the left shown in red the converged solution for a prescribed tolerance.

estimator is only based on the velocity error, the fluid dynamic force on the airfoil, that involves integrals of the pressure and the velocity gradient on the airfoil boundary, is properly captured. This is also confirmed depicting the convergence of aerodynamic coefficients, see Figure 13. In this case, the uniform degree computation with similar accuracy has 50% more DOF than the adaptive simulation.

It is important to notice that, even if the stopping criterion is limited to the area of interest of the domain depicted in Figure 7, the refinement process is performed in the whole computational mesh. This reduces the

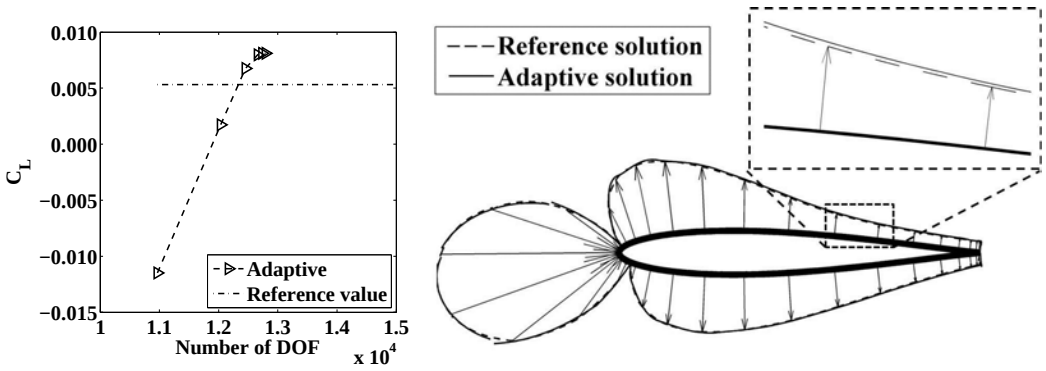

Figure 14: Non converging solution: C_L approximations and final pressure coefficient around the NACA 0012 airfoil at Re=10 000 with a too large error tolerance in the wake.

pollution error from the outer part of the domain on the area of interest, that could affect the final solution and also preclude the convergence of the iterative process. To underline the importance of the refinement in the wake, the simulation at Re=10 000 is also performed reducing the elemental tolerance ϵ_i outside the region of interest, that is, $\epsilon_i = 10^{-4}$ for $\Omega_i \subset \Omega_{\texttt{int}}$ and $\epsilon_i = 10^{-2}$ for $\Omega_i \subset \Omega \setminus \Omega_{\texttt{int}}$. No convergence is achieved in this case and the computation is stopped in the 5-th iteration, when in two successive iterations the degree remains unchanged for 99% of the elements. The resulting solution, see Figure 14, presents a noticeable error when compared to the reference solution. Even though globally the pressure coefficient seems to be well captured, the error in the pressure coefficient (which is shown in the zoom) produces an overestimate of the C_L of 50%.

4.3. Von Kármán street adaptive simulation

The objective of the last example is to study the performance of the proposed degree adaptive technique in a transient simulation. The periodic vortex shedding flow past a unitary radius cylinder is studied at Re=100. The computational mesh, shown in Figure 15, is has 728 unstructured and

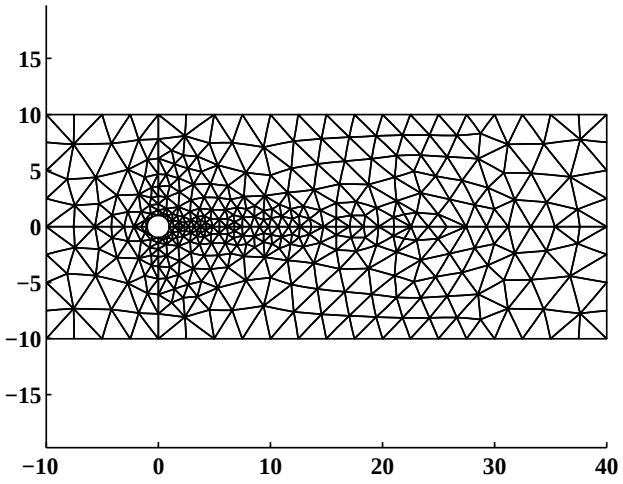

Figure 15: Von Kármán street example: computational mesh.

non-uniform triangular elements. It is refined around the cylinder, namely 12 curved elements with a geometrical degree $k = 2$ surround the cylinder. Since only space discretization errors are of concern here, no time step adaptation is implemented, and an overkilled time-step is imposed. The simulation is started incrementing Re from 0.5, reaching the prescribed value in 200 time steps. The adaptive scheme is set up with tolerance $\varepsilon = 10^{-3}$, uniform elemental tolerance $\epsilon_i = \varepsilon$ for $i = 1, \ldots, \mathtt{n_{el}}$, adaptation parameter $b = 10$ and minimum degree $k^- = 2$. No upper bound for the degree is prescribed in this case. The error estimation and degree adaptation is performed every 10 time steps.

Results of the adaptive simulation are compared with four uniform degree simulations, with degree $k = 3, 4, 5, 6$, considering the uniform $k = 6$ solution as the reference solution. Figure 16 depicts the variation in time of the number of DOF of the adaptive computation, compared with the constant DOF of the uniform degree computations. For the adaptive simulation, three different phases can be identified. First the number of DOF increases smoothly and reaches a constant value. In this part of the simulation, the

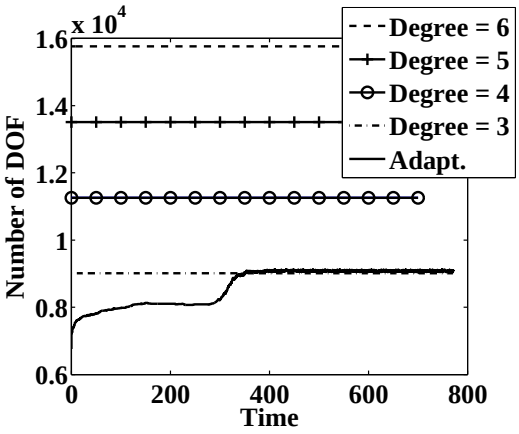

Figure 16: Von Kármán street past a cylinder at Re=100: time evolution of number of DOF for the adaptive and uniform degree ($k = 3, 4, 5, 6$) simulations.

cylinder is not shedding and the adaptive technique is placing high-order elements around the cylinder and in the recirculation zone behind it. The rapid increase in the number of DOF around time $t = 300$ corresponds to the start of the shedding pattern. This can also be observed in Figure 17, which shows the solution and the corresponding degree map $\{k_{\Omega_i}\}_{i=1}^{\mathtt{n_{el}}}$ at two time steps: just before and after the start of the vortex shedding. The final phase corresponds to a constant number of DOF, indicating that the periodic state is reached. In other words, the complexity of the solution is reflected in the number of DOF that the adaptive technique automatically places in the domain, underlining once more the efficiency of this methodology for transient problems.

Table 2 shows the error in the mean drag force on the cylinder, averaged in one period of the periodic shedding regime, for the adaptive computation and for uniform degree computations with $k = 3, 4, 5$. Since the adaptive computation never requires elemental degrees larger than 6 in any element, the mean drag force for the uniform degree computation with $k = 6$ is considered as reference value. The adaptive simulation provides the best accuracy,

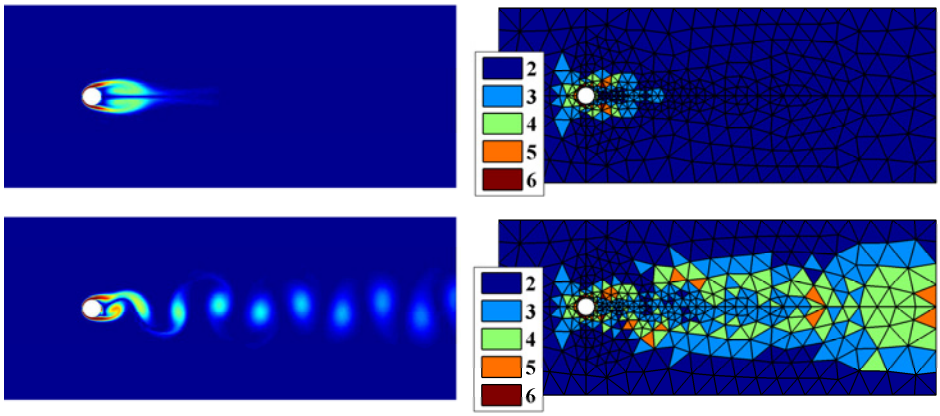

Figure 17: Von Kármán street past a cylinder at Re=100: vorticity pattern (left) and degree map (right) at a time step, $t = 14$, before (top) and, $t = 750$, after (bottom) the initiation of the shedding .

Table 2: Mean drag error for one period and % of DOF with respect to the reference solution ($k = 6$).

	$k = 3$	$k = 4$	$k = 5$	Adaptive
Mean drag error %	1.8	0.52	0.13	0.014
DOF %	57	71	86	≤ 58

with the minimum number of DOF. With a number of DOF close to the uniform degree $k = 3$ computation, the adaptive degree computation provides an accuracy with two more significant digits than uniform degree $k = 3$, and one more significant digit than the uniform degree $k = 5$ computation. In fact, the adaptive computation provides an accuracy similar to a uniform degree $k = 6$ computation, with less than 58% DOF in the complete time span.

5. Conclusions

A degree adaptive HDG method for the solution of the incompressible Navier-Stokes equations is proposed. The adaptive technique is based on the local modification of the polynomial degree of the approximation on the mesh elements and faces, and it is driven by an a-posteriori error estimator for the velocity field, based on the superconvergence properties of HDG. The proposed error estimator involves only element-by-element computations, thus it is computationally inexpensive, while providing accurate estimation of the velocity elemental error.

Three 2D numerical tests are shown. High efficiency of the error estimator is illustrated with a problem with known analytical solution, for both uniform and variable degree meshes. The methodology is then applied to two examples of engineering interest: the estimation of the aerodynamic characteristics of a NACA 0012 airfoil in laminar flow, and the evaluation of the drag force on a shedding cylinder. High accuracy and reduction in computational cost, compared with non-adaptive simulations, is found both for steady state and transient simulations. Numerical results also demonstrate that, even though the adaptive process is based on an error estimate only for the velocity field, accurate approximations of the fluid-dynamic forces are

obtained.

References

[1] J. Wu, J. Z. Zhu, J. Szmelter, O. C. Zienkiewicz, Error estimation and adaptivity in Navier-Stokes incompressible flows, Comput. Mech. 6 (4) (1990) 259–270.

[2] S. Prudhomme, J. Oden, A posteriori error estimation and error control for finite element approximations of the time-dependent Navier-Stokes equations, Finite Elem. Anal. Des. 33 (4) (1999) 247–262.

[3] J.-F. Remacle, J. E. Flaherty, M. S. Shephard, An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems, SIAM Rev. 45 (1) (2003) 53–72.

[4] A. Burbeau, P. Sagaut, A dynamic p-adaptive discontinuous Galerkin method for viscous flow with shocks, Comput. Fluids 34 (4-5) (2005) 401–417.

[5] E. Oñate, J. Arteaga, J. Garcia, R. Flores, Error estimation and mesh adaptivity in incompressible viscous flows using a residual power approach, Comput. Methods Appl. Mech. Eng. 195 (4-6) (2006) 339–362.

[6] K. J. Fidkowski, D. L. Darmofal, A triangular cut-cell adaptive method for high-order discretizations of the compressible Navier-Stokes equations, J. Comput. Phys. 225 (2) (2007) 1653–1672.

[7] K.-J. Bathe, H. Zhang, A mesh adaptivity procedure for CFD and fluid-structure interactions, Comput. Struct. 87 (11-12) (2009) 604–617.

[8] R. Hartmann, J. Held, T. Leicht, F. Prill, Discontinuous Galerkin methods for computational aerodynamics – 3D adaptive flow simulation with the DLR PADGE code, Aerosp. Sci. Technol. 14 (7) (2010) 512–519.

[9] R. Hartmann, J. Held, T. Leicht, Adjoint-based error estimation and adaptive mesh refinement for the RANS and $k-\omega$ turbulence model equations, J. Comput. Phys. 230 (11) (2011) 4268–4284.

[10] C. Eskilsson, An hp-adaptive discontinuous Galerkin method for shallow water flows, Int. J. Numer. Methods Fluids 67 (11) (2011) 1605–1623.

[11] A. Huerta, X. Roca, A. Aleksandar, J. Peraire, Are High-order and Hybridizable Discontinuous Galerkin methods competitive?, Oberwolfach Rep. 9 (1) (2012) 485–487, abstracts from the workshop held February 12–18, 2012, Organized by Olivier Allix, Carsten Carstensen, Jörg Schröder and Peter Wriggers, Oberwolfach Reports. Vol. 9, no. 1. doi:10.4171/OWR/2012/09.

[12] G. Giorgiani, D. Modesto, S. Fernández-Méndez, A. Huerta, High-order continuous and discontinuous Galerkin methods for wave problems, Int. J. Numer. Methods Fluids 73 (10) (2013) 883–903.

[13] A. Huerta, A. Angeloski, X. Roca, J. Peraire, Efficiency of high-order elements for continuous and discontinuous Galerkin methods, Int. J. Numer. Methods Eng. 96 (9) (2013) 529–560.

[14] B. Cockburn, G. Kanschat, D. Schötzau, The local discontinuous Galerkin method for linearized incompressible fluid flow: a review, Comput. Fluids 34 (4–5) (2005) 491–506.

[15] K. Shahbazi, P. F. Fischer, C. R. Ethier, A high-order discontinuous

Galerkin method for the unsteady incompressible Navier-Stokes equations, J. Comput. Phys. 222 (1) (2007) 391–407.

[16] F. Bassi, A. Crivellini, D. A. Di Pietro, S. Rebay, An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows, Comput. Fluids 36 (10) (2007) 1529–1546.

[17] B. Cockburn, G. Kanschat, D. Schötzau, An equal-order DG method for the incompressible Navier-Stokes equations, J. Sci. Comput. 40 (1-3) (2009) 188–210.

[18] A. Montlaur, S. Fernández-Méndez, J. Peraire, A. Huerta, Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations, Int. J. Numer. Methods Fluids 64 (5) (2010) 549–564.

[19] F. Bassi, N. Franchina, A. Ghidoni, S. Rebay, Spectral p-multigrid discontinuous Galerkin solution of the Navier-Stokes equations, Int. J. Numer. Methods Fluids 67 (11) (2011) 1540–1558.

[20] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations, J. Comput. Phys. 230 (4) (2011) 1147–1170.

[21] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems, SIAM J. Numer. Anal. 47 (2) (2009) 1319–1365.

[22] B. Cockburn, B. Dong, J. Guzmán, A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems, Math. Comp. 77 (264) (2008) 1887–1916.

[23] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations, J. Comput. Phys. 228 (9) (2009) 3232–3254.

[24] N. C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations, J. Comput. Phys. 228 (23) (2009) 8841–8855.

[25] N. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow, Comput. Methods Appl. Mech. Eng. 199 (9-12) (2010) 582–597.

[26] B. Cockburn, J. Gopalakrishnan, N. C. Nguyen, J. Peraire, F.-J. Sayas, Analysis of HDG methods for Stokes flow, Math. Comp. 80 (274) (2011) 723–760.

[27] G. Giorgiani, S. Fernández-Méndez, A. Huerta, Hybridizable Discontinuous Galerkin p-adaptivity for wave propagation problems, Int. J. Numer. Methods Fluids 72 (12) (2013) 1244–1262.

[28] Y. Chen, B. Cockburn, Analysis of variable-degree HDG methods for convection-diffusion equations. part I: Semimatching nonconforming meshes, IMA J. Numer. Anal. 32 (4) (2012) 1267–1293.

[29] Y. Chen, B. Cockburn, Analysis of variable-degree HDG methods for convection-diffusion equations. part II: Semimatching nonconforming meshes, Math. Comp.To appear.

[30] J. Donea, A. Huerta, Finite element methods for flow problems, John Wiley & Sons, Chichester (UK), 2003.

[31] A. Montlaur, S. Fernández-Méndez, A. Huerta, Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations, Int. J. Numer. Methods Fluids 57 (9) (2008) 1071–1092.

[32] M. A. Taylor, B. A. Wingate, R. E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal 38 (2000) 1707–1720.

[33] M. Ainsworth, J. T. Oden, A posteriori error estimation in finite element analysis, Pure and Applied Mathematics (New York), Wiley-Interscience [John Wiley & Sons], New York, 2000.

[34] A. Huerta, A. Rodriguez-Ferran, P. Díez, J. Sarrate, Adaptive finite element strategies based on error assessment, Int. J. Numer. Methods Eng. 46 (10) (1999) 1803–1818.

[35] P. Díez, M. Arroyo, A. Huerta, Adaptivity based on error estimation for viscoplastic softening materials, Mech. Cohesive-Frict. Mater. 5 (2) (2000) 87–112.

[36] A. M. Sauer-Budge, J. Bonet, A. Huerta, J. Peraire, Computing bounds for linear functionals of exact weak solutions to Poisson's equation, SIAM J. Numer. Anal. 42 (4) (2004) 1610–1630.

[37] N. Parés, J. Bonet, A. Huerta, J. Peraire, The computation of bounds for linear-functional outputs of weak solutions to the two-dimensional elasticity equations, Comput. Methods Appl. Mech. Eng. 195 (4-6) (2006) 406–429.

[38] N. Parés, P. Díez, A. Huerta, Bounds of functional outputs for parabolic problems. II. Bounds of the exact solution, Comput. Methods Appl. Mech. Eng. 197 (19-20) (2008) 1661–1679.

[39] N. Parés, P. Díez, A. Huerta, Exact bounds for linear outputs of the advection-diffusion-reaction equation using flux-free error estimates, SIAM J. Sci. Comput. 31 (4) (2009) 3064–3089.

[40] P. Díez, J. J. Rodenas, O. C. Zienkiewicz, Equilibrated patch recovery error estimates: simple and accurate upper bounds of the error, Int. J. Numer. Methods Eng. 69 (10) (2007) 2075–2098.

[41] P. Ladevèze, J.-P. Pelle, Mastering calculations in linear and nonlinear mechanics, Mechanical Engineering Series, Springer-Verlag, New York, 2005, translated from the 2001 French original by Theofanis Strouboulis.

[42] N. Parés, P. Díez, A. Huerta, Subdomain-based flux-free a posteriori error estimators, Comput. Methods Appl. Mech. Eng. 195 (4-6) (2006) 297–323.

[43] P. Díez, A. Huerta, A unified approach to remeshing strategies for finite element h-adaptivity, Comput. Methods Appl. Mech. Eng. 176 (1-4) (1999) 215–229.

[44] C. Y. Wang, Exact solutions of the steady-state Navier-Stokes equations, Annual Review of Fluid Mechanics 23 (1) (1991) 159–177.

[45] O. L. Maître, R. Scanlan, O. Knio, Estimation of the flutter derivatives of a NACA airfoil by means of Navier-Stokes simulation, J. Fluids Struct. 17 (1) (2003) 1–28.

[46] J. Peraire, N. C. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations, in: In Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit, AIAA-2010, Orlando, Florida,, 2010.

[47] R. Sevilla, S. Fernández-Méndez, A. Huerta, NURBS-enhanced finite element method (NEFEM), Int. J. Numer. Methods Eng. 76 (1) (2008) 56–83.

[48] R. Sevilla, S. Fernández-Méndez, A. Huerta, Comparison of high-order curved finite elements, Int. J. Numer. Methods Eng. 87 (8) (2011) 719–734.