J Sci Comput (2018) 77:1953–1980
https://doi.org/10.1007/s10915-018-0657-2

HDG-NEFEM with Degree Adaptivity for Stokes Flows

Ruben Sevilla[1] · **Antonio Huerta**[2]

Received: 1 September 2017 / Revised: 8 December 2017 / Accepted: 24 January 2018 /
Published online: 7 February 2018

Abstract The NURBS-enhanced finite element method (NEFEM) combined with a hybridisable discontinuous Galerkin (HDG) approach is presented for the first time. The proposed technique completely eliminates the uncertainty induced by a polynomial approximation of curved boundaries that is common within an isoparametric approach and, compared to other DG methods, provides a significant reduction in number of degrees of freedom. In addition, by exploiting the ability of HDG to compute a postprocessed solution and by using a local a priori error estimate valid for elliptic problems, an inexpensive, reliable and computable error estimator is devised. The proposed methodology is used to solve Stokes flow problems using automatic degree adaptation. Particular attention is paid to the importance of an accurate boundary representation when changing the degree of approximation in curved elements. Several strategies are compared and the superiority and reliability of HDG-NEFEM with degree adaptation is illustrated.

Keywords Hybridisable discontinuous Galerkin · NURBS-enhanced finite element method · Degree adaptivity · Stokes

Mathematics Subject Classification 65N08 · 65N12 · 65N22 · 65N30

✉ Ruben Sevilla
r.sevilla@swansea.ac.uk

Antonio Huerta
antonio.huerta@upc.edu

[1] Zienkiewicz Centre for Computational Engineering, College of Engineering, Swansea University, Bay Campus, Wales SA1 8EN, UK

[2] Laboratori de Calcul Numeric (LaCàN), ETS de Ingenieros de Caminos, Canales y Puertos, Universitat Politecnica de Catalunya, BarcelonaTech, Barcelona, Spain

1 Introduction

Early work on mesh and degree adaptivity schemes for the finite element method [30,44,65] already showed the advantages of adaptive schemes to achieve a required accuracy in an economic manner. The use of mesh adaptive methods is substantially more extended due to the popularity of low-order methods in the computational mechanics community. This is largely due, as discussed later, to the fact that mesh adaptation is easier to implement, compared to degree adaptivity, in standard finite element codes. But, with recent needs on high fidelity simulations for fluids and wave propagation phenomena [14,25,63], the interest in degree adaptive (or the combination of mesh and degree adaptivity) processes has increased [4,23,24,31].

One of the main reasons for the increasing popularity of degree adaptive schemes in the last years is the rise of discontinuous Galerkin (DG) methods as a viable alternative for convection dominated flow and wave propagation problems [13,15,21,26,43,58]. In a standard continuous Galerkin framework, the implementation of variable degree of approximation is cumbersome, whereas its application in a DG context is straightforward due to the weak imposition of the continuity of the solution by means of numerical fluxes. Despite traditional DG methods have not been able to consistently prove its superiority against low-order techniques traditionally employed in industry (e.g. finite volume methods), the recently proposed hybridisable DG (HDG) [11] has shown its superiority compared to traditional DG methods [9,27,33]. The ability to substantially reduce the number of degrees of freedom combined with the possibility to obtain a post-processed solution that converges at a faster rate to the exact solution are the two main properties of HDG methods behind its superiority compared to other DG methods [10,12,38,56]. Moreover, this is achieved while preserving the well-known advantages of DG for stabilising convection and circumventing the so-called Ladyzhenskaya-Babuška-Brezzi (LBB) condition in the incompressible limit.

A key aspect in any adaptive scheme is the ability to devise cheap and reliable error measures for a given numerical solution in order to decide the regions where a more accurate solution is required [1]. Error indicators and error estimators are typically employed to asses the error of a simulation with an adaptive framework [29]. Error indicators are computationally inexpensive but they are problem dependent whereas error estimators are considerably more expensive but more general [19,46,47]. A cheap, general and reliable error estimator was proposed in [23,24] by exploiting the ability of the HDG method to construct a post-processed solution, more accurate than the HDG solution.

One of the aspects that is normally ignored when devising degree adaptive schemes is the geometric representation of domains with curved boundaries. Despite it is now well known that a poor representation of the geometry can have an important effect on the results of a finite element simulation [2,7,54,60], the most extended practice consists on maintaining the shape of the elements during the degree adaptive process [23,24,31]. In the majority of cases, a polynomial representation of the boundary is selected whereas the polynomial degree of the functional approximation changes at each iteration of the degree adaptive scheme.

This work analyses and discusses three approaches to perform a degree adaptive process in domains with curved boundaries. The first one corresponds to the approach typically employed in practice, consisting of fixing the shape of the curved elements and changing the degree of the functional approximation as dictated by the degree adaptivity procedure. The second approach proposed in this work is to employ the so-called NURBS-enhanced finite element method (NEFEM) that enables to exactly represent the geometry of the computational domain, given by a CAD model, irrespectively of the degree of the polynomials used to approximate the solution. The third approach, despite not considered useful from a practical point of view, consists of changing the geometry representation of the computational domain to represent with the same degree of polynomials both the geometry and the solution at each iteration of the degree adaptive process. This approach is not considered of interest from a practical point of view because it requires communication with the CAD model at each iteration and re-generation of nodal distributions for curved elements.

The second approach proposed here considers, for the first time, the combination of the so-called NURBS-enhanced finite element method (NEFEM) and the HDG rationale. The resulting method combines all the advantages of both methods, that is the efficiency of HDG and the ability of NEFEM to decouple the functional approximation from the geometric representation, usually tied in traditional isoparametric implementations.

A number of numerical examples is considered in order to compare the different degree adaptivity approaches. Furthermore, this work presents a simple idea to verify computational methods that are able to use different degrees of approximation for the solution in different elements. The idea is based on an existing local a priori error estimator developed in [18] for elliptic problems.

The remainder of the paper is organised as follows. Section 2 briefly presents the model problem considered (i.e. Stokes flows) and the HDG formulation. The spatial discretisation of the HDG weak formulation is presented in Sect. 3 for both isoparametric and NEFEM, with particular emphasis on the differences between both formulations. The details about the proposed error estimator and degree adaptivity process proposed are presented in Sect. 4, including a discussion of the three approaches considered to perform a degree adaptive process. In Sect. 5 a simple technique to verify the implementation of a solver with variable degree of approximation is presented and used to test the implementation of the HDG code for Stokes flows with isoparametric and NEFEM. Section 6 presents a comparison of the different degree adaptive approaches and a number of numerical examples are used in section to show the potential of the proposed approach. Finally, Sect. 8 summarises the main conclusions of the work that has been presented.

2 Hybridisable Discontinuous Galerkin for Stokes Flow

2.1 Problem statement

Let us consider an open bounded domain $\Omega \in \mathbb{R}^{\mathtt{n_{sd}}}$ with boundary $\partial\Omega$, where $\mathtt{n_{sd}}$ the number of spatial dimensions. The strong form of the stationary Stokes problem is obtained by neglecting the transient and convective effects in the full incompressible Navier–Stokes equations [20]. The so-called velocity-pressure formulation is obtained by invoking the Stoke's law and results in

$$\begin{cases} -\nabla\cdot(\nu\nabla\boldsymbol{u}-p\boldsymbol{I})=\boldsymbol{s} & \text{in } \Omega, \\ \nabla\cdot\boldsymbol{u}=0 & \text{in } \Omega, \\ \boldsymbol{u}=\boldsymbol{u}_D & \text{on } \Gamma_D, \\ \boldsymbol{n}\cdot(\nu\nabla\boldsymbol{u}-p\boldsymbol{I})=\boldsymbol{t} & \text{on } \Gamma_N, \end{cases} \tag{1}$$

where $\boldsymbol{u}$ is the velocity vector, ν is the kinematic viscosity, p denotes the dynamic pressure, $\boldsymbol{s}$ is a body force, $\boldsymbol{u}_D$ is the imposed velocity on the Dirichlet boundary Γ_D, $\boldsymbol{n}$ is the outward unit normal vector to $\partial\Omega$ and $\boldsymbol{t}$ is the *pseudo-traction* vector imposed on the Neumann boundary Γ_N. The disjoint boundaries Γ_D and Γ_N satisfy $\partial\Omega=\overline{\Gamma}_D\cup\overline{\Gamma}_N$.

In what follow, $\left(\cdot,\cdot\right)_D$ denotes the $\mathcal{L}_2$ scalar product in a generic subdomain $D \subset \Omega$, that is

$$\left(u, v\right)_D = \int_D uv d\Omega \qquad \left(\boldsymbol{u}, \boldsymbol{v}\right)_D = \int_D \boldsymbol{u} \cdot \boldsymbol{v} d\Omega \quad \text{and} \quad \left(\boldsymbol{U}, \boldsymbol{V}\right)_D = \int_D \boldsymbol{U} : \boldsymbol{V} d\Omega,$$

for scalars, vectors and second order tensors respectively. Analogously, $\left\langle\cdot,\cdot\right\rangle_S$ denotes the $\mathcal{L}_2$ scalar product in any domain $S \subset \Gamma \cup \partial\Omega$.

The free divergence condition in Eq. (1) induces the compatibility condition

$$\left\langle \boldsymbol{u}_D \cdot \boldsymbol{n}, 1\right\rangle_{\Gamma_D} + \left\langle \boldsymbol{u} \cdot \boldsymbol{n}, 1\right\rangle_{\Gamma_N} = 0. \tag{2}$$

It is worth noting that, if only Dirichlet boundary conditions are considered (i.e. $\Gamma_N = \emptyset$), an additional constraint on the pressure must be imposed to avoid its indeterminacy. It is common [10,12,38] to impose the mean pressure on the element boundary, namely

$$\left\langle p, 1\right\rangle_{\partial\Omega} = 0. \tag{3}$$

2.2 HDG Weak Formulation

The domain Ω is assumed partitioned in $\mathtt{n_{el}}$ disjoint subdomains Ω_e with boundaries $\partial\Omega_e$, which define an internal interface Γ

$$\Gamma := \left[\bigcup_{e=1}^{\mathtt{n_{el}}} \partial\Omega_e\right] \setminus \partial\Omega. \tag{4}$$

The corresponding strong form of the Stokes system given in Eq. (1) can be written in mixed form and in the broken computational domain as

$$\begin{cases} \boldsymbol{L} + \nabla\boldsymbol{u} = \boldsymbol{0} & \text{in } \Omega_e, \\ \nabla\cdot(\nu\boldsymbol{L} + p\boldsymbol{I}) = \boldsymbol{s} & \text{in } \Omega_e, \\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega_e, \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \partial\Omega_e \cap \Gamma_D, \\ \boldsymbol{u} = \hat{\boldsymbol{u}} & \text{on } \partial\Omega_e \setminus \Gamma_D, \\ \boldsymbol{n}\cdot(\nu\boldsymbol{L} + p\boldsymbol{I}) = -\boldsymbol{t} & \text{on } \partial\Omega_e \cap \Gamma_N, \\ [\![\boldsymbol{u}\otimes\boldsymbol{n}]\!] = \boldsymbol{0} & \text{on } \Gamma, \\ [\![\boldsymbol{n}\cdot(\nu\boldsymbol{L} + p\boldsymbol{I})]\!] = \boldsymbol{0} & \text{on } \Gamma, \end{cases} \tag{5}$$

for $e = 1, \ldots, \mathtt{n_{el}}$, where $\boldsymbol{I}$ is the identity tensor of dimension $\mathtt{n_{sd}}$, $\boldsymbol{L} = -\nabla\boldsymbol{u}$ is a new variable (the second order velocity gradient tensor) which is introduced after splitting the second order momentum conservation equation in two first order equations and $\hat{\boldsymbol{u}}$ is an independent variable representing the trace of the solution in $\partial\Omega_e \setminus \Gamma_D$.

The free divergence condition in Eq. (5) induces the compatibility condition

$$\left\langle \boldsymbol{u}_D \cdot \boldsymbol{n}, 1\right\rangle_{\partial\Omega_e\cap\Gamma_D} + \left\langle \hat{\boldsymbol{u}} \cdot \boldsymbol{n}, 1\right\rangle_{\partial\Omega_e\setminus\Gamma_D} = 0. \tag{6}$$

The last two equations in (5) impose the continuity of velocity and continuity of the normal component of the pseudo-stress across the interior faces respectively, where the *jump* $[\![\cdot]\!]$ operator has been introduced following the definition in [37], such that, along each portion of the interface Γ it sums the values from the element on the left and right of say, Ω_e and Ω_l, namely

$$[\![\odot]\!] = \odot_e + \odot_l.$$

The HDG method solves problem (5) in two stages, see for instance [8,11,40–42]. First, an element-by-element problem is defined with $(\boldsymbol{L}, \boldsymbol{u}, p)$ as unknowns. This is the so-called local problem and is given by

$$\begin{cases} \boldsymbol{L} + \boldsymbol{\nabla} \boldsymbol{u} = \boldsymbol{0} & \text{in } \Omega_e, \\ \boldsymbol{\nabla} \cdot (\nu \boldsymbol{L} + p\boldsymbol{I}) = \boldsymbol{s} & \text{in } \Omega_e, \\ \boldsymbol{\nabla} \cdot \boldsymbol{u} = 0 & \text{in } \Omega_e, \\ \boldsymbol{u} = \boldsymbol{u}_D & \text{on } \partial\Omega_e \cap \Gamma_D, \\ \boldsymbol{u} = \hat{\boldsymbol{u}} & \text{on } \partial\Omega_e \setminus \Gamma_D, \\ \langle p, 1 \rangle_{\partial\Omega_e} = \rho_e, \end{cases} \tag{7}$$

for $e = 1, \dots, \mathtt{n_{el}}$, where the last equation in (7) has been introduced to remove the indeterminacy of the pressure and ρ_e denotes the mean pressure on the boundary of element Ω_e. The local problem is used to obtain the solution in each element, $\boldsymbol{L}$, $\boldsymbol{u}$ and p, for $e = 1, \dots, \mathtt{n_{el}}$, in terms of $\hat{\boldsymbol{u}}$ and ρ along the interface $\Gamma \cup \Gamma_N$.

Second, a global problem is defined to determine the traces of the velocity and the mean pressure, denoted by $\hat{\boldsymbol{u}}$ and ρ, on the element boundaries. This is given by

$$\begin{cases} [\![\boldsymbol{u} \otimes \boldsymbol{n}]\!] = \boldsymbol{0} & \text{on } \Gamma, \\ [\![\boldsymbol{n} \cdot (\nu \boldsymbol{L} + p\boldsymbol{I})]\!] = \boldsymbol{0} & \text{on } \Gamma, \\ \boldsymbol{n} \cdot (\nu \boldsymbol{L} + p\boldsymbol{I}) = -\boldsymbol{t} & \text{on } \Gamma_N \end{cases} \tag{8}$$

where the first equation is automatically satisfied due to the unique definition of the hybrid variable $\hat{\boldsymbol{u}}$ on each face of the mesh skeleton and the condition $\boldsymbol{u} = \hat{\boldsymbol{u}}$ on Γ, as imposed in the local problem.

The weak formulation for each element equivalent to (7) is as follows: for $e = 1, \dots, \mathtt{n_{el}}$, given $\boldsymbol{u}_D$ on Γ_D and $\hat{\boldsymbol{u}}$ on $\Gamma \cup \Gamma_N$, find $(\boldsymbol{L}, \boldsymbol{u}, p) \in \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}} \times \mathtt{n_{sd}}} \times \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}}} \times \mathcal{L}_2(\Omega_e)$ that satisfies

$$-\left(\boldsymbol{G}, \boldsymbol{L}\right)_{\Omega_e} + \left(\boldsymbol{\nabla} \cdot \boldsymbol{G}, \boldsymbol{u}\right)_{\Omega_e} = \left\langle \boldsymbol{n} \cdot \boldsymbol{G}, \boldsymbol{u}_D \right\rangle_{\partial\Omega_e \cap \Gamma_D} + \left\langle \boldsymbol{n} \cdot \boldsymbol{G}, \hat{\boldsymbol{u}} \right\rangle_{\partial\Omega_e \setminus \Gamma_D} \tag{9a}$$

$$-\left(\boldsymbol{\nabla} \boldsymbol{v}, \nu \boldsymbol{L}\right)_{\Omega_e} - \left(\boldsymbol{\nabla} \cdot \boldsymbol{v}, p\right)_{\Omega_e} + \left\langle \boldsymbol{v}, \boldsymbol{n} \cdot (\nu \widehat{\boldsymbol{L} + p} \boldsymbol{I}) \right\rangle_{\partial\Omega_e} = \left(\boldsymbol{v}, \boldsymbol{s}\right)_{\Omega_e} \tag{9b}$$

$$\left(\boldsymbol{\nabla} w, \boldsymbol{u}\right)_{\Omega_e} = \left\langle w, \boldsymbol{u}_D \cdot \boldsymbol{n} \right\rangle_{\partial\Omega_e \cap \Gamma_D} + \left\langle w, \hat{\boldsymbol{u}} \cdot \boldsymbol{n} \right\rangle_{\partial\Omega_e \setminus \Gamma_D} \tag{9c}$$

$$\left\langle p, 1 \right\rangle_{\partial\Omega_e} = \rho_e \tag{9d}$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w) \in \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}} \times \mathtt{n_{sd}}} \times \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}}} \times \mathcal{L}_2(\Omega_e)$, where the numerical trace of the normal flux is defined as

$$\boldsymbol{n} \cdot (\nu \widehat{\boldsymbol{L} + p} \boldsymbol{I}) := \begin{cases} \boldsymbol{n} \cdot (\nu \boldsymbol{L} + p\boldsymbol{I}) + \tau(\boldsymbol{u} - \boldsymbol{u}_D) & \text{on } \partial\Omega_e \cap \Gamma_D \\ \boldsymbol{n} \cdot (\nu \boldsymbol{L} + p\boldsymbol{I}) + \tau(\boldsymbol{u} - \hat{\boldsymbol{u}}) & \text{elsewhere,} \end{cases} \tag{10}$$

with τ being a stabilisation parameter, whose selection has an important effect on the stability, accuracy and convergence properties of the resulting HDG method. The influence of the stabilisation parameter has been studied extensively by Cockburn and co-workers, see for instance [8,11,39–42].

Introducing the definition of the numerical trace of the normal flux in Eq. 9 leads to the weak form of the local problem: for $e = 1, \dots, \mathtt{n_{el}}$, find $(\boldsymbol{L}, \boldsymbol{u}, p) \in \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}} \times \mathtt{n_{sd}}} \times \left[\mathcal{H}^1(\Omega_e)\right]^{\mathtt{n_{sd}}} \times \mathcal{L}_2(\Omega_e)$ such that

$$
\begin{aligned}
-\big(\boldsymbol{G}, \boldsymbol{L}\big)_{\Omega_e} + \big(\nabla\cdot\boldsymbol{G}, \boldsymbol{u}\big)_{\Omega_e} &= \big\langle \boldsymbol{n}\cdot\boldsymbol{G}, \boldsymbol{u}_D\big\rangle_{\partial\Omega_e\cap\Gamma_D} + \big\langle \boldsymbol{n}\cdot\boldsymbol{G}, \hat{\boldsymbol{u}}\big\rangle_{\partial\Omega_e\setminus\Gamma_D}, && \text{(11a)}\\
&\quad -\big(\nabla\boldsymbol{v}, \nu\boldsymbol{L}\big)_{\Omega_e} + \big\langle \boldsymbol{v}, \boldsymbol{n}\cdot\nu\boldsymbol{L}\big\rangle_{\partial\Omega_e} + \big\langle \boldsymbol{v}, \tau\boldsymbol{u}\big\rangle_{\partial\Omega_e}\\
&\quad -\big(\nabla\cdot\boldsymbol{v}, p\big)_{\Omega_e} + \big\langle \boldsymbol{v}, p\boldsymbol{n}\big\rangle_{\partial\Omega_e}\\
&= \big(\boldsymbol{v}, \boldsymbol{s}\big)_{\Omega_e} + \big\langle \boldsymbol{v}, \tau\boldsymbol{u}_D\big\rangle_{\partial\Omega_e\cap\Gamma_D} + \big\langle \boldsymbol{v}, \tau\hat{\boldsymbol{u}}\big\rangle_{\partial\Omega_e\setminus\Gamma_D}, && \text{(11b)}\\
\big(\nabla w, \boldsymbol{u}\big)_{\Omega_e} &= \big\langle w, \boldsymbol{u}_D\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_e\cap\Gamma_D} + \big\langle w, \hat{\boldsymbol{u}}\cdot\boldsymbol{n}\big\rangle_{\partial\Omega_e\setminus\Gamma_D} && \text{(11c)}\\
\big\langle p, 1\big\rangle_{\partial\Omega_e} &= \rho && \text{(11d)}
\end{aligned}
$$

for all $(\boldsymbol{G}, \boldsymbol{v}, w) \in \big[\mathcal{H}^1(\Omega_e)\big]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}} \times \big[\mathcal{H}^1(\Omega_e)\big]^{\mathtt{n_{sd}}} \times \mathcal{L}_2(\Omega_e)$.

For the global problem, the weak formulation equivalent to (8) is: find $\hat{\boldsymbol{u}} \in [\mathcal{L}_2(\Gamma\cup\Gamma_N)]^{\mathtt{n_{sd}}}$ and $\rho \in \mathbb{R}^{\mathtt{n_{el}}}$ that satisfies

$$
\begin{aligned}
&\sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \big\langle \boldsymbol{\mu}, \boldsymbol{n}\cdot(\nu\widehat{\boldsymbol{L} + p}\boldsymbol{I})\big\rangle_{\partial\Omega_e\setminus\partial\Omega} + \big\langle \boldsymbol{\mu}, \boldsymbol{n}\cdot(\nu\widehat{\boldsymbol{L} + p}\boldsymbol{I}) + \boldsymbol{t}\big\rangle_{\partial\Omega_e\cap\Gamma_N} \Big\} = 0 && \text{(12a)}\\
&\big\langle \hat{\boldsymbol{u}}\cdot\boldsymbol{n}, 1\big\rangle_{\partial\Omega_e\setminus\Gamma_D} = -\big\langle \boldsymbol{u}_D\cdot\boldsymbol{n}, 1\big\rangle_{\partial\Omega_e\cap\Gamma_D} \qquad \text{for } e = 1, \ldots, \mathtt{n_{el}}, && \text{(12b)}
\end{aligned}
$$

for all $\boldsymbol{\mu} \in [\mathcal{L}_2(\Gamma\cup\Gamma_N)]^{\mathtt{n_{sd}}}$.

Introducing the definition of the numerical trace of the normal flux in Eq. 12 leads to the weak form of the global problem: find $\hat{\boldsymbol{u}} \in [\mathcal{L}_2(\Gamma\cup\Gamma_N)]^{\mathtt{n_{sd}}}$ and $\rho \in \mathbb{R}^{\mathtt{n_{el}}}$ such that, for all $\boldsymbol{\mu} \in [\mathcal{L}_2(\Gamma\cup\Gamma_N)]^{\mathtt{n_{sd}}}$,

$$
\begin{aligned}
&\sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \big\langle \boldsymbol{\mu}, \boldsymbol{n}\cdot\nu\boldsymbol{L}\big\rangle_{\partial\Omega_e\setminus\Gamma_D} + \big\langle \boldsymbol{\mu}, \tau\boldsymbol{u}\big\rangle_{\partial\Omega_e\setminus(\Gamma_D\cup\Gamma_S)} + \big\langle \boldsymbol{\mu}, p\boldsymbol{n}\big\rangle_{\partial\Omega_e\setminus\Gamma_D)}\\
&\qquad - \big\langle \boldsymbol{\mu}, \tau\hat{\boldsymbol{u}}\big\rangle_{\partial\Omega_e\setminus\Gamma_D} \Big\} = -\sum_{e=1}^{\mathtt{n_{el}}} \Big\{ \big\langle \boldsymbol{\mu}, \boldsymbol{t}\big\rangle_{\partial\Omega_e\cap\Gamma_N} \Big\}, && \text{(13a)}\\
&\big\langle \hat{\boldsymbol{u}}\cdot\boldsymbol{n}, 1\big\rangle_{\partial\Omega_e\setminus\Gamma_D} = -\big\langle \boldsymbol{u}_D\cdot\boldsymbol{n}, 1\big\rangle_{\partial\Omega_e\cap\Gamma_D}. && \text{(13b)}
\end{aligned}
$$

3 Spatial Discretisation

This section presents the discretisation of the HDG weak forms derived in the previous section. Both the standard isoparametric and the so-called NEFEM formulations are presented. Special attention is paid to the differences between both formulations as this represents the first time NEFEM is considered in an HDG framework.

3.1 Isoparametric Elements

Standard isoparametric formulations map each element Ω_e and face Γ_e in the physical domain into a reference element, $\widetilde{\Omega}$, and a reference face, $\widetilde{\Gamma}$, where polynomial functional approximations characterize the discrete finite dimensional spaces. Namely, $\mathcal{P}^k(\widetilde{\Omega})$ and $\mathcal{P}^{\hat{k}}(\widetilde{\Gamma})$ are the spaces of polynomial functions of degree at most $k \geq 1$ and $\hat{k} \geq 1$ in the reference element and the reference face respectively. Finally, the approximations for each variable are defined as

$$
\boldsymbol{u}(\boldsymbol{\xi}) \simeq \boldsymbol{u}^h(\boldsymbol{\xi}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathbf{u}_j N_j(\boldsymbol{\xi}) \qquad \in \Big[\{v \in \mathcal{L}_2(\Omega);\ v|_{\Omega_e} \in \mathcal{P}^k(\widetilde{\Omega})\}\Big]^{\mathtt{n_{sd}}}, \tag{14}
$$

$$p(\boldsymbol{\xi}) \simeq p^h(\boldsymbol{\xi}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathrm{p}_j N_j(\boldsymbol{\xi}) \qquad \in \{q \in \mathcal{L}_2(\Omega);\ q|_{\Omega_e} \in \mathcal{P}^k(\widetilde{\Omega})\}, \tag{15}$$

$$\boldsymbol{L}(\boldsymbol{\xi}) \simeq \boldsymbol{L}^h(\boldsymbol{\xi}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathbf{L}_j N_j(\boldsymbol{\xi}) \qquad \in \left[\{v \in \mathcal{L}_2(\Omega);\ v|_{\Omega_e} \in \mathcal{P}^k(\widetilde{\Omega})\}\right]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}, \tag{16}$$

$$\hat{\boldsymbol{u}}(\boldsymbol{\eta}) \simeq \hat{\boldsymbol{u}}^h(\boldsymbol{\eta}) = \sum_{j=1}^{\mathtt{n_{fn}}} \hat{\mathbf{u}}_j \hat{N}_j(\boldsymbol{\eta}) \qquad \in \left[\{v \in \mathcal{L}_2(\Gamma);\ v|_{\Gamma_e} \in \mathcal{P}^{\hat{k}}(\widetilde{\Gamma})\}\right]^{\mathtt{n_{sd}}}, \tag{17}$$

where $\mathbf{u}_j$, p_j, $\mathbf{L}_j$ and $\hat{\mathbf{u}}_j$ are nodal values, N_j are polynomial shape functions of order k in the reference element, $\mathtt{n_{en}}$ is the number of nodes per element, $\hat{N}_j$ are polynomial shape functions of order $\hat{k}$ in the reference face and $\mathtt{n_{fn}}$ is the corresponding number of nodes per face. Note that equal interpolation is used for all element variables (i.e. velocity, pressure and gradient of velocity). Recall that HDG allows for equal interpolation because of the numerical fluxes and the stabilisation parameter τ. They ensure solvability and stability, see [8], without the need of an enriched space for the gradient variable, or a reduced space for the trace variable.

An isoparametric mapping is used to link the reference element $\hat{\Omega}$ and the computational element Ω_e^h

$$\begin{aligned} \boldsymbol{\varphi}\ :\ \widetilde{\Omega} \subset \mathbb{R}^{\mathtt{n_{sd}}} &\longrightarrow \Omega_e^h \subset \mathbb{R}^{\mathtt{n_{sd}}} \\ \boldsymbol{\xi} &\longmapsto \boldsymbol{\varphi}(\boldsymbol{\xi}) := \sum_{j=1}^{\mathtt{n_{en}}} \boldsymbol{x}_j N_j(\boldsymbol{\xi}), \end{aligned} \tag{18}$$

where $\boldsymbol{x}_j$ are the nodal coordinates of the computational element Ω_e^h.

It is worth noting that in general, when the physical element Ω_e is curved, the isoparametric mapping is non-linear and the approximation defined in the reference element do not induce a polynomial interpolation in the physical space. In addition, the computational element Ω_e^h is just an approximation of Ω_e, see [53] for a detailed discussion.

Similarly, an isoparametric mapping is used to link the reference face $\hat{\Gamma}$ and the computational face Γ_e^h

$$\begin{aligned} \boldsymbol{\psi}\ :\ \widetilde{\Gamma} \subset \mathbb{R}^{\mathtt{n_{sd}}-1} &\longrightarrow \Gamma_e^h \subset \mathbb{R}^{\mathtt{n_{sd}}} \\ \boldsymbol{\eta} &\longmapsto \boldsymbol{\psi}(\boldsymbol{\eta}) := \sum_{j=1}^{\mathtt{n_{fn}}} \boldsymbol{x}_j \hat{N}_j(\boldsymbol{\eta}), \end{aligned} \tag{19}$$

where $\boldsymbol{x}_j$ denote the face nodal coordinates.

Using the mappings in Eqs. (18) and (19), the integrals appearing in the weak form of the local problems are transformed to the reference element and reference face/edge respectively. Then, the nodal interpolations given by Eqs. (14)–(17) are introduced, leading to a system of equations for each element with the following structure

$$\begin{bmatrix} \mathbf{A}_{LL} & \mathbf{A}_{Lu} & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{uL} & \mathbf{A}_{uu} & \mathbf{A}_{up} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{pu} & \mathbf{0} & \boldsymbol{a}_{\rho p}^T \\ \mathbf{0} & \mathbf{0} & \boldsymbol{a}_{\rho p} & 0 \end{bmatrix} \begin{Bmatrix} \mathbf{L} \\ \mathbf{u} \\ \mathbf{p} \\ \zeta \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_L \\ \mathbf{f}_u \\ \mathbf{f}_p \\ 0 \end{Bmatrix} + \begin{bmatrix} \mathbf{A}_{L\hat{u}} \\ \mathbf{A}_{u\hat{u}} \\ \mathbf{A}_{p\hat{u}} \\ \mathbf{0} \end{bmatrix} \hat{\mathbf{u}} + \begin{Bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ 1 \end{Bmatrix} \rho, \tag{20}$$

where ζ is the Lagrange multiplier corresponding to the constraint of Eq. (11d).

Analogously, using the isoparametric mappings, the nodal interpolations of the corresponding variables and introducing the expression of $\mathbf{L}$, $\mathbf{u}$ and $\mathbf{p}$ from Eq. (20) in the global problem of Eq. (13), a global system of equations is obtained

$$\widehat{\mathbf{K}}\hat{\mathbf{U}} = \hat{\mathbf{f}}, \tag{21}$$

where the vector of unknowns $\hat{\mathbf{U}}$ contains the nodal values of the trace of the velocity on the elementa faces and the mean pressure within each element.

3.2 NEFEM Elements

In NEFEM, the boundary of the computational domain $\partial\Omega$ is exactly represented by NURBS. In what follows, in order to simplify the presentation and without loss of generality, the NURBS are restricted to two dimensional problems, see [52] for a detailed description of the three dimensional case. An edge is given by $\Gamma_e := \boldsymbol{C}([\lambda_a^e, \lambda_b^e])$, where $\boldsymbol{C}$ is the NURBS boundary parametrisation and λ_a and λ_b are the parametric coordinates (in the parametric space of the NURBS) of the end points of Γ_e.

The discrete approximations are defined now as:

$$\boldsymbol{u}(\boldsymbol{x}) \simeq \boldsymbol{u}^h(\boldsymbol{x}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathbf{u}_j N_j(\boldsymbol{x}) \qquad \in \left[\{v \in \mathcal{L}_2(\Omega);\ v|_{\Omega_e} \in \mathcal{P}^k(\Omega_e)\}\right]^{\mathtt{n_{sd}}}, \tag{22}$$

$$p(\boldsymbol{x}) \simeq p^h(\boldsymbol{x}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathrm{p}_j N_j(\boldsymbol{x}) \qquad \in \{q \in \mathcal{L}_2(\Omega);\ q|_{\Omega_e} \in \mathcal{P}^k(\Omega_e)\}, \tag{23}$$

$$\boldsymbol{L}(\boldsymbol{x}) \simeq \boldsymbol{L}^h(\boldsymbol{x}) = \sum_{j=1}^{\mathtt{n_{en}}} \mathbf{L}_j N_j(\boldsymbol{x}) \qquad \in \left[\{v \in \mathcal{L}_2(\Omega);\ v|_{\Omega_e} \in \mathcal{P}^k(\Omega_e)\}\right]^{\mathtt{n_{sd}}\times\mathtt{n_{sd}}}, \tag{24}$$

$$\hat{\boldsymbol{u}}(\boldsymbol{\lambda}) \simeq \hat{\boldsymbol{u}}^h(\boldsymbol{\lambda}) = \sum_{j=1}^{\mathtt{n_{fn}}} \hat{\mathbf{u}}_j \hat{N}_j(\boldsymbol{\lambda}) \qquad \in \left[\{v \in \mathcal{L}_2(\Gamma);\ v|_{[\lambda_a^e,\lambda_b^e]} \in \mathcal{P}^{\hat{k}}([\lambda_a^e, \lambda_b^e])\}\right]^{\mathtt{n_{sd}}}, \tag{25}$$

where $\mathbf{u}_j$, p_j, $\mathbf{L}_j$ and $\hat{\mathbf{u}}_j$ are nodal values, N_j are polynomial shape functions of order k in the physical element, $\mathtt{n_{en}}$ is the number of nodes per element, $\hat{N}_j$ are polynomial shape functions of order $\hat{k}$ in $[\lambda_a^e, \lambda_b^e]$ and $\mathtt{n_{fn}}$ is the corresponding number of nodes per face.

The main differences of NEFEM with respect to the isoparametric formulation are:

- The exact description of the computational domain is considered by means of its NURBS boundary representation.
- The approximation of the elemental variables directly in the physical space, with Cartesian coordinates.
- The approximation of the trace of the velocity is defined in the parametric space of the NURBS. It is worth noting that other options could be considered such as defining the approximation directly in the physical space. The main advantage of defining the approximation in the parametric space of the NURBS is that the number of unknowns remains the same as in the isoparametric formulation. In contrast, if the approximation of this variable is selected in the physical space it would require further degrees of freedom [53,54].

In addition, from the computational point of view, NEFEM uses specifically designed numerical quadratures that provide a more efficient alternative to standard quadratures defined in a reference triangle [51,52]. For instance, in two dimensions, the following mapping is introduced between a reference rectangle and the physical element

$$\begin{aligned} \boldsymbol{\psi} : R \subset \mathbb{R}^{\mathtt{nsd}} &\longrightarrow \Omega_e \subset \mathbb{R}^{\mathtt{nsd}} \\ \boldsymbol{\lambda} &\longmapsto \boldsymbol{\psi}(\boldsymbol{\lambda}) := (1-\vartheta)\boldsymbol{C}(\lambda_1) + \lambda_2 \boldsymbol{x}_I, \end{aligned} \tag{26}$$

where $R = [\lambda_a^e, \lambda_b^e] \times [0, 1]$ and $\boldsymbol{x}_I$ is the internal vertex of Ω_e.

Using the mapping in Eq. (26) and the NURBS boundary representation given by $\boldsymbol{C}$, the integrals appearing in the weak form of the local problems are transformed to the reference rectangle and the parametric space of the NURBS respectively. Then, the nodal interpolations given by Eqs. (22), (23), (24) and (25) are introduced, leading to a system of equations similar to Eq. (20). Analogously, the global problem with NEFEM leads to a global system of equations similar to Eq. (21).

4 Error Estimation and Adaptivity

In HDG, the possibility to obtain a postprocessed solution [11] that converges at a higher rate (i.e. $k+2$) than the HDG solution, not only provides a higher accurate solution to the problem at hand but it can also be used to build an inexpensive, reliable and computable error estimator [23,24]. In this section, particular attention is paid to the fact that, when the degree of approximation is changed in a curved element, a choice must be made regarding the geometric definition of the element.

An element by element measure of the error is defined by employing the HDG solution and the postprocessed solution as proposed in [24]

$$E_e = \left[\frac{1}{|\Omega_e|} \int_{\Omega_e} \big(\boldsymbol{u}^\star - \boldsymbol{u}\big) \cdot \big(\boldsymbol{u}^\star - \boldsymbol{u}\big)\, d\Omega \right]^{1/2}, \tag{27}$$

where the normalisation becomes critical when meshes with different element sizes are considered [18].

For elliptic problems, and by using that the influence of pollution errors becomes negligible if the mesh is sufficiently refined in the area where the pollution error is generated [28], the following a priori error estimate was derived in [18]

$$\varepsilon_e = \|\boldsymbol{u} - \boldsymbol{u}_h\|_{\Omega_e} \le C h_e^{k_e + 1 + \mathtt{nsd}/2}, \tag{28}$$

for $e = 1, \ldots, \mathtt{n_{el}}$, where C is a constant, h_e the characteristic element size of Ω_e and k_e the degree of approximation used in Ω_e. It is worth noting that the error estimate of Eq. (28) was initially derived for the standard finite element method but its extension to the HDG method is straightforward.

By applying a standard Richardson extrapolation, it is possible to predict the required change in the degree of approximation in order to ensure that the error in each element is lower than a desired accuracy ϵ, namely

$$\Delta k_e = \left\lceil \frac{\log(\varepsilon / E_e)}{\log(h_e)} \right\rceil \tag{29}$$

for $e = 1, \ldots, \mathtt{n_{el}}$, where $\lceil \cdot \rceil$ denotes the ceiling function.

The adaptive procedure consist on solving the Stokes problem using the HDG formulation as described in Sect. 3 and estimating the required degree of approximation in each element

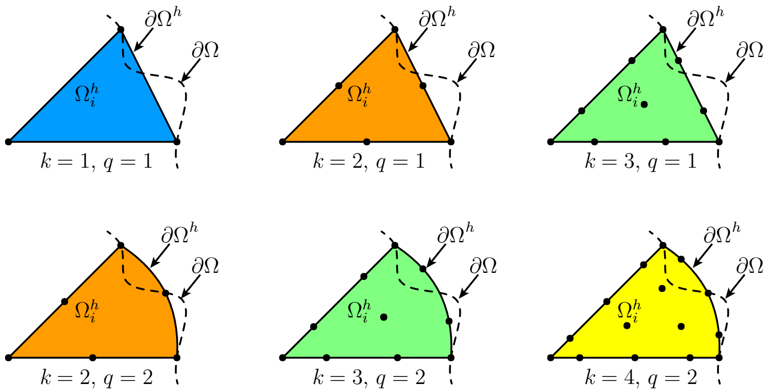


Fig. 1 Illustration of a degree adaptation in an element with linear (top) and quadratic (bottom) approximation of the geometry

according to Eq. (29). The process is repeated until convergence is achieved, meaning that the error in each element ε_e is lower than the desired error ϵ.

4.1 Geometry Update

The technique described to drive a degree adaptive process only focuses on the degree of approximation used for the functional approximation, but in the presence of curved boundaries it is known that high-order approximations of both the solution and the geometry are required to exploit the full potential of a high-order method [2,17,34,55]. This aspect is usually ignored as degree adaptive procedures are applied to problems involving polygonal boundaries, see for instance [16,22,45,61]. Here three options are discussed and assessed and compared later using numerical examples.

The first, and the one typically considered in a degree adaptive process, technique consists of defining a polynomial representation of the curved boundaries that is maintained during the adaptive process, irrespectively of the degree of approximation used for the solution [23,24,35]. This option is attractive because when the degree of approximation is changed in an element, there is no need to communicate with a CAD library to re-generate the nodal distributions in curved elements at each iteration of the degree adaptive process. The strategy is illustrated in Fig. 1. The first row of plots show triangular elements where the geometric approximation is linear ($q = 1$) and the polynomial degree of approximation of the solution increases from $k = 1$ to $k = 3$. The second row shows a similar situation where the boundary of the computational domain is described using quadratic polynomials ($q = 2$) and the degree of approximation of the solution is increased from $k = 2$ to $k = 4$. The boundary of the computational domain is denoted by $\partial\Omega^h$ whereas the exact boundary is denoted by $\partial\Omega$.

The second alternative, proposed in this work, consists of using NEFEM, where the exact boundary representation of the computational domain is considered irrespectively of the degree of approximation considered for the solution. As NEFEM encapsulates the necessary information to define the approximation and perform the numerical integration in curved elements in contact with a NURBS boundary, communication with a CAD library is avoided. The strategy is illustrated in Fig. 2, showing a NEFEM element where the exact boundary representation is considered and a degree of approximation for the solution being increased from $k = 1$ to $k = 3$.

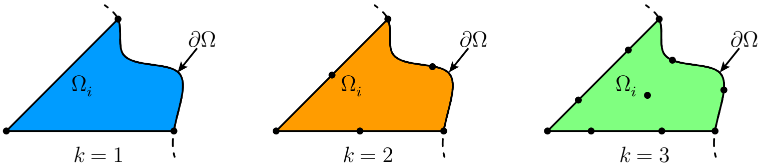


Fig. 2 Illustration of a degree adaptation in a NEFEM element

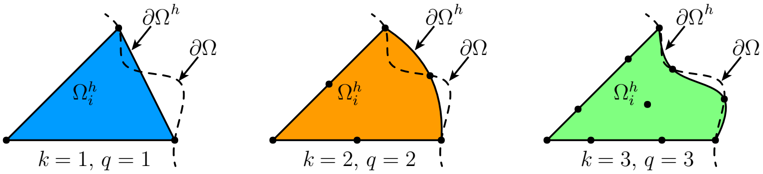


Fig. 3 Illustration of a degree adaptation in an element where the same degree of approximation is used for both the solution and the geometry

A third alternative, not considered in practice, consists of communicating with the CAD model after each iteration of the degree adaptive process in order to re-generate the nodal distribution of curved elements by placing the nodes over the true boundary. The strategy is illustrated in Fig. 3, showing a triangular element where both the degree of the functional approximation and the degree of the polynomials used to approximate the solution are updated at each iteration. This strategy has not been considered in practical applications due to the cost associated to communicating with the CAD model at each iteration.

Remark 1 It is important to note that the first strategy, where the geometry remains unchanged, does not guarantee the convergence of the numerical solution to the physical solution in domains with curved boundaries because the distance between the computational domain and the physical domain does not converge to zero with as the degree of approximation is increased, see [6,49] for more details. For the second strategy, proposed here, convergence to the physical solution is guaranteed because no geometrical error is introduced [54]. Finally, for the third approach, convergence is also guaranteed if the distance between the computational boundary and the physical boundary tends to zero as the order of the approximation is increased and the derivatives of the isoparametric mapping up to order $k + 1$ are bounded by h^s, for $s = 2, \dots, k + 1$ [6,49], where h denotes the characteristic element size. It is worth noting that a specifically designed nodal distribution for curved elements is required in the third approach to guarantee that the second hypothesis is fulfilled [6].

5 Validation of the HDG Formulation with Variable Degree of Approximation

The first example provides a novel and simple technique to fully validate a solver that employs variable different degree of approximation in different elements for the solution of elliptic problems. The idea consists of utilising the local a priori error estimate of Eq. (28) that states how the error, measured in an element, decreases when the mesh is refined.

To illustrate the proposed technique and validate the HDG isoparametric and NEFEM implementations with variable degree of approximation, the Stokes equations are solved in

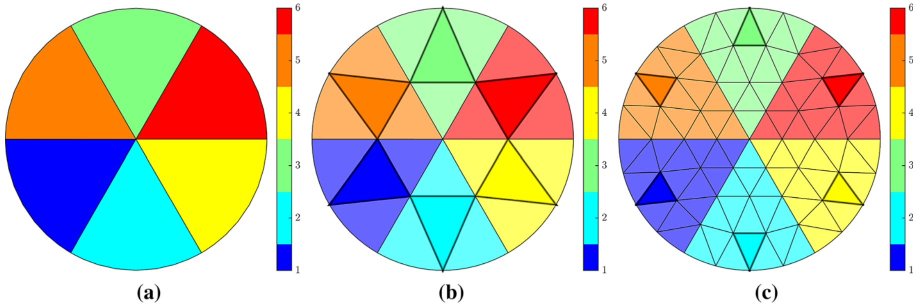


Fig. 4 First three NEFEM meshes where the colour indicates the degree of approximation used in each element and the highlighted element represents the region where the error is measured for each degree of approximation. **a** Mesh 1. **b** Mesh 2. **c** Mesh 3

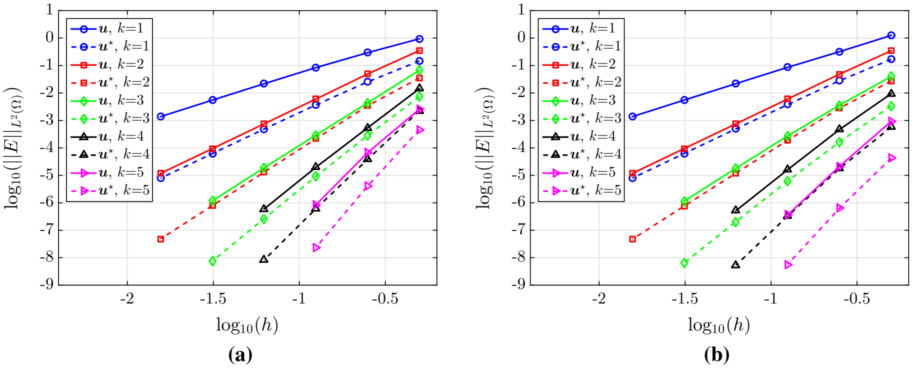


Fig. 5 Error of the solution $\boldsymbol{u}$ and the postprocessed solution $\boldsymbol{u}^\star$ in the $\mathcal{L}_2(\Omega_e)$ norm for different degrees of approximation in each element. **a** HDG Isoparametric. **b** HDG NEFEM

a circle of radius 0.5 centred at (0.5, 0.5) with Dirichlet boundary conditions. The viscosity is considered as $\nu = 1$ and source and boundary conditions are taken such that the analytical solution is given by

$$\boldsymbol{u} = \begin{pmatrix} x^2(1-x)^2(2y-6y^2+4y^3) \\ -y^2(1-y)^2(2x-6x^2+4x^3) \end{pmatrix} \qquad p = x(1-x). \tag{30}$$

Six triangular meshes of the domain are generated using nested refinement. The first three meshes are shown in Fig. 4, where the colour of each element represents the degree of approximation used, ranging from $k = 1$ to $k = 6$. In each mesh, there is one element per degree of approximation highlighted with a thicker line and darker colour, representing the region where the error is measured to test the local a priori error estimate. It is worth noting that the meshes shown in Fig. 4 are NEFEM meshes, as the exact boundary representation is always employed, even for $k = 1$, whereas for the computations both NEFEM and isoparametric meshes are employed.

The results of the h-convergence study are presented in Figs. 5 and 6. Figure 5 shows the error of the solution $\boldsymbol{u}$ and the postprocessed solution $\boldsymbol{u}^\star$ in the $\mathcal{L}_2(\Omega_e)$ norm for isoparametric and NEFEM elements for $k = 1, \ldots, 5$. The optimal rate of convergence is obtained in all cases for both the solution $\boldsymbol{u}$ (rate $k+2$) and the postprocessed solution $\boldsymbol{u}^\star$ (rate $k+3$).

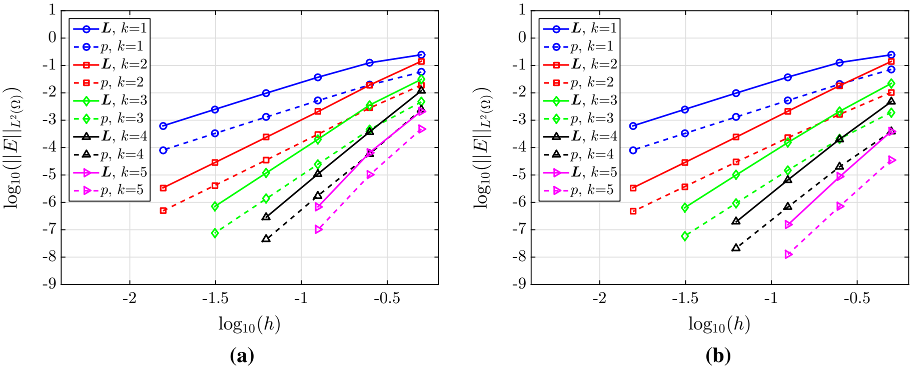


Fig. 6 Error of the dual variable $\boldsymbol{L}$ and the pressure p in the $\mathcal{L}_2(\Omega_e)$ norm for different degrees of approximation in each element. **a** HDG Isoparametric. **b** HDG NEFEM

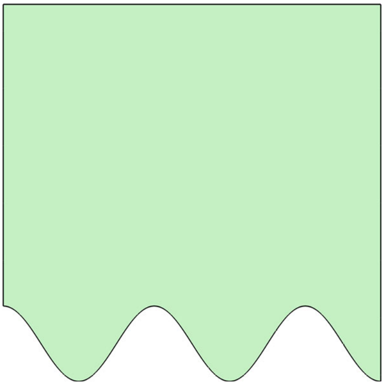


Fig. 7 Computational domain for the test problem used to compare the different geometry update options in a degree adaptive process

In Fig. 6 a similar analysis is conducted, but now the error is measured for the dual variable $\boldsymbol{L}$ and the pressure p, also in the $\mathcal{L}_2(\Omega_e)$ norm and for isoparametric and NEFEM elements.

Again, the optimal rate of convergence is obtained in all cases for both $\boldsymbol{L}$ and p (rate $k+2$).

6 Comparison of Degree Adaptivity Strategies

The same model problem employed in the previous example is utilised to compare the strategies described in Sect. 4.1 to update the geometry during a degree adaptive process. The computational domain selected, shown in Fig. 7, features an oscillatory boundary and represents a common problem encountered in biological transport applications, see for instance [50]. More precisely, the curved part of the boundary is given by the curve $f(x) = (1 + \cos(5\pi x))/10$.

Dirichlet boundary conditions are imposed on the polygonal part of the boundary whereas a Neumann boundary condition, corresponding to the exact traction derived from the solution in Eq. (30) is imposed on the oscillatory part of the boundary.

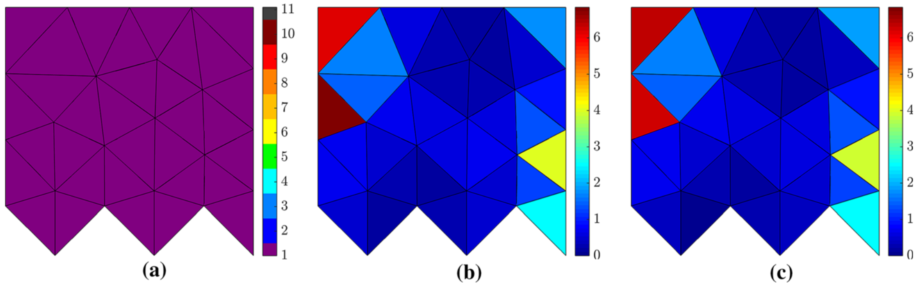


Fig. 8 First iteration of the degree adaptivity procedure with HDG isoparametric elements. **a** Degree. **b** Estimated error. **c** Exact error

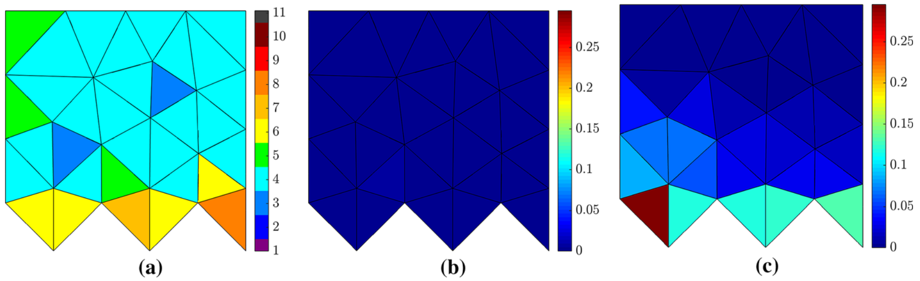


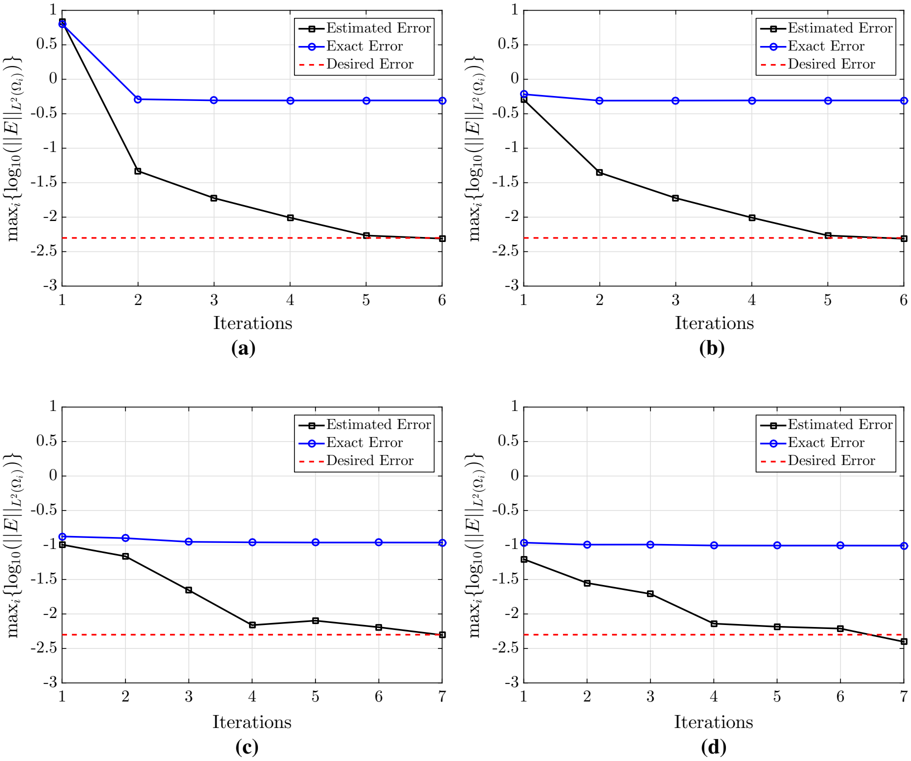
Fig. 9 Sixth iteration of the degree adaptivity procedure with no geometric update using HDG isoparametric elements and $q = 1$. **a** Degree. **b** Estimated error. **c** Exact error

6.1 No Geometric Update

First, the degree adaptive process with no communication with the CAD model is studied, as illustrated in Fig. 1. The process starts with a degree of approximation $k = 1$ in all elements. At each iteration the degree of the functional approximation is adapted according to the strategy presented in Sect. 4 whereas a linear approximation of the geometry is kept irrespectively of the degree of the functional approximation. Figure 8 shows the original mesh, the estimated error and the exact error, computed by using the known analytical solution. The $\mathcal{L}_2$ norm of the error is represented as a constant value in each element, showing a good agreement between the estimated and the exact error.

After six iterations of the adaptivity process, the degree of approximation is adapted in each element as shown in Fig. 9a but a linear geometric approximation of the curved boundary is still considered. The estimated error in each element, shown in Fig. 9b, is below the desired error which is 0.5×10^{-2} in this example but the computation of the exact error, shown in Fig. 9c, reveals a significant disparity when compared to the estimated error.

To better analyse the results, Fig. 10 shows the evolution of the maximum estimated error in each element and the maximum exact in each element for different geometric approximations of the curved boundary. Figure 10a corresponds to the case illustrated in Figs. 8 and 9, where a linear approximation of the geometry is considered ($q = 1$). It can be clearly observed that, as the degree adaptive process evolves, the difference between estimated and exact error becomes more and more sizeable. In the sixth iteration, the adaptive process shows convergence, with an estimated error less than the desired error 0.5×10^{-2} but this is two orders of magnitude lower than the exact error, clearly indicating that the



(a)

(b)

(c)

(d)

Fig. 10 Evolution of the estimated and exact errors during a degree adaptivity process for different degrees of the polynomials used to approximate the geometry (q). **a** $q = 1$. **b** $q = 2$. **c** $q = 3$. **d** $q = 4$

error estimator is not reliable because the estimator assumes that the geometry is exactly represented.

As a linear approximation of the geometry is well known to be not suitable when high order functional approximations are considered [2,7,54,60], the same experiment is repeated by using a more accurate boundary representation. The plots in Fig. 10b–d show the evolution of the maximum estimated error in each element and the maximum exact in each element for quadratic, cubic and quartic approximation of the geometry. In all cases it is clearly observed that the error estimator is not reliable because the adaptive process converges but the exact error is more than one order of magnitude higher than the desired error.

The degree of approximation, estimated error and exact error obtained in the last iteration of the adaptive process for $q = 4$ is represented in Fig. 11. The results show that even with a more accurate geometric approximation, the exact error in the elements close to the curved boundary is much higher than the estimated error. There is clear evidence that, if no communication with a CAD model is undertaken during the degree adaptive process, the original mesh must be pre-adapted manually in order to ensure that the geometric error is small enough in order to ensure that the error estimator is reliable, clearly compromising the robustness of the whole adaptivity process.

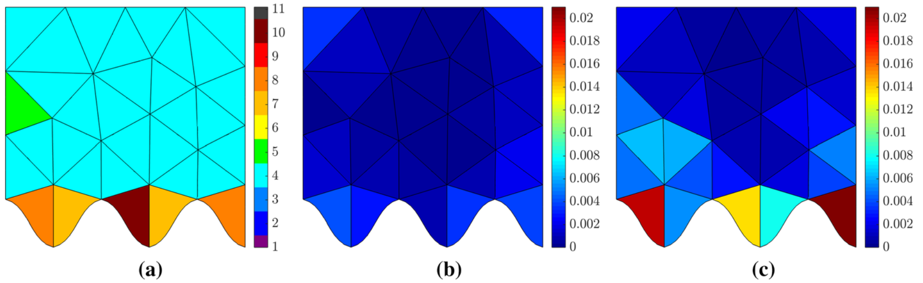


Fig. 11 Sixth iteration of the degree adaptivity procedure with no geometric update using HDG isoparametric elements and $q = 4$. **a** Degree. **b** Estimated error. **c** Exact error

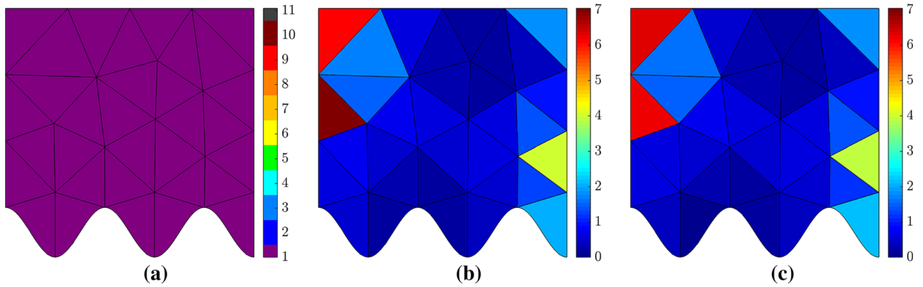


Fig. 12 First iteration of the degree adaptivity procedure with geometric update using HDG NEFEM elements. **a** Degree. **b** Estimated error. **c** Exact error

6.2 NEFEM HDG

The strategy proposed in this work consists of utilising NEFEM, where the geometry is always given by its CAD boundary representation, irrespective of the degree of the functional approximation. In the context of a degree adaptive process, this means that no communication the CAD model is required as the exact boundary representation is already used by the NEFEM solver.

The process starts with a degree of approximation $k = 1$ in all elements. At each iteration the degree of the functional approximation is adapted according to the strategy presented in Sect. 4 and a new nodal distribution is generated for each curved element.

Figure 12 shows the original mesh, the estimated error and the exact error, computed by using the known analytical solution. It is worth emphasising that, even when the degree of the functional approximation used is linear ($k = 1$) the exact boundary representation is considered, as shown in Fig. 12a. The results show a very similar distribution for the estimated and exact errors.

In this case, after only three iterations of the degree adaptive process convergence is achieved. The degree of approximation used in each element, the estimated and the exact errors in each element are represented in Fig. 13.

Remark 2 As discussed in Sect. 4.1, an alternative, not employed in practice due to the high cost induced by the re-generation of the mesh at each iteration of the adaptive process, consists of changing both the degree of approximation for the solution and for the geometry during the adaptivity process, as illustrated in Fig. 3.

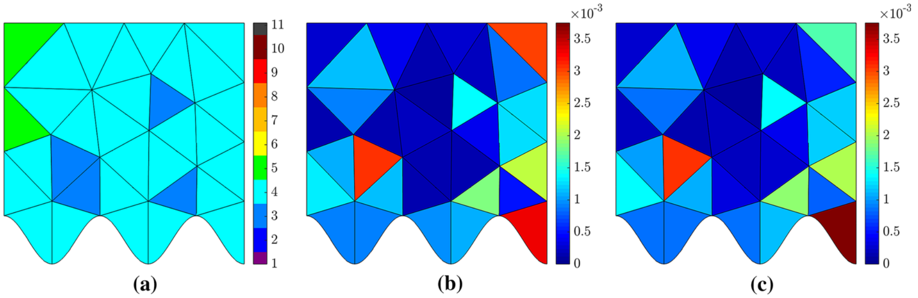


Fig. 13 Third iteration of the degree adaptivity procedure with geometric update using HDG NEFEM elements. **a** Degree. **b** Estimated error. **c** Exact error

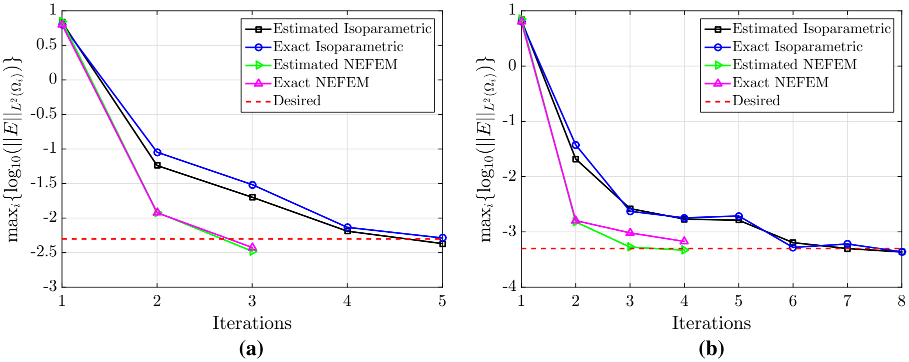


Fig. 14 Estimated and exact errors for isoparametric and NEFEM. **a** Desired error 0.5×10^{-2}. **b** Desired error 0.5×10^{-3}

To illustrate the superiority of NEFEM, Fig. 14 shows the evolution of the maximum estimated error in each element and the maximum exact error in each element for isoparametric and NEFEM approaches and for two magnitudes of the desired error. Figure 10a corresponds to the case previously illustrated, where the desired error is 0.5×10^{-2}, whereas Fig. 10b shows the same study but with a desired error of 0.5×10^{-3}. In both cases the superiority of NEFEM is clear as the desired error is achieved by substantially reducing the computational cost (i.e. the number of iterations of the degree adaptive process and the number of degrees of freedom required to achieve the required accuracy).

In addition, it is worth emphasising that the isoparametric approach requires communication with the CAD model in each iteration to re-generated the high-order nodal distribution. These nodal distributions in curved elements must be specifically designed to ensure optimal convergence of the isoparametric approach [3,5], while for NEFEM the Cartesian approximation of the solution ensures that the accuracy of the approximation is much less sensitive to the quality of the nodal distribution.

7 Numerical Examples

This section presents four numerical examples to illustrate the potential of NEFEM when combined with HDG to perform a degree adaptive process. The examples involve geometries

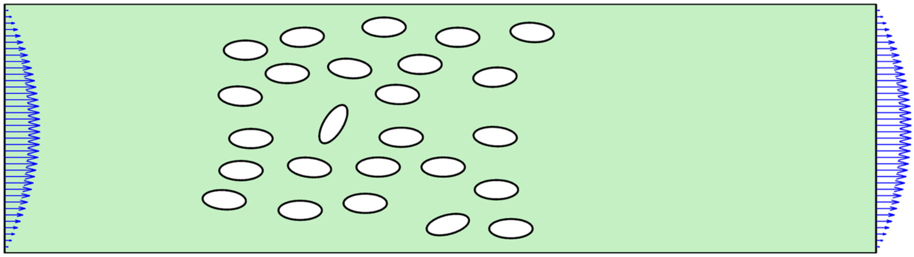


Fig. 15 Computational domain and boundary conditions for the problem involving a flow in a channel with randomly distributed ellipses

with curved boundaries and where coarse meshes are considered to show the robustness of the proposed methodology. In all the examples the high-order isoparametric and NEFEM meshes are generated using the techniques described in [48,64] and [57] respectively.

7.1 Flow in a Channel with Randomly Distributed Ellipses

The first example, similar to a test case presented in [36], considers the flow around a set of randomly distributed set of 25 ellipses in a channel. Dirichlet boundary conditions are considered in the whole domain corresponding to a parabolic velocity profile on the left (inflow) and right (outflow) boundaries and zero (no-slip) Dirichlet boundary condition on the top and bottom walls and on the boundary of the ellipses, as illustrated in Fig. 15.

A coarse mesh with 2,443 triangular elements is first considered. As no analytical solution is available, a reference solution is computed in a much finer mesh with 28,150 elements and by employing a degree of approximation $k = 4$. This reference solution is used to measure the accuracy of the adaptive computations performed in much coarse meshes.

An adaptive process is performed using a quadratic approximation of the curved boundaries and standard isoparametric elements with a desired error of 0.5×10^{-3}. Figure 16a shows the computational mesh and degree of approximation after five iterations. In the vicinity of the ellipses the majority of elements have a cubic degree of approximation where the elements in contact with the ellipses need a higher order of approximation to capture all the flow features. The highest order of approximation is $k = 6$, used, as expected, with the regions with higher curvature of the boundary. Figure 16b, c show the estimated and the reference errors after the adaptive process converged. The discrepancy between the estimated and the reference errors is clearly observed. Despite the adaptive process converges, meaning that all elements have an estimated error below the desired error, a total of 408 elements have a reference error above the desired tolerance.

When the same computation is performed by considering a cubic approximation of the geometry (not reported for brevity), the adaptive process converges again in five iterations. he highest order of approximation used in a few elements is now $k = 7$, indicating that a different geometric representation leads to a different degree of approximation required to achieve convergence. In addition, the error estimator is again not reliable as there are 15 elements where the reference error is above the desired error.

It is apparent that an adaptive computation with isoparametric elements requires an initial pre-adaptation of the mesh and the degree of approximation used to approximate the geometry in order to obtain a reliable error estimator. Next, the finer mesh with 4,048 triangular elements depicted in Fig. 17a is considered. When a quadratic approximation of the geometry is considered, the adaptive process converges in four iterations but the results are still not

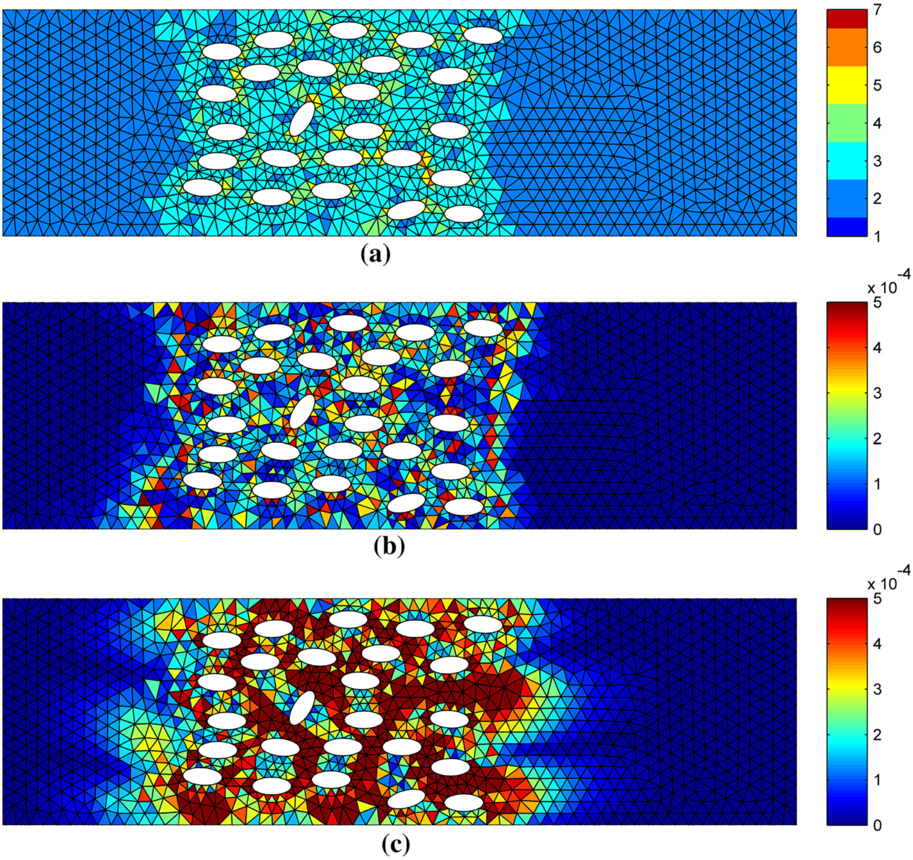
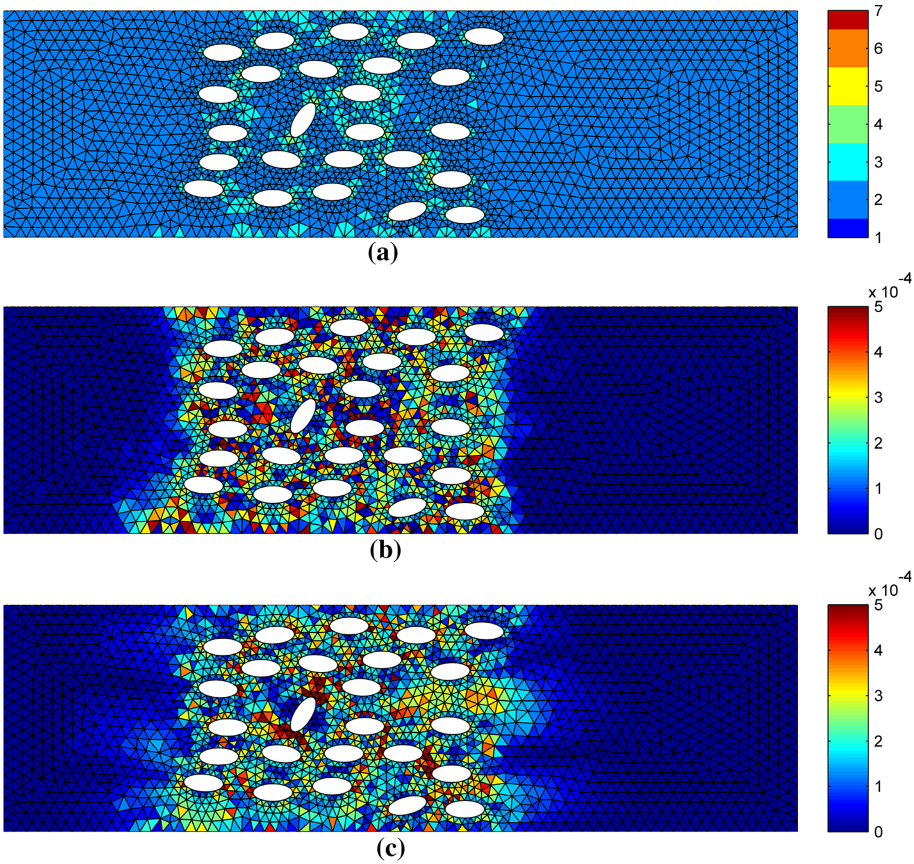


Fig. 16 Final degree distribution, estimated and reference errors for an adaptive computation with isoparametric HDG and quadratic approximation of the curved boundaries in a coarse mesh with 2443 elements. **a** Degree. **b** Estimated error. **c** Exact error

satisfactory as there are 155 elements where the estimated error, shown in Fig. 17b, is above the desired error, represented in Fig. 17c.

Finally, if a cubic approximation of the geometry is employed the adaptive process converges in only three iterations with one element still showing an error above the desired tolerance. The results clearly indicate that in the presence of curved boundaries the level of pre-adaptation required negates all the advantages of an an automatic adaptive process as the initial mesh has to be designed in such a way so that the error in the first computation is near the desired error.

To show the potential of NEFEM in this scenario, an adaptive process is performed employing the coarse mesh with 2,443 triangular elements and starting with a degree of approximation $k = 1$. The adaptive process converges in four iterations. The final degree of approximation used in each element is shown in Fig. 18a, with two elements having the maximum degree of approximation, $k = 6$, required to achieve the desired error. The estimated and reference errors, depicted in Fig. 18b, c, respectively, shows a consistent behaviour that illustrates the reliability of the proposed strategy to estimate the error due to the use of the exact boundary representation. It is worth noting that in the majority of the elements sur-


(a)

(b)

(c)

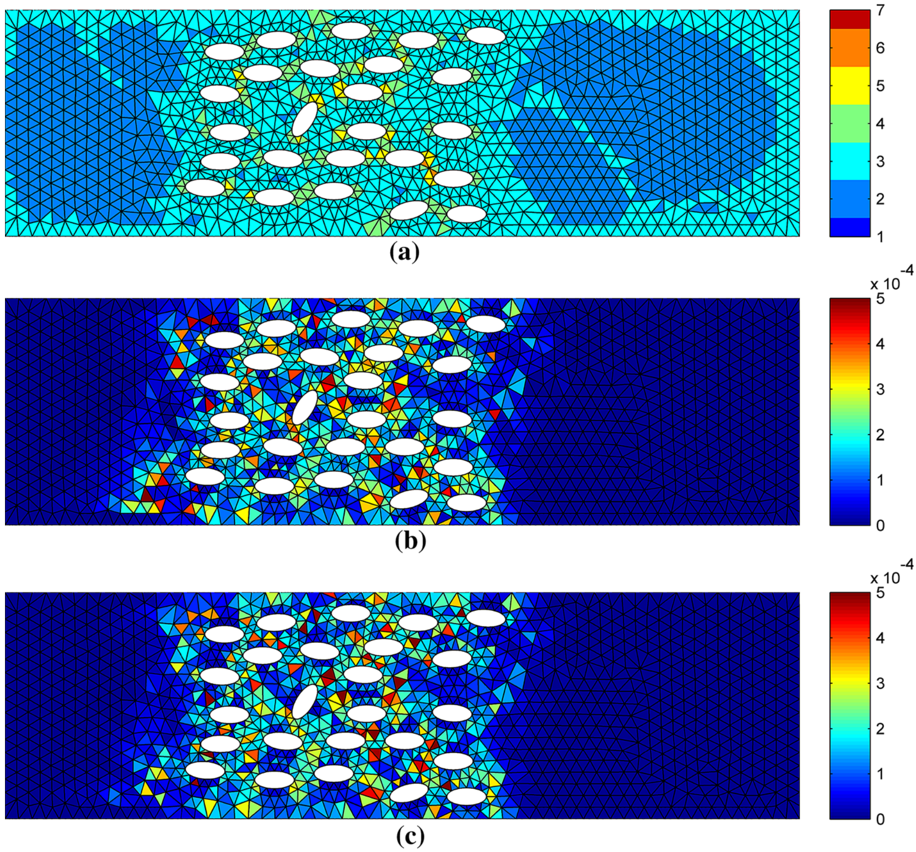
Fig. 17 Final degree distribution, estimated and reference errors for an adaptive computation with isoparametric HDG and quadratic approximation of the curved boundaries in a finer mesh with 4048 elements. **a** Degree. **b** Estimated error. **c** Exact error

rounding the ellipses a degree of approximation $k = 3$ is enough to obtain the required error, illustrating why cubic isoparametric elements outperformed the use of quadratic elements in the previous computations.

The velocity field computed with NEFEM on the mesh shown in Fig. 18a is depicted in Fig. 19.

To better analyse the effect of an accurate boundary representation in a degree adaptive procedure, Fig. 20 shows the evolution of the estimated and exact errors as a function of the number of iterations of the adaptive process by using the coarse mesh with 2,443 elements. For a desired error of 0.5×10^{-3} a quadratic approximation of the geometry prevents convergence of the exact error whereas better results are obtained with a cubic representation of the geometry. It is worth noting that in both cases the exact error stagnates, indicating that the geometric error dominates over the interpolation error. Even when a cubic representation of the geometry is considered, the exact error is not decreasing after the third iteration.

To further illustrate the limitations of an isoparametric formulation during a degree adaptivity procedure, the same analysis is repeated with a lower desired error, namely 0.5×10^{-4}.



(a)

(b)

(c)

Fig. 18 Final degree distribution, estimated and reference errors for an adaptive computation with HDG-NEFEM in a coarse mesh with 2443 elements. **a** Degree. **b** Estimated error. **c** Exact error

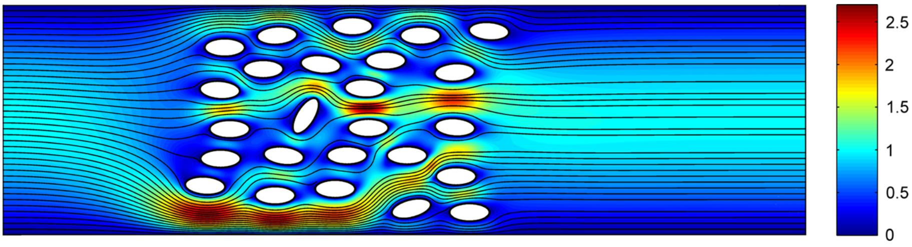


Fig. 19 Magnitude of velocity and isolines computed with HDG-NEFEM on the mesh shown in Fig. 18a after four iterations of the degree adaptive process

Figure 21 shows the evolution of the estimated and exact errors as a function of the number of iterations of the adaptive process by using the coarse mesh with 2,443 elements.

The results show that a cubic approximation of the geometry is not enough because the difference between the estimated and exact error in the final computation with cubic elements is almost an order of magnitude. This suggests, once more that the initial mesh has to be pre-adapted to achieve a reliable degree adaptive process.

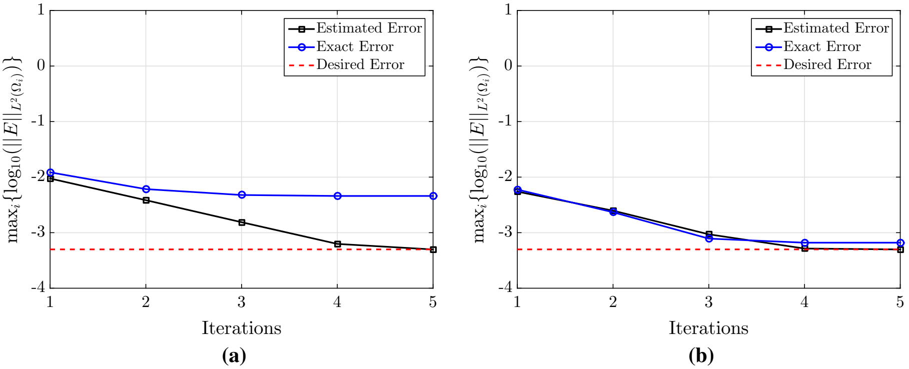


Fig. 20 Evolution of the estimated and exact errors during the degree adaptivity process for $q = 2$ and $q = 3$ with isoparametric HDG in the coarse mesh with 2443 elements and a desired error of 0.5×10^{-3}. **a** $q = 2$. **b** $q = 3$

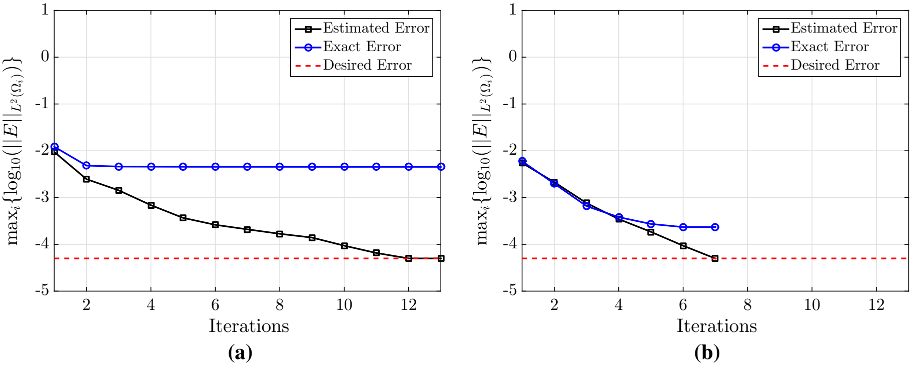


Fig. 21 Evolution of the estimated and exact errors during the degree adaptivity process for $q = 2$ and $q = 3$ with isoparametric HDG in the coarse mesh with 2443 elements and a desired error of 0.5×10^{-4}. **a** $q = 2$. **b** $q = 3$

Finally, Fig. 22 shows the results obtained with NEFEM using the coarse mesh with 2,443 elements and starting with a linear approximation of the solution $k = 1$. The robustness of the proposed approach is clearly illustrated as convergence of both the estimated and exact errors is achieved in the coarse mesh even for a desired error of 0.5×10^{-4}. It is worth emphasising that with NEFEM the adaptive process provides a reliable error estimator even when the desired error is several orders of magnitude lower than the error of the computation in the first mesh. The results clearly indicate that no pre-adaptation of the mesh is required with NEFEM as the geometry is exactly represented irrespectively of the spatial discretisation. Therefore, the adaptive process is purely driven by the functional approximation and not by the geometric error as it happens with an isoparametric formulation.

Further numerical experiments, not reported here, indicate that NEFEM is also superior to an isoparametric approach where the mesh is re-generated at each iteration of the adaptive procedure by using the same degree of the approximation for both the geometry and the solution. In all cases, not only the time required by NEFEM is lower (due to the extra time required to communicate with the CAD model and the mesh generator) but also due

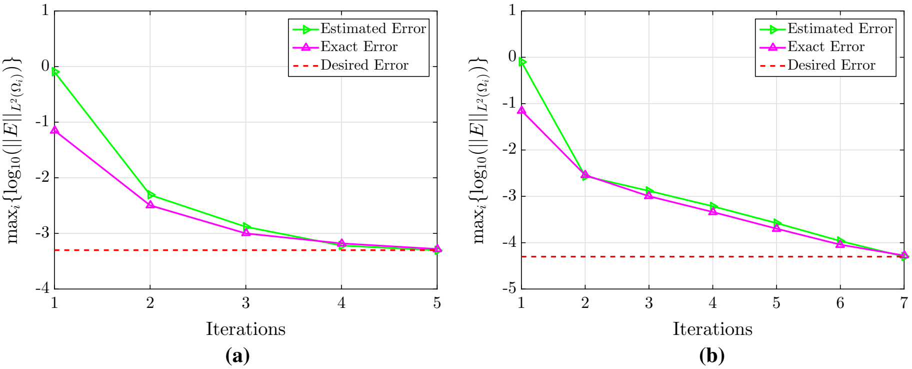


Fig. 22 Evolution of the estimated and exact errors during the degree adaptivity process for NEFEM HDG in the coarse mesh with 2443 elements. **a** Desired error 0.5 × 10^{-3}. **b** Desired error 0.5 × 10^{-4}

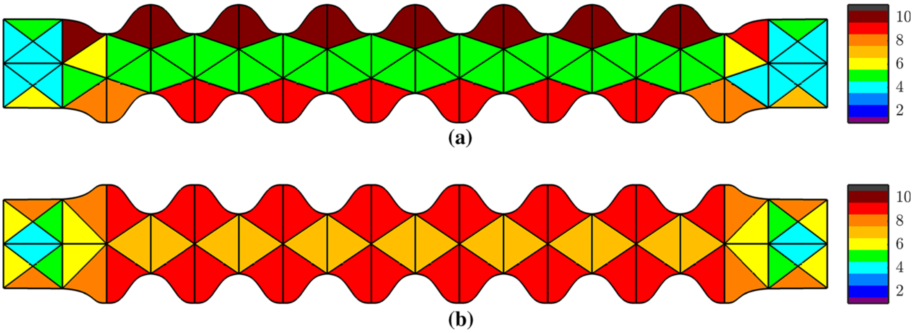


Fig. 23 Mesh and degree of approximation of the converged degree adaptive procedure with HDG-NEFEM for the computation of the flow in a channel with oscillations of the top and bottom boundaries. **a** On phase oscillations. **b** Out of phase oscillations

to the fact that more iterations of the adaptive process are required with an isoparametric formulation.

7.2 Flow in a Channel with Wavy Boundaries

The next example considers the flow in a channel with oscillatory boundaries. This problem, of interest to the micro and nano-fluidics community, is often considered to study the flow structure induced by the different phase of the oscillations of the top and bottom boundaries [32,62]. The two extreme cases are considered here, where the oscillations are exactly on phase and completely out of phase. Figure 23a shows the coarse computational mesh employed with HDG-NEFEM and the final degree of approximation obtained after ten iterations of the degree adaptive process for the case where the boundary oscillations are on phase. Similarly, Fig. 23b shows the mesh and degree of approximation obtained after eight iterations of the degree adaptive process for the case where the boundary oscillations are completely out of phase.

The velocity fields obtained for both cases are represented in Fig. 24 showing the ability of the proposed approach to capture the different flow structure induced by the oscillatory boundaries.

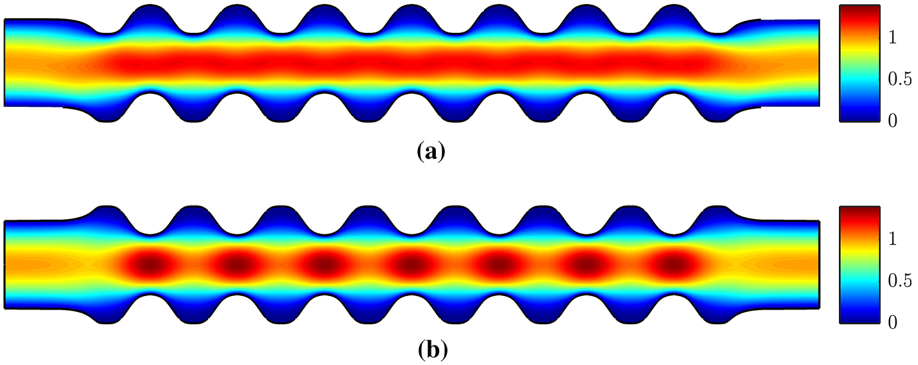


Fig. 24 Magnitude of velocity computed with HDG-NEFEM after convergence of the degree adaptive procedure on the meshes shown in Fig. 23. **a** On phase oscillations. **b** Out of phase oscillations

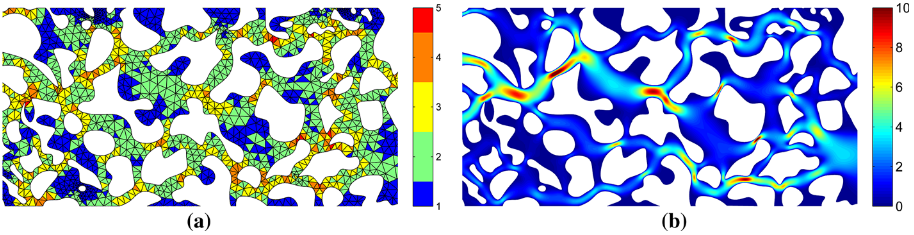


Fig. 25 Mesh and degree of approximation of the converged degree adaptive procedure with HDG-NEFEM and velocity field. **a** Degree. **b** Velocity

7.3 Flow in a Porous Media

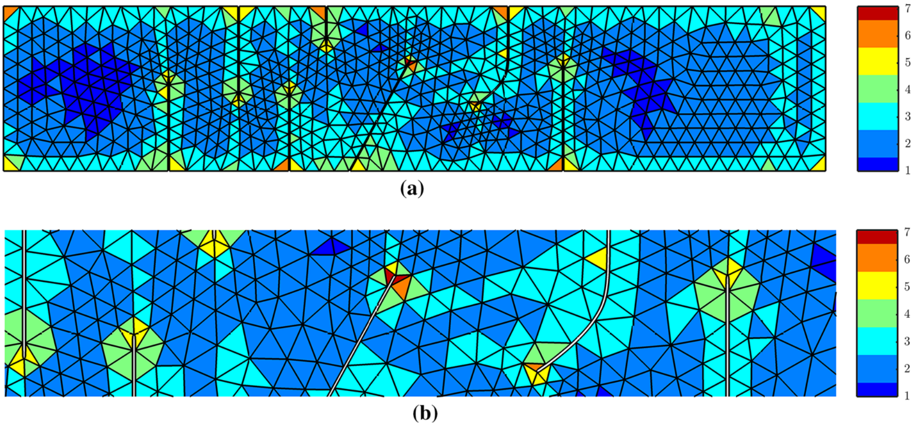
The next example, taken from [59], considers the flow in the interstices of a porous media. The geometry consists of the surroundings of a large number of particles in the porous media. Figure 25 shows the mesh and degree of approximation after eight iterations of the degree adaptivity procedure.

It is worth noting that a linear degree of approximation is used in many elements in contact with curved boundaries. This shows that the proposed adaptivity strategy is completely driven by the complexity of the solution and not by the complexity of the geometry.

7.4 Flow in a Channel with Thin Obstacles

The last example shows a further benefit of using NEFEM by demonstrating its unique ability to obtain accurate solutions with ultra-coarse meshes even when geometric features, smaller than the element size, are present in the boundary representation of the computational domain.

The flow in a channel with a number of thin obstacles is considered. The thickness of the obstacles is approximately 0.08 whereas the minimum element size of the mesh that has been generated, using the technique proposed in [57], is 0.32. The degree adaptive process is started, as in previous examples, with a linear approximation of the solution and convergence is achieved in four iterations. The final degree of approximation in each element is represented in Fig. 26a. Figure 26 shows a detailed view of a region in the channel showing the elements near the end of some of the obstacles. This plot shows not only that the element



(a)

(b)

Fig. 26 Mesh and degree of approximation of the converged degree adaptive procedure with HDG-NEFEM for the computation of the flow in a channel with high curvature. **a** Whole domain. **b** Zoom

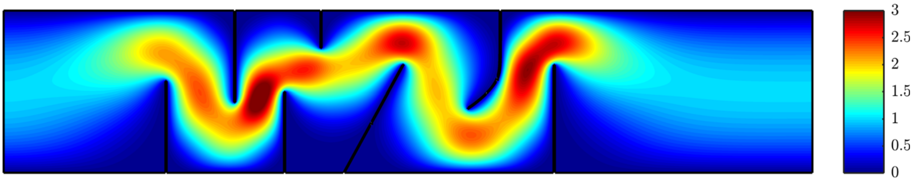


Fig. 27 Velocity computed with HDG-NEFEM after convergence of the degree adaptive procedure in the mesh of Fig. 26

size is independent on the geometric complexity but it also demonstrates the robustness of the proposed degree adaptive technique. The adaptivity process is clearly driven only by the complexity of the solution as a different degree of approximation is employed in elements with almost identical geometric complexity due to the different complexity of the solution.

The velocity field, obtained on the mesh shown in Fig. 26, is represented in Fig. 27.

8 Concluding Remarks

A new degree adaptive methodology that combines the advantages of the HDG formulation and NEFEM has been presented. The proposed method results in a cheap and reliable error estimator due to the cheap computation of a post-processed solution provided by HDG and the ability to exactly represent curved boundaries irrespectively of the polynomial degree used for the functional approximation that is characteristic of NEFEM.

The proposed approach is compared against two alternative options to perform degree adaptivity. The first approach, broadly used in practice, consists of keeping the shape of the curved elements during the degree adaptivity process. It is found that this approach leads to an unreliable error estimator. The numerical examples show that even when the estimated error is below the required tolerance, the exact error can be orders of magnitude higher. The second approach, not used in practice, consists of changing the shape of the curved elements during the adaptive process. The main drawback is its high cost due to the need to constantly communicate with the CAD model and re-generate the nodal distributions for curved elements.

The proposed approach considers, for the first time, the implementation of the NEFEM rationale in an HDG framework. A number of numerical examples have been presented to compare the performance of the proposed methodology and to shows its superiority on a number of problems involving domains with curved boundaries.

Acknowledgements The authors gratefully acknowledge the financial support of the Ministerio de Economía y Competitividad (Grant Number DPI 2014-51844-C2-2-R). The first author also gratefully acknowledges the financial support provided by the Sêr Cymru National Research Network for Advanced Engineering and Materials (Grant Number NRN045). The second author is also grateful for the financial support provided by the Generalitat de Catalunya (Grant Number 2014-SGR-1471).

References

1. Bangerth, W., Rannacher, R.: Adaptive Finite Element Methods for Differential Equations. Birkhäuser, Boston (2013)
2. Bassi, F., Rebay, S.: High-order accurate discontinuous finite element solution of the 2D Euler equations. J. Comput. Phys. **138**(2), 251–285 (1997)
3. Bernardi, C.: Optimal finite-element interpolation on curved domains. SIAM J. Numer. Anal. **26**(5), 1212–1240 (1989)
4. Burbeau, A., Sagaut, P.: A dynamic p-adaptive discontinuous Galerkin method for viscous flow with shocks. Comput. Fluids **34**(4), 401–417 (2005)
5. Chen, Q., Babuška, I.: Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. Comput. Methods Appl. Mech. Eng. **128**(3–4), 405–417 (1995)
6. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. SIAM, Philadelphia (2002)
7. Cirak, F., Ortiz, M., Schroder, P.: Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. Int. J. Numer. Methods Eng. **47**(12), 2039–2072 (2000)
8. Cockburn, B., Dong, B., Guzmán, J.: A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. Math. Comput. **77**(264), 1887–1916 (2008)
9. Cockburn, B., Dong, B., Guzmán, J., Restelli, M., Sacco, R.: A hybridizable discontinuous Galerkin method for steady-state convection–diffusion–reaction problems. SIAM J. Sci. Comput. **31**(5), 3827–3846 (2009)
10. Cockburn, B., Gopalakrishnan, J.: The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. SIAM J. Numer. Anal. **47**(2), 1092–1125 (2009)
11. Cockburn, B., Gopalakrishnan, J., Lazarov, R.: Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM J. Numer. Anal. **47**(2), 1319–1365 (2009)
12. Cockburn, B., Shi, K.: Devising HDG methods for Stokes flow: an overview. Comput. Fluids **98**, 221–229 (2014)
13. Cockburn, B., Shu, C.W.: Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16**(3), 173–261 (2001)
14. Cohen, G.C., Liu, Q.H.: Higher-order numerical methods for transient wave equations. J. Acoust. Soc. Am. **114**(1), 21–21 (2003)
15. De Basabe, J.D., Sen, M.K., Wheeler, M.F.: The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion. Geophys. J. Int. **175**(1), 83–93 (2008)
16. Demkowicz, L.: Fully automatic hp-adaptivity for Maxwell's equations. Comput. Methods Appl. Mech. Eng. **194**(2), 605–624 (2005)
17. Dey, S., Shephard, M.S., Flaherty, J.E.: Geometry representation issues associated with p-version finite element computations. Comput. Methods Appl. Mech. Eng. **150**(1–4), 39–55 (1997)
18. Díez, P., Huerta, A.: A unified approach to remeshing strategies for finite element h-adaptivity. Comput. Methods Appl. Mech. Eng. **176**(1–4), 215–229 (1999)
19. Díez, P., Ródenas, J.J., Zienkiewicz, O.C.: Equilibrated patch recovery error estimates: simple and accurate upper bounds of the error. Int. J. Numer. Methods Eng. **69**(10), 2075–2098 (2007)

20. Donea, J., Huerta, A.: Finite Element Methods for Flow Problems. Wiley, Chichester (2003)
21. Dumbser, M., Käser, M.: An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-II. The three-dimensional isotropic case. Geophys. J. Int. **167**(1), 319–336 (2006)
22. Dumbser, M., Käser, M., Toro, E.F.: An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes—V. Local time stepping and p-adaptivity. Geophys. J. Int. **171**(2), 695–717 (2007)
23. Giorgiani, G., Fernández-Méndez, S., Huerta, A.: Hybridizable discontinuous Galerkin p-adaptivity for wave propagation problems. Int. J. Numer. Methods Fluids **72**(12), 1244–1262 (2013)
24. Giorgiani, G., Fernández-Méndez, S., Huerta, A.: Hybridizable discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations. Comput. Fluids **98**, 196–208 (2014)
25. Hesthaven, J.S.: High-order accurate methods in time-domain computational electromagnetics: a review. Adv. Imaging Electron Phys. **127**, 59–123 (2003)
26. Hesthaven, J.S., Warburton, T.: Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell's equations. J. Comput. Phys. **181**(1), 186–221 (2002)
27. Huerta, A., Angeloski, A., Roca, X., Peraire, J.: Efficiency of high-order elements for continuous and discontinuous Galerkin methods. Int. J. Numer. Methods Eng. **96**(9), 529–560 (2013)
28. Huerta, A., Díez, P.: Error estimation including pollution assessment for nonlinear finite element analysis. Comput. Methods Appl. Mech. Eng. **181**(1), 21–41 (2000)
29. Huerta, A., Rodríguez-Ferran, A., Díez, P., Sarrate, J.: Adaptive finite element strategies based on error assessment. Int. J. Numer. Methods Eng. **46**(10), 1803–1818 (1999)
30. Johnson, C., Hansbo, P.: Adaptive finite element methods in computational mechanics. Comput. Methods Appl. Mech. Eng. **101**(1–3), 143–181 (1992)
31. Karban, P., Mach, F., Doležel, I.: Advanced adaptive algorithms in 2D finite element method of higher order of accuracy. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **32**(3), 834–849 (2013)
32. Khuri, S.: Stokes flow in curved channels. J. Comput. Appl. Math. **187**(2), 171–191 (2006)
33. Kirby, R., Sherwin, S.J., Cockburn, B.: To CG or to HDG: a comparative study. J. Sci. Comput. **51**(1), 183–212 (2011)
34. Krivodonova, L., Berger, M.: High-order accurate implementation of solid wall boundary conditions in curved geometries. J. Comput. Phys. **211**(2), 492–512 (2006)
35. Li, L., Lanteri, S., Perrussel, R.: Numerical investigation of a high order hybridizable discontinuous Galerkin method for 2D time-harmonic Maxwell's equations. COMPEL Int. J. Comput. Math. Electr. Electron. Eng. **32**(3), 1112–1138 (2013)
36. Liu, Y.: A new fast multipole boundary element method for solving 2-D Stokes flow problems based on a dual BIE formulation. Eng. Anal. Bound. Elem. **32**(2), 139–151 (2008)
37. Montlaur, A., Fernández-Méndez, S., Huerta, A.: Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. Int. J. Numer. Methods Fluids **57**(9), 1071–1092 (2008)
38. Nguyen, N., Peraire, J., Cockburn, B.: A hybridizable discontinuous Galerkin method for Stokes flow. Comput. Methods Appl. Mech. Eng. **199**(9), 582–597 (2010)
39. Nguyen, N., Peraire, J., Cockburn, B.: A hybridizable discontinuous Galerkin method for Stokes flow. Comput. Methods Appl. Mech. Eng. **199**(9–12), 582–597 (2010)
40. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for linear convection–diffusion equations. J. Comput. Phys. **228**(9), 3232–3254 (2009)
41. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection–diffusion equations. J. Comput. Phys. **228**(23), 8841–8855 (2009)
42. Nguyen, N.C., Peraire, J., Cockburn, B.: An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations. J. Comput. Phys. **230**(4), 1147–1170 (2011)
43. Nguyen, N.C., Persson, P.O., Peraire, J.: RANS solutions using high order discontinuous Galerkin methods. AIAA Pap. **914**, 2007 (2007)
44. Oden, J.T., Demkowicz, L., Rachowicz, W., Westermann, T.: Toward a universal hp adaptive finite element strategy, Part 2. A posteriori error estimation. Comput. Methods Appl. Mech. Eng. **77**(1–2), 113–180 (1989)
45. Pardo, D.: Multigoal-oriented adaptivity for hp-finite element methods. Procedia Comput. Sci. **1**(1), 1953–1961 (2010)
46. Parés, N., Díez, P., Huerta, A.: Subdomain-based flux-free a posteriori error estimators. Comput. Methods Appl. Mech. Eng. **195**(4), 297–323 (2006)
47. Parés, N., Díez, P., Huerta, A.: Exact bounds for linear outputs of the advection–diffusion–reaction equation using flux-free error estimates. SIAM J. Sci. Comput. **31**(4), 3064–3089 (2009)
48. Poya, R., Sevilla, R., Gil, A.J.: A unified approach for a posteriori high-order curved mesh generation using solid mechanics. Comput. Mech. **58**(3), 457–490 (2016)

49. Raviart, P.A., Thomas, J.M., Ciarlet, P.G., Lions, J.L.: Introduction à l'analyse numérique des équations aux dérivées partielles, vol. 2. Dunod, Paris (1998)
50. Scholle, M.: Creeping Couette flow over an undulated plate. Arch. Appl. Mech. **73**(11–12), 823–840 (2004)
51. Sevilla, R., Fernández-Méndez, S.: Numerical integration over 2D NURBS-shaped domains with applications to NURBS-enhanced FEM. Finite Elem. Anal. Des. **47**(10), 1209–1220 (2011)
52. Sevilla, R., Fernández-Méndez, S., Huerta, A.: 3D NURBS-enhanced finite element method (NEFEM). Int. J. Numer. Methods Eng. **88**(2), 103–125 (2011)
53. Sevilla, R., Fernández-Méndez, S., Huerta, A.: Comparison of high-order curved finite elements. Int. J. Numer. Methods Eng. **87**(8), 719–734 (2011)
54. Sevilla, R., Fernández-Méndez, S., Huerta, A.: NURBS-enhanced finite element method (NEFEM). Arch. Comput. Methods Eng. **18**(4), 441–484 (2011)
55. Sevilla, R., Hassan, O., Morgan, K.: An analysis of the performance of a high-order stabilised finite element method for simulating compressible flows. Comput. Methods Appl. Mech. Eng. **253**, 15–27 (2013)
56. Sevilla, R., Huerta, A.: Tutorial on hybridizable discontinuous galerkin (HDG) for second-order elliptic problems. In: Schröder, J., Wriggers, P. (eds.) Advanced Finite Element Technologies, pp. 105–129. Springer, Berlin (2016)
57. Sevilla, R., Rees, L., Hassan, O.: The generation of triangular meshes for NURBS-enhanced FEM. Int. J. Numer. Methods Eng. **108**(8), 941–968 (2016)
58. Shu, C.W.: Discontinuous Galerkin methods for time-dependent convection dominated problems: basics, recent developments and comparison with other methods. In: Barrenechea, G.R., Brezzi, F., Cangiani, A., Georgoulis, E.H. (eds.) Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations, pp. 369–397. Springer, Berlin (2016)
59. Sirivithayapakorn, S., Keller, A.: Transport of colloids in saturated porous media: a pore-scale observation of the size exclusion effect and colloid acceleration. Water Resour. Res. **39**(4), WR001,583 (2003)
60. Soghrati, S., Merel, R.A.: NURBS enhanced HIFEM: a fully mesh-independent method with zero geometric discretization error. Finite Elem. Anal. Des. **120**, 68–79 (2016)
61. Šolín, P., Červenỳ, J., Doležel, I.: Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM. Math. Comput. Simul. **77**(1), 117–132 (2008)
62. Wang, C.: On Stokes slip flow through a transversely wavy channel. Mech. Res. Commun. **38**(3), 249–254 (2011)
63. Wang, Z.J., Fidkowski, K., Abgrall, R., Bassi, F., Caraeni, D., Cary, A., Deconinck, H., Hartmann, R., Hillewaert, K., Huynh, H.T., et al.: High-order CFD methods: current status and perspective. Int. J. Numer. Methods Fluids **72**(8), 811–845 (2013)
64. Xie, Z.Q., Sevilla, R., Hassan, O., Morgan, K.: The generation of arbitrary order curved meshes for 3D finite element analysis. Comput. Mech. **51**, 361–374 (2013)
65. Zienkiewicz, O., Zhu, J., Gong, N.: Effective and practical h-p-version adaptive analysis procedures for the finite element method. Int. J. Numer. Methods Eng. **28**(4), 879–891 (1989)