Journal of Computational Physics 230 (2011) 1147–1170

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier–Stokes equations

N.C. Nguyen [a,*], J. Peraire [a], B. Cockburn [b]

[a] Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
[b] School of Mathematics, University of Minnesota, Minneapolis, MN 55455, USA

ARTICLE INFO

Article history:
Received 2 January 2010
Received in revised form 20 October 2010
Accepted 25 October 2010
Available online 2 November 2010

Keywords:
Finite element methods
Discontinuous Galerkin methods
Hybrid/mixed methods
Augmented Lagrangian
Incompressible Navier–Stokes equations

ABSTRACT

We present an implicit high-order hybridizable discontinuous Galerkin method for the steady-state and time-dependent incompressible Navier–Stokes equations. The method is devised by using the discontinuous Galerkin discretization for a velocity gradient-pressure–velocity formulation of the incompressible Navier–Stokes equations with a special choice of the numerical traces. The method possesses several unique features which distinguish itself from other discontinuous Galerkin methods. First, it reduces the globally coupled unknowns to the approximate trace of the velocity and the mean of the pressure on element boundaries, thereby leading to a significant reduction in the degrees of freedom. Moreover, if the augmented Lagrangian method is used to solve the linearized system, the globally coupled unknowns become the approximate trace of the velocity only. Second, it provides, for smooth viscous-dominated problems, approximations of the velocity, pressure, and velocity gradient which converge with the optimal order of $k+1$ in the L^2-norm, when polynomials of degree $k \geqslant 0$ are used for all components of the approximate solution. And third, it displays superconvergence properties that allow us to use the above-mentioned optimal convergence properties to define an element-by-element postprocessing scheme to compute a new and better approximate velocity. Indeed, this new approximation is exactly divergence-free, $\boldsymbol{H}(\mathrm{div})$-conforming, and converges with order $k+2$ for $k \geqslant 1$ and with order 1 for $k=0$ in the L^2-norm. Moreover, a novel and systematic way is proposed for imposing boundary conditions for the stress, viscous stress, vorticity and pressure which are not naturally associated with the weak formulation of the method. This can be done on different parts of the boundary and does not result in the degradation of the optimal order of convergence properties of the method. Extensive numerical results are presented to demonstrate the convergence and accuracy properties of the method for a wide range of Reynolds numbers and for various polynomial degrees.

1. Introduction

In recent years, discontinuous Galerkin (DG) methods have emerged as a competitive alternative for numerically solving the incompressible Navier–Stokes equations [3,18,19,32,44]. The advantages of the DG methods over classical continuous Galerkin finite element, finite difference and finite volume methods are well-documented in the literature (see [5,6,21] and references therein): the DG methods work well on arbitrary meshes, result in stable high order discretizations of the convective and diffusive operators, allow for a simple and unambiguous imposition of boundary conditions and are well suited for parallelization and adaptivity. Despite all these advantages, there are still obstacles which prevent DG methods

[*] Corresponding author. Tel.: +1 617 253 8080.
E-mail address: cuongng@mit.edu (N.C. Nguyen).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.10.032

from becoming the method of choice for a wide class of applications. One such obstacle is the high computational cost associated with DG methods which can be traced to the larger number of globally coupled unknowns when compared to continuous Galerkin finite elements, finite differences, or finite volume schemes.

In this paper, we introduce a hybridizable discontinuous Galerkin (HDG) method for the numerical solution of the incompressible Navier–Stokes equations. The HDG methods were first introduced for diffusion [13] and continuum mechanics [26,45] problems. They were analyzed in [7,15,17], see also [16], in the setting of diffusion problems, and then developed for linear and nonlinear convection–diffusion problems [8,37,38,46], Stokes flow [4,11,14,39,40]. The HDG method for the compressible Euler and Navier–Stokes equations is introduced in [43]. An overview of recent development of HDG methods is provided in [41]. The HDG methods retain the advantages of standard DG methods and provide a significantly reduced number of globally-coupled degrees of freedom, thereby allowing for a substantial reduction in the computational cost.

In this paper, we devise the first HDG method for the incompressible Navier–Stokes equations by extending the HDG methods for convection–diffusion [37,38] and the HDG methods for the Stokes system [14,39,40], and showing that the distinctive advantages of those HDG methods are retained, namely:

- **Reduced number of degrees of freedom**. Unlike *all* known other DG methods, which result in a final system involving the degrees of freedom of the approximate velocity and pressure, the HDG method produces a final system involving the degrees of freedom of the *approximate trace* of the velocity and the *mean* of pressure. Since the approximate trace is defined on the element borders only and since the mean of pressure is a piece-constant function, the HDG method has significantly less globally coupled unknowns than other DG methods, especially for high-degree polynomial approximations. Moreover, if the augmented Lagrangian method [23] is used to solve the linearized system, the globally coupled unknowns become the approximate trace of the velocity only. This large reduction in the degrees of freedom leads to significant savings for both computational time and memory storage.
- **Optimal convergence**. The HDG method provides an approximate velocity, pressure and velocity gradient converging with the optimal order $k + 1$ in the L^2-norm for viscous-dominated flows with smooth solution; here k is the degree of the polynomials used to represent all components of the approximate solution. This has to be contrasted with the fact that *all* known DG methods display the suboptimal order of convergence of k for the approximate pressure and for the velocity gradient or the vorticity. This includes, the first DG method for the Navier–Stokes equations [28,29], as well as the family of DG methods for the Navier–Stokes equations proposed in [18,19].
- **Superconvergence and local postprocessing**. The HDG method has superconvergence properties for the velocity which, combined with the above-mentioned optimal converge properties, allows us to use an element-by-element postprocessing, proposed in [14] for HDG methods for Stokes flow, to obtain a new and better approximation of the velocity. Unlike the original velocity, the postprocessed velocity is exactly divergence-free, $\boldsymbol{H}(\text{div})$-conforming, and converges with order $k + 2$ for $k \geqslant 1$. Since the postprocessing is performed at the element level, the computational cost involved in obtaining the postprocessed velocity is very small.
- **Unified treatment of boundary conditions and the numerical fluxes.** The HDG method entails a single numerical flux formulae containing both the viscous and inviscid numerical fluxes. Different boundary conditions can be included in a single framework by defining appropriate numerical fluxes on the boundaries of the physical domain. This *novel and systematic* manner of imposing boundary conditions allows for pressure, vorticity and stress boundary conditions to be prescribed on different parts of the boundary.

Let us briefly emphasize the fact that, to the knowledge of the authors, no other known DG or mixed method for the Navier–Stokes equations has all the above four properties of these HDG methods. Note that, as we pointed out in [40], some DG methods provide velocities that are divergence-free inside each of the element which, however, do not lie on $\boldsymbol{H}(\text{div})$ since their normal component has no interelement continuity. Examples are the first DG method proposed for the Stokes system [2] and for the Navier–Stokes equations in [28] and, more recently, and the DG method for the Stokes equations proposed in [33] and in [34] for the Navier–Stokes equations. Note also that there are DG methods that do provide velocities that are divergence-free and belong to $\boldsymbol{H}(\text{div})$. A wide family DG methods with this property were introduced in [18] for the Navier–Stokes equations, even though only a particular case was treated in detail therein. Other particular cases were developed later in [19] and (for the Stokes problem) in [47,49]; the latter method was then extended to the Navier–Stokes equation in [48]. Finally, note that the first DG method whose formulation involves $\boldsymbol{H}(\text{div})$-conforming velocities with are also divergence-free was proposed in [4] for the Stokes equations; see also [9,10] for an extension of this approach to a mixed method. However, their velocities converge with order at most $k + 1$ for $k \geqslant 1$.

Recently, there have been new developments in DG methods—the multiscale discontinuous Galerkin (MDG) method [27] and the embedded discontinuous Galerkin method (EDG) [13,26]—which aim to reduce the globally coupled degrees in a DG discretization. Like the HDG method, these DG methods solve for the approximate trace of the field variables. However, unlike the HDG method, the approximate trace in these methods resides in a C^0 space. Therefore, neither the MDG nor the EDG method have the local conservativity property of the HDG method. As a consequence, see [16], these DG methods do not have some superconvergence properties of the HDG method and hence their approximate solution can not be postprocessed to yield a higher-order convergent approximation. Further development of the MDG method for the incompressible Navier–Stokes equations leads to the so-called Galerkin interface stabilisation (GIS) method [31]. As expected, the GIS method provides suboptimal convergence for the approximate pressure and stress even for the Stokes system.

The paper is organized as follows. In Section 2 we define the notation used throughout the paper. In Section 3 we introduce the HDG method for numerically solving the steady incompressible Navier–Stokes equations and present the element-by-element postprocessing to compute a new approximation of the velocity proposed in [14]. In this section, we also extend the method to treat different boundary conditions involving derivatives of the velocity and to solve the time-dependent incompressible Navier–Stokes equations. In Section 4 we describe two different strategies to implement the HDG method. In Section 5 we provide numerical results to assess the convergence and accuracy of the method. Finally, in Section 6 we present some concluding remarks.

2. Governing equations and notation

2.1. The incompressible Navier–Stokes equations

We consider the steady incompressible Navier–Stokes equations

$$\begin{aligned} & -\nu\Delta\boldsymbol{u} + \nabla p + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) = \boldsymbol{f}, \quad \text{in } \Omega, \\ & \nabla\cdot\boldsymbol{u} = 0, \quad \text{in } \Omega, \\ & \boldsymbol{u} = \boldsymbol{g}, \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where $\boldsymbol{u}$ and p are velocity vector and pressure, respectively, and $\boldsymbol{f}$ is a known body force. It is assumed that the prescribed boundary velocity $\boldsymbol{g}$ satisfies the incompressibility constraint $\int_{\partial\Omega}\boldsymbol{g}\cdot\boldsymbol{n} = 0$. The pressure is made unique by requiring that $\int_\Omega p = 0$. Here Ω is a polygonal domain in $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$, and ν is a kinematic viscosity.

We introduce the velocity gradient tensor $\mathbf{L} = \nabla\boldsymbol{u}$ and rewrite the above equation as a first order system of equations as

$$\begin{aligned} & \mathbf{L} - \nabla\boldsymbol{u} = 0, \quad \text{in } \Omega, \\ & -\nu\nabla\cdot\mathbf{L} + \nabla p + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) = \boldsymbol{f}, \quad \text{in } \Omega, \\ & \nabla\cdot\boldsymbol{u} = 0, \quad \text{in } \Omega, \\ & \boldsymbol{u} = \boldsymbol{g}, \quad \text{on } \partial\Omega. \end{aligned} \tag{2}$$

Next, we introduce the notation we are going to use to describe the HDG method for numerically solving this system.

2.2. Mesh and trace operators

We employ the notation used in [14,39]. We denote by $\mathcal{T}_h$ a collection of disjoint regular elements K that partition Ω and set $\partial\mathcal{T}_h := \{\partial K : K \in \mathcal{T}_h\}$. For an element K of the collection $\mathcal{T}_h,\ F = \partial K \cap \partial\Omega$ is the boundary face if the $d-1$ Lebesgue measure of F is nonzero. For two elements K^+ and K^- of the collection $\mathcal{T}_h,\ F = \partial K^+ \cap \partial K^-$ is the interior face between K^+ and K^- if the $d-1$ Lebesgue measure of F is nonzero. We denote by $\mathcal{E}_h^o$ and $\mathcal{E}_h^\partial$ the set of interior and boundary faces, respectively. We set $\mathcal{E}_h = \mathcal{E}_h^o \cup \mathcal{E}_h^\partial$.

Let $\boldsymbol{n}^+$ and $\boldsymbol{n}^-$ be the outward unit normal vectors on two neighboring elements K^+ and K^-, respectively. We use $(\mathbf{G}^\pm, \boldsymbol{v}^\pm, q^\pm)$ to denote the traces of $(\mathbf{G}, \boldsymbol{v}, q)$ on F from the interior of the elements $K^\pm$, where $\mathbf{G}$, $\boldsymbol{v}$, and q are second-order tensorial, vectorial, and scalar functions, respectively. Then, we define the jumps $[\![\cdot]\!]$ as follows. For $F \in \mathcal{E}_h^o$, we set

$$\begin{aligned} & [\![\mathbf{G}\boldsymbol{n}]\!] = \mathbf{G}^+\boldsymbol{n}^+ + \mathbf{G}^-\boldsymbol{n}^-, \\ & [\![\boldsymbol{v}\odot\boldsymbol{n}]\!] = \boldsymbol{v}^+\odot\boldsymbol{n}^+ + \boldsymbol{v}^-\odot\boldsymbol{n}^-, \\ & [\![q\boldsymbol{n}]\!] = q^+\boldsymbol{n}^+ + q^-\boldsymbol{n}^-. \end{aligned}$$

Here $\odot$ is either $\cdot$ or $\otimes$ which denote the usual dot product and tensor product, respectively. For $F \in \mathcal{E}_h^\partial$, the set of boundary edges on which $\mathbf{G}$, $\boldsymbol{v}$ and q are single valued, we set

$$\begin{aligned} & [\![\mathbf{G}\boldsymbol{n}]\!] = \mathbf{G}\boldsymbol{n}, \\ & [\![\boldsymbol{v}\odot\boldsymbol{n}]\!] = \boldsymbol{v}\odot\boldsymbol{n}, \\ & [\![q\boldsymbol{n}]\!] = q\boldsymbol{n}, \end{aligned}$$

where $\boldsymbol{n}$ is the unit outward normal to $\partial\Omega$.

2.3. Approximation spaces

Let $\mathcal{P}_k(D)$ denote the space of polynomials of degree at most k on a domain D and let $L^2(D)$ be the space of square integrable functions on D. We set $\boldsymbol{\mathcal{P}}_k(D) = [\mathcal{P}_k(D)]^d,\ \mathbf{P}_k(D) = [\mathcal{P}_k(D)]^{d\times d},\ \boldsymbol{L}^2(D) = [L^2(D)]^d$, and $\mathbf{L}^2(D) = [L^2(D)]^{d\times d}$. We introduce the following discontinuous finite element approximation spaces for the gradient, velocity, and pressure:

$$\begin{aligned}
\mathbf{G}_h &= \{\mathbf{G} \in \mathbf{L}^2(\mathcal{T}_h) : \mathbf{G}|_K \in \mathbf{P}_k(K),\ \forall K \in \mathcal{T}_h\},\\
\boldsymbol{V}_h &= \{\boldsymbol{v} \in \boldsymbol{L}^2(\mathcal{T}_h) : \boldsymbol{v}|_K \in \boldsymbol{\mathcal{P}}_k(K),\ \forall K \in \mathcal{T}_h\},\\
P_h &= \{q \in L^2(\mathcal{T}_h) : q|_K \in \mathcal{P}_k(K),\ \forall K \in \mathcal{T}_h\}.
\end{aligned}$$

In addition, we introduce a finite element approximation space for the approximate trace of the velocity

$$\boldsymbol{M}_h = \{\boldsymbol{\mu} \in \boldsymbol{L}^2(\mathcal{E}_h) : \boldsymbol{\mu}|_F \in \boldsymbol{\mathcal{P}}_k(F),\ \forall F \in \mathcal{E}_h\}$$

and set $\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu} \in \boldsymbol{M}_h\colon \boldsymbol{\mu} = \mathsf{P}_\partial \boldsymbol{g} \text{ on } \partial\Omega\}$, where P_∂ is the $L^2(\partial\Omega)$ projection into the space $\{\boldsymbol{\mu}|_{\partial\Omega}, \forall\ \boldsymbol{\mu} \in \boldsymbol{M}_h\}$. Note that $\boldsymbol{M}_h$ consists of functions which are continuous inside the faces (or edges) $F \in \mathcal{E}_h$ and discontinuous at their borders. We further denote by $\overline{\Psi}_h$ the set of functions in $L^2(\partial\mathcal{T}_h)$ that are constant on each ∂K for all elements K

$$\overline{\Psi}_h = \{r \in L^2(\partial\mathcal{T}_h) : r \in \mathcal{P}_0(\partial K),\ \forall\ K \in \mathcal{T}_h\}.$$

The mean of our approximate pressure will belong to this space. Here the mean is defined as follows. For a given function q in $L^2(\partial\mathcal{T}_h)$, the mean of q on ∂K is given by $\bar{q}|_{\partial K} = \frac{1}{|\partial K|}\int_{\partial K} q$. Obviously, we have $\bar{q} = q$ for any q in $\overline{\Psi}_h$.

Finally, we define various inner products for our finite element spaces. We write $(w, v)_{\mathcal{T}_h} := \sum_{K\in\mathcal{T}_h}(w, v)_K$, where $(w, v)_D$ denotes the integral of wv over the domain $D \subset \mathbb{R}^d$ for $w,\ v \in P_h$. We also write $(\boldsymbol{w}, \boldsymbol{v})_{\mathcal{T}_h} := \sum_{i=1}^d (w_i, v_i)_{\mathcal{T}_h}$ and $(\mathbf{N}, \mathbf{Z})_{\mathcal{T}_h} := \sum_{i,j=1}^d (\mathrm{N}_{ij}, \mathrm{Z}_{ij})_{\mathcal{T}_h}$, for $\boldsymbol{w},\ \boldsymbol{v} \in \boldsymbol{V}_h$ and $\mathbf{N},\ \mathbf{Z} \in \mathbf{G}_h$. Finally, we write $\langle \eta, \zeta\rangle_{\partial\mathcal{T}_h} := \sum_{K\in\mathcal{T}_h}\langle \eta, \zeta\rangle_{\partial K}$ and $\langle \boldsymbol{\eta}, \boldsymbol{\zeta}\rangle_{\partial\mathcal{T}_h} := \sum_{i=1}^d \langle \eta_i, \zeta_i\rangle_{\partial\mathcal{T}_h}$, for $\boldsymbol{\eta},\ \boldsymbol{\zeta} \in \boldsymbol{M}_h$, where $\langle \eta, \zeta\rangle_D$ denotes the integral of $\eta\zeta$ over the domain $D \subset \mathbb{R}^{d-1}$.

3. The hybridizable discontinuous Galerkin method

3.1. Definition of the HDG method

We seek an approximation $(\mathbf{L}_h, \boldsymbol{u}_h, p_h, \hat{\boldsymbol{u}}_h)$ of $(\mathbf{L}|_\Omega, \boldsymbol{u}|_\Omega, p|_\Omega, \boldsymbol{u}|_{\mathcal{E}_h})$ in the space $\mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\boldsymbol{g})$ such that

$$(\mathbf{L}_h, \mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h, \nabla\cdot\mathbf{G})_{\mathcal{T}_h} - \langle \hat{\boldsymbol{u}}_h, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0, \tag{3a}$$

$$(\nu\mathbf{L}_h - p_h\mathbf{I} - \boldsymbol{u}_h \otimes \boldsymbol{u}_h, \nabla\boldsymbol{v})_{\mathcal{T}_h} + \left\langle \left(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h\right)\boldsymbol{n}, \boldsymbol{v}\right\rangle_{\partial\mathcal{T}_h} = (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h}, \tag{3b}$$

$$-(\boldsymbol{u}_h, \nabla q)_{\mathcal{T}_h} + \langle \hat{\boldsymbol{u}}_h \cdot \boldsymbol{n}, q\rangle_{\partial\mathcal{T}_h} = 0, \tag{3c}$$

$$\left\langle \left(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h\right)\boldsymbol{n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} = 0, \tag{3d}$$

$$(p_h, 1)_{\mathcal{T}_h} = 0, \tag{3e}$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$, where

$$-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h = -\nu\mathbf{L}_h + p_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h + \boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h) \otimes \boldsymbol{n}. \tag{4}$$

Here $\mathbf{I}$ is the second-order identity tensor, and $\boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h)$ is the *stabilization vector-valued function*. The choice of the stabilization function $\boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h)$ is crucial since it does have an important effect on both the stability and accuracy of the method. We consider an extension of the expression for $\boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h)$ proposed in [14,39] for the Stokes system as

$$\boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h) = \mathbf{S}(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h)(\boldsymbol{u}_h - \hat{\boldsymbol{u}}_h), \tag{5}$$

where $\mathbf{S}(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h)$ is the *stabilization tensor* which may depend on $\boldsymbol{u}_h$ and $\hat{\boldsymbol{u}}_h$.

Let us briefly comment on the equations defining the HDG method. The first three equations are obtained by multiplying the first three equations in (2) by test functions and integrating by parts. The fourth Eq. (3d) enforces the continuity of the normal component of the numerical trace of the total stress (including both inviscid and viscous fluxes) on the interelement boundaries, that is,

$$[\![\mathsf{P}((-\nu\mathbf{L}_h + p_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h, \hat{\boldsymbol{u}}_h))]\!] = 0 \tag{6}$$

pointwise over the interior faces $\mathcal{E}_h^{\circ}$, where P is the L^2-projection over $(\boldsymbol{M}_h)^d$. Finally, the last Eq. (3e) is just the average pressure constraint and is needed for the sake of well-posedness of the method.

It is important to note that the numerical trace $\hat{\boldsymbol{u}}_h$ is single-valued over any $F \in \mathcal{E}_h$ since $\hat{\boldsymbol{u}}_h$ belongs to $\boldsymbol{M}_h$. As a consequence, both numerical traces $\hat{\boldsymbol{u}}_h$ and $\mathsf{P}(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h)$ are *conservative* according to the definition in [1].

Note that when the nonlinear inertial term is eliminated and the stabilization tensor is independent of $\boldsymbol{u}_h$ and $\hat{\boldsymbol{u}}_h$, the HDG method we obtain is exactly the one proposed in [39] for the Stokes flow.

3.2. The general form of the numerical traces

When $\mathbf{S}$ is constant on each face F of $\partial\mathcal{T}_h$, we can rewrite the HDG method as a method given by the first three equations in (3) where all the numerical traces are expressed in terms of the approximation $(\mathbf{L}_h, \boldsymbol{u}_h, p_h)$. Inserting (4) into the jump condition (6), we obtain that, on $\mathcal{E}_h^o$,

$$[\![(-\nu\mathbf{L}_h + p_h\mathbf{I})\boldsymbol{n}]\!] + \mathbf{S}^+\boldsymbol{u}_h^+ + \mathbf{S}^-\boldsymbol{u}_h^- - (\mathbf{S}^+ + \mathbf{S}^-)\hat{\boldsymbol{u}}_h = 0.$$

Since both $\mathbf{S}^-$ and $\mathbf{S}^+$ are positive definite, and $\mathbf{S}^- + \mathbf{S}^+$ is invertible we obtain that, on $\mathcal{E}_h^o$,

$$\hat{\boldsymbol{u}}_h = \mathbf{A}(\mathbf{S}^+\boldsymbol{u}_h^+ + \mathbf{S}^-\boldsymbol{u}_h^-) + \mathbf{A}[\![(-\nu\mathbf{L}_h + p_h\mathbf{I})\boldsymbol{n}]\!], \tag{7}$$

where $\mathbf{A} = (\mathbf{S}^- + \mathbf{S}^+)^{-1}$. Substituting this expression into (4) yields on $\mathcal{E}_h^o$,

$$-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} = \mathbf{S}^-\mathbf{A}(-\nu\mathbf{L}_h^+ + p_h^+\mathbf{I}) + \mathbf{S}^+\mathbf{A}(-\nu\mathbf{L}_h^- + p_h^-\mathbf{I}) + \mathbf{S}^-\mathbf{A}\mathbf{S}^+\boldsymbol{u}_h^+ \otimes \boldsymbol{n}^+ + \mathbf{S}^+\mathbf{A}\mathbf{S}^-\boldsymbol{u}_h^- \otimes \boldsymbol{n}^-. \tag{8}$$

These two expressions define the numerical traces in terms of $(\mathbf{L}_h, \boldsymbol{u}_h, p_h)$.

Therefore, we can view the HDG method as a DG method which seeks an approximate solution $(\mathbf{L}_h, \boldsymbol{u}_h, p_h) \in (\mathbf{G}_h, \boldsymbol{V}_h, P_h)$ such that

$$(\mathbf{L}_h, \mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h, \nabla\cdot\mathbf{G})_{\mathcal{T}_h} - \langle\hat{\boldsymbol{u}}_h, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0, \tag{9a}$$

$$(\nu\mathbf{L}_h - p_h\mathbf{I} - \boldsymbol{u}_h\otimes\boldsymbol{u}_h, \nabla\boldsymbol{v})_{\mathcal{T}_h} + \left\langle\left(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h\right)\boldsymbol{n}, \boldsymbol{v}\right\rangle_{\partial\mathcal{T}_h} = (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h}, \tag{9b}$$

$$-(\boldsymbol{u}_h, \nabla q)_{\mathcal{T}_h} + \langle\hat{\boldsymbol{u}}_h\cdot\boldsymbol{n}, q\rangle_{\partial\mathcal{T}_h} = 0, \tag{9c}$$

$$(p_h, 1)_{\mathcal{T}_h} = 0 \tag{9d}$$

for all $(\mathbf{G}, \boldsymbol{v}, q) \in (\mathbf{G}_h, \boldsymbol{V}_h, P_h)$, where the numerical traces, $\hat{\boldsymbol{u}}_h$ and $-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I}$, are given by Eqs. (7) and (8).

For example, for the choice $\mathbf{S} = \tau\mathbf{I}$, where τ is a constant on $\mathcal{E}_h$, the numerical traces become

$$\begin{aligned}\hat{\boldsymbol{u}}_h &= \frac{1}{2}(\boldsymbol{u}_h^+ + \boldsymbol{u}_h^-) + \frac{1}{2\tau}[\![(-\nu\mathbf{L}_h + p_h\mathbf{I})\boldsymbol{n}]\!],\\ -\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} &= \frac{1}{2}(-\nu\mathbf{L}_h^+ + p_h^+\mathbf{I}) + \frac{1}{2}(-\nu\mathbf{L}_h^- + p_h^-\mathbf{I}) + \frac{\tau}{2}[\![\boldsymbol{u}_h\otimes\boldsymbol{n}]\!].\end{aligned} \tag{10}$$

These numerical traces of the HDG method are similar to those of the LDG method [20] except that $\hat{\boldsymbol{u}}_h$ depends on the jump term $[\![(-\nu\mathbf{L}_h + p_h\mathbf{I})\boldsymbol{n}]\!]$ for the HDG method, but not for the LDG method. This subtle difference is responsible for some significant differences in the implementation aspect and approximation properties of the methods. As regards the implementation aspect, the HDG method can reduce the globally coupled unknowns to the approximate trace of the velocity and the average of the pressure by means of a hybridization technique, whereas the LDG method has the approximate velocity and pressure as the globally coupled unknowns. As regards the approximation properties, the HDG method yields optimal convergence for all the approximate variables, whereas the LDG method produces optimal convergence for the approximate velocity only.

3.3. Local postprocessing

We apply the element-by-element postprocessing proposed in [14] to obtain a new approximate velocity which is exactly divergence-free and $\boldsymbol{H}(div)$-conforming.

In the three dimensional case, we define the postprocessed approximate velocity $\boldsymbol{u}_h^\star$ on the tetrahedron $K \in \mathcal{T}_h$ as the element of $\boldsymbol{\mathcal{P}}_{k+1}(K)$ such that

$$\langle(\boldsymbol{u}_h^\star - \hat{\boldsymbol{u}}_h)\cdot\boldsymbol{n}, \mu\rangle_F = 0 \quad \forall\mu\in\mathcal{P}_k(F), \tag{11a}$$

$$\langle(\boldsymbol{n}\times\nabla)(\boldsymbol{u}_h^\star\cdot\boldsymbol{n}) - \boldsymbol{n}\times(\{\{\mathbf{L}_h^t\}\}\boldsymbol{n}), (\boldsymbol{n}\times\nabla)\mu\rangle_F = 0 \quad \forall\mu\in\mathcal{P}_{k+1}(F)^\perp \tag{11b}$$

for all faces F of K, and such that

$$(\boldsymbol{u}_h^\star - \boldsymbol{u}_h, \nabla w)_K = 0 \quad \forall w\in\mathcal{P}_k(K), \tag{11c}$$

$$(\nabla\times\boldsymbol{u}_h^\star - \mathbf{w}_h, (\nabla\times\boldsymbol{v})\mathbf{B}_K)_K = 0 \quad \forall\boldsymbol{v}\in\boldsymbol{\mathcal{S}}_k(K). \tag{11d}$$

In (11b),

$$\mathcal{P}_{k+1}(F)^\perp := \{\mu\in\mathcal{P}_{k+1}(F) : \langle\mu, \tilde{\mu}\rangle_F = 0, \quad \forall\tilde{\mu}\in\mathcal{P}_k(F)\},$$

$\mathbf{n}\times\nabla$ is the tangential gradient and the function $\{\{\mathbf{L}_h^t\}\}$ is the single-valued function on $\mathcal{E}_h$ equal to $((\mathbf{L}_h^t)^+ + (\mathbf{L}_h^t)^-)/2$ on the set $\mathcal{E}_h\setminus\partial\Omega$ and equal to $\mathbf{L}_h^t$ on $\partial\Omega$. In (11d),

$$\mathbf{w}_h := (\mathbf{L}_{32h} - \mathbf{L}_{23h}, \mathbf{L}_{13h} - \mathbf{L}_{31h}, \mathbf{L}_{21h} - \mathbf{L}_{12h})$$

is the approximation to the vorticity and $\mathbf{B}_K$ is the so-called *symmetric bubble matrix* introduced in [12], namely,

$$\mathbf{B}_K := \sum_{\ell=0}^{3} \lambda_{\ell-3}\lambda_{\ell-2}\lambda_{\ell-1}\nabla\lambda_\ell \otimes \nabla\lambda_\ell,$$

where λ_i are the barycentric coordinates associated with the tetrahedron K, the subindices being counted modulo 4. Finally, $\mathcal{S}_k(K) := \sum_{\ell=1}^{k}\boldsymbol{S}_\ell(K)$ where $\boldsymbol{S}_\ell$ is the space of vector-valued homogeneous polynomials $\boldsymbol{v}$ of degree ℓsuch that $\boldsymbol{v}\cdot\boldsymbol{x} = 0$, see [35,36].

In the two dimensional case, the postprocessing is defined by the above equations if $\boldsymbol{n}\times\nabla$ is replaced by the tangential derivative $n_2\partial_1 + n_1\partial_2$, $\boldsymbol{n}\times\boldsymbol{a}$ is replaced by $n_1a_2 - n_2a_1$, if $\nabla\times\boldsymbol{u}$ is replaced by $\nabla\times\boldsymbol{u} := \partial_1u_2 - \partial_2u_1$, and if Eq. (11d) is replaced by

$$\left(\nabla\times\boldsymbol{u}_h^\star - \mathrm{w}_h, wb_K\right)_K = 0 \quad \forall w \in \mathcal{P}_{k-1}(K),$$

where $b_K := \lambda_0\lambda_1\lambda_2$ and $\mathrm{w}_h := \mathbf{L}_{21h} - \mathbf{L}_{12h}$.

We refer the reader to [14] for a proof of the fact that $\boldsymbol{u}_h^\star$ is a divergence-free approximate velocity in $\boldsymbol{H}(div,\Omega)$.

3.4. Imposing non-compatible boundary conditions

Let us extend the method to treat boundary conditions of the form

$$\mathbf{B}\boldsymbol{n} = \boldsymbol{g}_N, \quad \text{on } \partial\Omega_N, \tag{12}$$

where $\partial\Omega_N$ is a part of the boundary $\partial\Omega$ such that $\partial\Omega_N \cup \partial\Omega_D = \partial\Omega$ and $\partial\Omega_N \cap \partial\Omega_D = \emptyset$. Here $\mathbf{B}$ is a linear trace operator that depends on $(\mathbf{L},\boldsymbol{u},p)$. Examples of the form of $\mathbf{B}$ are given in Table 1. Note that the boundary conditions are not necessarily compatible with the weak formulation defining the HDG method.

Note that the third and fourth examples in the above table provide boundary conditions on the vorticity. Indeed, we have that $(\mathbf{L} - \mathbf{L}^t)\boldsymbol{n} = \boldsymbol{\omega}\times\boldsymbol{n}$, where $\boldsymbol{\omega} = (\mathrm{L}_{32} - \mathrm{L}_{23}, \mathrm{L}_{13} - \mathrm{L}_{31}, \mathrm{L}_{21} - \mathrm{L}_{12})$ is the vorticity vector.

In order to incorporate the above boundary condition, we redefine $\boldsymbol{M}_h(\boldsymbol{g})$ as

$$\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu}\in\boldsymbol{M}_h : \boldsymbol{\mu} = \mathsf{P}_\partial\boldsymbol{g} \text{ on } \partial\Omega_D\} \tag{13a}$$

and, for the fourth example in Table 1, as

$$\begin{aligned}&\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu}\in\boldsymbol{M}_h : \boldsymbol{\mu} = \mathsf{P}_{\partial\Omega_D}\boldsymbol{g} \text{ on } \partial\Omega_D,\\ &\boldsymbol{\mu}\cdot\boldsymbol{n} = \mathrm{P}_{\partial\Omega_N}\boldsymbol{g}\cdot\boldsymbol{n} \text{ on } \partial\Omega_N\},\end{aligned} \tag{13b}$$

with the obvious definitions for $\mathsf{P}_{\partial\Omega_D}$ and $\mathrm{P}_{\partial\Omega_N}$. We then replace the fourth Eq. (3d) in (3) with

$$\left\langle\left(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h\right)\boldsymbol{n},\boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_N} + \langle\widehat{\mathbf{B}}_h\boldsymbol{n},\boldsymbol{\mu}\rangle_{\partial\Omega_N} = \langle\boldsymbol{g}_N,\boldsymbol{\mu}\rangle_{\partial\Omega_N}, \quad \forall\boldsymbol{\mu}\in\boldsymbol{M}_h(\mathbf{0}). \tag{14}$$

Here the numerical flux $\widehat{\mathbf{B}}_h$ is an approximation to $\mathbf{B}$ on $\partial\Omega_N$ and is tabulated in Table 1. We thus obtain that $(\mathbf{L}_h,\boldsymbol{u}_h,p_h,\hat{\boldsymbol{u}}_h)\in\mathbf{G}_h\times\boldsymbol{V}_h\times P_h\times\boldsymbol{M}_h(\boldsymbol{g})$ satisfies

$$\begin{aligned}&(\mathbf{L}_h,\mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h,\nabla\cdot\mathbf{G})_{\mathcal{T}_h} - \langle\hat{\boldsymbol{u}}_h,\mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = 0,\\ &(\nu\mathbf{L}_h - p_h\mathbf{I} - \boldsymbol{u}_h\otimes\boldsymbol{u}_h,\nabla\boldsymbol{v})_{\mathcal{T}_h} + \left\langle(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h)\boldsymbol{n},\boldsymbol{v}\right\rangle_{\partial\mathcal{T}_h} = (\boldsymbol{f},\boldsymbol{v})_{\mathcal{T}_h},\\ &-(\boldsymbol{u}_h,\nabla q)_{\mathcal{T}_h} + \langle\hat{\boldsymbol{u}}_h\cdot\boldsymbol{n},q\rangle_{\partial\mathcal{T}_h} = 0,\\ &\left\langle(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h\otimes\hat{\boldsymbol{u}}_h)\boldsymbol{n},\boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h\setminus\partial\Omega_N} + \left\langle\widehat{\mathbf{B}}_h\boldsymbol{n},\boldsymbol{\mu}\right\rangle_{\partial\Omega_N} = \langle\boldsymbol{g}_N,\boldsymbol{\mu}\rangle_{\partial\Omega_N}\end{aligned} \tag{15}$$

for all $(\mathbf{G},\boldsymbol{v},q,\boldsymbol{\mu})\in\mathbf{G}_h\times\boldsymbol{V}_h\times P_h\times\boldsymbol{M}_h(\mathbf{0})$.

Table 1
Examples of other boundary conditions for the incompressible Navier–Stokes equations. Note that the asterisk symbol * indicates that the average pressure condition $(p_h,1)_\Omega = 0$ is also imposed. The dagger symbol † indicates that a Dirichlet boundary condition for the normal component of the velocity has also to be provided.

Condition type	$\mathbf{B}$	$\widehat{\mathbf{B}}_h$
Stress	$-\nu(\mathbf{L}+\mathbf{L}^t)+p\mathbf{I}$	$-\nu(\mathbf{L}_h+\mathbf{L}_h^t)+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
Viscous stress*	$-\nu(\mathbf{L}+\mathbf{L}^t)$	$-\nu(\mathbf{L}_h+\mathbf{L}_h^t)+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
Vorticity + pressure	$-\nu(\mathbf{L}-\mathbf{L}^t)+p\mathbf{I}$	$-\nu(\mathbf{L}_h-\mathbf{L}_h^t)+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
Vorticity*,†	$-\nu(\mathbf{L}-\mathbf{L}^t)$	$-\nu(\mathbf{L}_h-\mathbf{L}_h^t)+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
Gradient + pressure	$-\nu\mathbf{L}+p\mathbf{I}$	$-\nu\mathbf{L}_h+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
Gradient*	$-\nu\mathbf{L}$	$-\nu\mathbf{L}_h+\boldsymbol{s}_h(\boldsymbol{u}_h,\hat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$

Finally, note that we can consider multiple boundary conditions

$$\mathbf{B}^j \boldsymbol{n} = \boldsymbol{g}_N^j, \quad \text{on } \partial\Omega_N^j, \quad 1 \leqslant j \leqslant Q, \tag{16}$$

where $\partial\Omega_N = \bigcup_{j=1}^{Q} \partial\Omega_N^j$ and $\mathbf{B}^j$, $1 \leqslant j \leqslant Q$, are trace operators $\mathbf{B}$ whose image is evaluated on $\partial\Omega_N^j$, $1 \leqslant j \leqslant Q$, respectively. We enforce the boundary conditions (16) by replacing the last equation in (15) with

$$\left\langle \left(-\nu\widehat{\mathbf{L}}_h + \hat{p}_h\mathbf{I} + \hat{\boldsymbol{u}}_h \otimes \hat{\boldsymbol{u}}_h\right)\boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial\mathcal{T}_h \setminus \partial\Omega_N} + \sum_{j=1}^{Q} \left\langle \widehat{\mathbf{B}}_h^j \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial\Omega_N^j} = \sum_{j=1}^{Q} \left\langle \boldsymbol{g}_N^j, \boldsymbol{\mu} \right\rangle_{\partial\Omega_N^j}, \quad \forall \boldsymbol{\mu} \in \boldsymbol{M}_h(\mathbf{0}) \tag{17}$$

and by suitably modifying the space $\boldsymbol{M}_h(\cdot)$. This allows for different boundary conditions to be prescribed on different parts of the boundary.

3.5. Time-dependent incompressible Navier–Stokes equations

We end this section by extending the HDG method described above to the unsteady incompressible Navier–Stokes equations

$$\begin{aligned}
&\mathbf{L} - \nabla \boldsymbol{u} = 0, \quad \text{in } \Omega \times (0, T], \\
&\frac{\partial \boldsymbol{u}}{\partial t} - \nu \nabla \cdot \mathbf{L} + \nabla p + \nabla \cdot (\boldsymbol{u} \otimes \boldsymbol{u}) = \boldsymbol{f}, \quad \text{in } \Omega \times (0, T], \\
&\nabla \cdot \boldsymbol{u} = 0, \quad \text{in } \Omega \times (0, T], \\
&\boldsymbol{u} = \boldsymbol{g}, \quad \text{on } \partial\Omega \times (0, T], \\
&\boldsymbol{u} = \boldsymbol{u}_0, \quad \text{on } \Omega \times \{t = 0\}.
\end{aligned} \tag{18}$$

Using the backward-Euler scheme at time level t^n with timestep Δt^n we seek an approximation $(\mathbf{L}_h^n, \boldsymbol{u}_h^n, p_h^n, \hat{\boldsymbol{u}}_h^n) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\boldsymbol{g})$ such that

$$\begin{aligned}
&(\mathbf{L}_h^n, \mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h^n, \nabla \cdot \mathbf{G})_{\mathcal{T}_h} - \langle \hat{\boldsymbol{u}}_h^n, \mathbf{G}\boldsymbol{n} \rangle_{\partial\mathcal{T}_h} = 0, \\
&\left(\frac{\boldsymbol{u}_h^n}{\Delta t^n}, \boldsymbol{v}\right)_{\mathcal{T}_h} + (\nu\mathbf{L}_h^n - p_h^n\mathbf{I} - \boldsymbol{u}_h^n \otimes \boldsymbol{u}_h^n, \nabla \boldsymbol{v})_{\mathcal{T}_h} + \left\langle (-\nu\widehat{\mathbf{L}}_h^n + \hat{p}_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n \otimes \hat{\boldsymbol{u}}_h^n)\boldsymbol{n}, \boldsymbol{v} \right\rangle_{\partial\mathcal{T}_h} = (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h} + \left(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{v}\right)_{\mathcal{T}_h}, \\
&-(\boldsymbol{u}_h^n, \nabla q)_{\mathcal{T}_h} + \langle \hat{\boldsymbol{u}}_h^n \cdot \boldsymbol{n}, q \rangle_{\partial\mathcal{T}_h} = 0, \\
&\left\langle (-\nu\widehat{\mathbf{L}}_h^n + \hat{p}_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n \otimes \hat{\boldsymbol{u}}_h^n)\boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial\mathcal{T}_h} = 0, \\
&(p_h^n, 1)_{\mathcal{T}_h} = 0
\end{aligned} \tag{19}$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$, where

$$-\nu\widehat{\mathbf{L}}_h^n + \hat{p}_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n \otimes \hat{\boldsymbol{u}}_h^n = -\nu\mathbf{L}_h^n + p_h^n\mathbf{I} + \hat{\boldsymbol{u}}_h^n \otimes \hat{\boldsymbol{u}}_h^n + \boldsymbol{s}_h(\boldsymbol{u}_h^n, \hat{\boldsymbol{u}}_h^n) \otimes \boldsymbol{n}. \tag{20}$$

This discrete system is similar to the system (3) for the steady-state case except that there are two additional terms due to the backward difference discretization of the time derivative. We can thus apply the same solution procedure described in the next section for the steady-state case to the time-dependent case at every time level.

Furthermore, we note that using higher-order BDF schemes or diagonally implicit Runge–Kutta methods would yield a discrete system similar to (19). As a result, the HDG method for spatial discretization can be used with these implicit high-order time-stepping schemes to numerically solve the time-dependent incompressible Navier–Stokes system (18). This leads to the so-called *implicit high-order HDG* method.

Finally, we emphasize that the post-processing method described earlier for the steady-state case can also be applied to the time-dependent case without any modification. Moreover, the postprocessing only needs to be done at those time levels for which a more accurate result is desired.

4. Implementation

In this section, we describe the implementation of the HDG method for the steady-state case since the same procedure can be applied to the time-dependent case at every time level.

4.1. Newton–Raphson procedure

We consider the Newton–Raphson method for solving the nonlinear system (3): Given the mth current iterate $(\mathbf{L}_h^m, \boldsymbol{u}_h^m, p_h^m, \hat{\boldsymbol{u}}_h^m)$, we find an increment $(\delta\mathbf{L}_h^m, \delta\boldsymbol{u}_h^m, \delta p_h^m, \delta\hat{\boldsymbol{u}}_h^m) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$ such that

$$
\begin{aligned}
&(\delta \mathbf{L}_h^m, \mathbf{G})_{\mathcal{T}_h} + (\delta \boldsymbol{u}_h^m, \nabla\cdot\mathbf{G})_{\mathcal{T}_h} - \langle \delta\hat{\boldsymbol{u}}_h^m, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} = r^1(\mathbf{G}),\\
&(\nu\delta\mathbf{L}_h^m - \delta p_h^m\mathbf{I} - \delta\boldsymbol{u}_h^m\otimes\boldsymbol{u}_h^m - \boldsymbol{u}_h^m\otimes\delta\boldsymbol{u}_h^m, \nabla\boldsymbol{v})_{\mathcal{T}_h}\\
&\quad + \langle(-\nu\delta\mathbf{L}_h^m + \delta p_h^m\mathbf{I} + \delta\hat{\boldsymbol{u}}_h^m\otimes\hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m\otimes\delta\hat{\boldsymbol{u}}_h^m)\boldsymbol{n}, \boldsymbol{v}\rangle_{\partial\mathcal{T}_h}\\
&\quad + \langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\delta\boldsymbol{u}_h^m + \partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\delta\hat{\boldsymbol{u}}_h^m, \boldsymbol{v}\rangle_{\partial\mathcal{T}_h} = r^2(\boldsymbol{v}),\\
&-(\delta\boldsymbol{u}_h^m, \nabla q)_{\mathcal{T}_h} + \langle\delta\hat{\boldsymbol{u}}_h^m\cdot\boldsymbol{n}, q\rangle_{\partial\mathcal{T}_h} = r^3(q),\\
&\langle(-\nu\delta\mathbf{L}_h^m + \delta p_h^m\mathbf{I} + \delta\hat{\boldsymbol{u}}_h^m\otimes\hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m\otimes\delta\hat{\boldsymbol{u}}_h^m)\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} + \langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\delta\boldsymbol{u}_h^m + \partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\delta\hat{\boldsymbol{u}}_h^m, \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h} = r^4(\boldsymbol{\mu}),\\
&(\delta p_h^m, 1)_{\mathcal{T}_h} = 0
\end{aligned}
\tag{21}
$$

forall $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$, where the right-hand side residuals are evaluated from (3) at the current iterate as

$$
\begin{aligned}
r^1(\mathbf{G}) &= -(\mathbf{L}_h^m, \mathbf{G})_{\mathcal{T}_h} - (\boldsymbol{u}_h^m, \nabla\cdot\mathbf{G})_{\mathcal{T}_h} + \langle\hat{\boldsymbol{u}}_h^m, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h},\\
r^2(\boldsymbol{v}) &= (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h} - (\nu\mathbf{L}_h^m - p_h^m\mathbf{I} - \boldsymbol{u}_h^m\otimes\boldsymbol{u}_h^m, \nabla\boldsymbol{v})_{\mathcal{T}_h}\\
&\qquad - \langle(-\nu\mathbf{L}_h^m + p_h^m\mathbf{I} + \hat{\boldsymbol{u}}_h^m\otimes\hat{\boldsymbol{u}}_h^m)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m), \boldsymbol{v}\rangle_{\partial\mathcal{T}_h},\\
r^3(q) &= (\boldsymbol{u}_h^m, \nabla q)_{\mathcal{T}_h} - \langle\hat{\boldsymbol{u}}_h^m\cdot\boldsymbol{n}, q\rangle_{\partial\mathcal{T}_h},\\
r^4(\boldsymbol{\mu}) &= -\left\langle(-\nu\widehat{\mathbf{L}}_h^m + \hat{p}_h^m\mathbf{I} + \hat{\boldsymbol{u}}_h^m\otimes\hat{\boldsymbol{u}}_h^m)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m), \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h}.
\end{aligned}
\tag{22}
$$

Note that $\partial_1\boldsymbol{s}_h$ and $\partial_2\boldsymbol{s}_h$ denote the partial derivatives of $\boldsymbol{s}_h(\cdot,\cdot)$ with respect to the first and second arguments, respectively.

Below we describe two different solution strategies for solving the above linearized system. The first strategy involves introducing the mean of the pressure increment $\delta\rho_h^m \in \overline{\Psi}$ and applying a hybridization technique to obtain a reduced system in terms of $(\delta\hat{\boldsymbol{u}}_h^m, \delta\rho_h^m)$ only. The second strategy involves introducing a *pseudo* timestepping for the pressure increment and applying the hybridization technique to obtain a reduced system in terms of $\delta\hat{\boldsymbol{u}}_h^m$ at each pseudo subiteration. The first approach requires only *one* solution of the linear system for $(\delta\hat{\boldsymbol{u}}_h^m, \delta\rho_h^m)$, while the second approach requires the linear system to be solved *several* times for different right-hand sides. However, the system of the second approach is *smaller* than that of the first approach.

4.2. Solution strategy A

We observe that the first three equations in (21) define the following what we call the local solver. The local solver $\mathcal{L}_h^{\mathsf{A}}$ maps any given $(r, \boldsymbol{\eta}, \psi) \in L_h \times \boldsymbol{M}_h \times \overline{\Psi}_h$, where L_h denotes the space of linear functionals $r := (r^1, r^2, r^3)$ over $\mathbf{G}_h \times \boldsymbol{V}_h \times P_h$, to the function $(\mathfrak{L}_h, \mathfrak{u}_h, \mathfrak{p}_h) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h$ satisfying

$$
\begin{aligned}
&(\mathfrak{L}_h, \mathbf{G})_K + (\mathfrak{u}_h, \nabla\cdot\mathbf{G})_K = l_K^1(\mathbf{G}),\\
&(\nu\mathfrak{L}_h - \mathfrak{p}_h\mathbf{I} - \mathfrak{u}_h\otimes\boldsymbol{u}_h^m - \boldsymbol{u}_h^m\otimes\mathfrak{u}_h, \nabla\boldsymbol{v})_K + \langle(-\nu\mathfrak{L}_h + \mathfrak{p}_h\mathbf{I})\boldsymbol{n} + \partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\mathfrak{u}_h, \boldsymbol{v}\rangle_{\partial K} = l_K^2(\boldsymbol{v}),\\
&-(\mathfrak{u}_h, \nabla q)_K = l_K^3(q),\\
&\bar{\mathfrak{p}}_h = \psi
\end{aligned}
\tag{23}
$$

for all $(\mathbf{G}, \boldsymbol{v}, q) \in \mathbf{P}_k(K) \times \boldsymbol{P}_k(K) \times P_k(K)$, where

$$
\begin{aligned}
l_K^1(\mathbf{G}) &\equiv r_K^1(\mathbf{G}) + \langle\boldsymbol{\eta}, \mathbf{G}\boldsymbol{n}\rangle_{\partial K},\\
l_K^2(\boldsymbol{v}) &\equiv r_K^2(\boldsymbol{v}) - \langle(\boldsymbol{\eta}\otimes\hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m\otimes\boldsymbol{\eta})\boldsymbol{n}, \boldsymbol{v}\rangle_{\partial K} - \langle\partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\boldsymbol{\eta}, \boldsymbol{v}\rangle_{\partial K},\\
l_K^3(q) &\equiv r_K^3(q).
\end{aligned}
$$

for each element $K \in \mathcal{T}_h$. We take the following particular case of linear functionals:

$$
\begin{aligned}
r_K^1(\mathbf{G}) &= -(\mathbf{L}_h^m, \mathbf{G})_K - (\boldsymbol{u}_h^m, \mathrm{div}\,\mathbf{G})_K + \langle\hat{\boldsymbol{u}}_h^m, \mathbf{G}\boldsymbol{n}\rangle_{\partial K},\\
r_K^2(\boldsymbol{v}) &= (\boldsymbol{f}, \boldsymbol{v})_K + (-\nu\mathbf{L}_h^m + p_h^m\mathbf{I} + \boldsymbol{u}_h^m\otimes\boldsymbol{u}_h^m, \nabla\boldsymbol{v})_K - \langle(-\nu\mathbf{L}_h^m + p_h^m\mathbf{I} + \hat{\boldsymbol{u}}_h^m\otimes\hat{\boldsymbol{u}}_h^m)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m), \boldsymbol{v}\rangle_{\partial K},\\
r_K^3(q) &= (\boldsymbol{u}_h^m, \nabla q)_K - \langle\hat{\boldsymbol{u}}_h^m\cdot\boldsymbol{n}, q - \bar{q}\rangle_{\partial K}.
\end{aligned}
$$

We recall here that the space $\overline{\Psi}_h$ was defined in Section 2.3 and that $\bar{q}$ indicates the mean of $q \in P_h$.

Setting

$$
\begin{aligned}
(\mathfrak{L}_h^r, \mathfrak{u}_h^r, \mathfrak{p}_h^r) &:= \mathcal{L}_h^{\mathsf{A}}(r, \mathbf{0}, 0),\\
(\mathfrak{L}_h^{\boldsymbol{\eta}}, \mathfrak{u}_h^{\boldsymbol{\eta}}, \mathfrak{p}_h^{\boldsymbol{\eta}}) &:= \mathcal{L}_h^{\mathsf{A}}(\mathbf{0}, \boldsymbol{\eta}, 0),\\
(\mathfrak{L}_h^{\psi}, \mathfrak{u}_h^{\psi}, \mathfrak{p}_h^{\psi}) &:= \mathcal{L}_h^{\mathsf{A}}(\mathbf{0}, \mathbf{0}, \psi),
\end{aligned}
$$

we obtain the following result proceeding; see [39] for details.

Lemma 4.1. *Let* $(\delta \mathbf{L}_h^m, \delta \boldsymbol{u}_h^m, \delta p_h^m, \delta \hat{\boldsymbol{u}}_h^m)$ *solve the linearized system* (21)*. Then we have*

$$\begin{aligned}
\delta \mathbf{L}_h^m &= \mathfrak{L}_h^r + \mathfrak{L}_h^{\delta \lambda_h^m} + \mathfrak{L}_h^{\delta \rho_h^m},\\
\delta \boldsymbol{u}_h^m &= \mathfrak{u}_h^r + \mathfrak{u}_h^{\delta \lambda_h^m} + \mathfrak{u}_h^{\delta \rho_h^m},\\
\delta p_h^m &= \mathfrak{p}_h^r + \mathfrak{p}_h^{\delta \lambda_h^m} + \mathfrak{p}_h^{\delta \rho_h^m},\\
\delta \hat{\boldsymbol{u}}_h^m &= \delta \boldsymbol{\lambda}_h^m,
\end{aligned} \tag{24}$$

where $(\delta \boldsymbol{\lambda}_h^m, \delta \rho_h^m) \in \boldsymbol{M}_h(\mathbf{0}) \times \overline{\Psi}$ *is the solution of the following weak formulation*

$$\begin{aligned}
&a_h(\delta \boldsymbol{\lambda}_h^m, \boldsymbol{\mu}) + b_h(\delta \rho_h^m, \boldsymbol{\mu}) = \ell_h(\boldsymbol{\mu}), \quad \forall \boldsymbol{\mu} \in \boldsymbol{M}_h(\mathbf{0}),\\
&b_h(\psi, \delta \boldsymbol{\lambda}_h^m) = 0, \quad \forall \psi \in \overline{\Psi}.
\end{aligned} \tag{25}$$

Here the forms are given by

$$\begin{aligned}
a_h(\boldsymbol{\eta}, \boldsymbol{\mu}) &= \left\langle (-\nu \mathfrak{L}_h^{\boldsymbol{\eta}} + \mathfrak{p}_h^{\boldsymbol{\eta}} \mathbf{I} + \boldsymbol{\eta} \otimes \hat{\mathbf{u}}_h^m + \hat{\mathbf{u}}_h^m \otimes \boldsymbol{\eta}) \boldsymbol{n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h} + \left\langle \partial_1 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\mathbf{u}}_h^m) \mathfrak{u}_h^{\boldsymbol{\eta}} + \partial_2 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\mathbf{u}}_h^m) \boldsymbol{\eta}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h},\\
b_h(\psi, \boldsymbol{\mu}) &= -\left\langle \psi, \boldsymbol{\mu} \cdot \boldsymbol{n} \right\rangle_{\partial \mathcal{T}_h},\\
\ell_h(\boldsymbol{\mu}) &= -\left\langle (-\nu \mathfrak{L}_h^r + \mathfrak{p}_h^r \mathbf{I}) \boldsymbol{n} + \partial_1 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\mathbf{u}}_h^m) \mathfrak{u}_h^r, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h} + r^4(\boldsymbol{\mu})
\end{aligned} \tag{26}$$

for all η, $\mu \in M_h$.

The weak formulation (25) gives rise to a matrix system typical of the saddle point problem as

$$\begin{pmatrix} A^m & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \delta \Lambda^m \\ \delta \Upsilon^m \end{pmatrix} = \begin{pmatrix} R^m \\ 0 \end{pmatrix}, \tag{27}$$

where $\delta \Lambda^m$ and $\delta \Upsilon^m$ represent the vectors of degrees of freedom of $\delta \boldsymbol{\lambda}_h^m$ and $\delta \rho_h^m$, respectively. Once $\delta \Lambda^m$ and $\delta \Upsilon^m$ are obtained by solving (27), we compute the increment $(\delta \mathbf{L}_h^m, \delta \boldsymbol{u}_h^m, \delta p_h^m, \delta \hat{\boldsymbol{u}}_h^m)$ from (24).

We briefly discuss the computational complexity and memory storage required by the HDG method. In each Newton iteration, we need to form and solve the matrix system (27). The formation of the linear system (27) requires us to solve the local solver (23) for every $K \in \mathcal{T}_h$. We solve (23) by eliminating the approximate velocity gradient and pressure to arrive at a smaller system in terms of the approximate velocity only. The cost of the local solver per element is thus $O(N_u^3)$, where $N_u = \dim(\boldsymbol{\mathcal{P}}_k(K))$ is the degrees of freedom of the approximate velocity per element. Hence, the total cost of the local solver is $O(N_K N_u^3)$, where N_K is the number of elements of the triangulation.

We now describe the size and sparsity structure of the linear system (27), restricting our attention to the case of a conforming triangulation $\mathcal{T}_h$ (no hanging nodes). To this end, we note that the matrix A^m has a block structure in which the number of block rows and block columns is equal to the number of interior faces N_F. On each block row, there are at most $(2d + 1)$ non-zero blocks since one interior face is connected to $2d$ neighboring faces and the size of each block is equal to $N_\lambda \times N_\lambda$ with $N_\lambda = \dim(\boldsymbol{\mathcal{P}}_k(F))$ being the degrees of freedom of the approximate trace of the velocity per face. Therefore, the size of A^m is $N_A \times N_A$ with $N_A = N_F N_\lambda$ and the number of nonzero entries of A^m is $N_F(2d+1)N_\lambda^2 = (2d+1)N_\lambda N_A$. In addition, B has N_K rows and N_A columns, and each row of B has exactly $(d + 1)N_\lambda$ nonzero entries. Therefore, the matrix in (27) is of dimension $(N_A + N_K) \times (N_A + N_K)$ and has $(2d + 1)N_\lambda N_A + 2(d + 1)N_\lambda N_K$ nonzero entries.

In general, the solution of the linear system (27) will cost $O((N_A + N_K)^\gamma)$ with $\gamma \sim 2$ typically. Therefore, the computational complexity of the HDG method will be dominated by the cost of solving the linear system(27) since the operation count of the local solver scales linearly with N_K.

4.3. Solution strategy B

We consider the augmented Lagrangian approach [23] for solving the linearized system (21). In this approach the solution of the system (21) is computed in an iterative fashion until convergence: given the previous iterate of the pressure increment $\delta p_h^{m,n-1}$ with $\delta p_h^{m,0} = 0$, we find $(\delta \mathbf{L}_h^{m,n}, \delta \boldsymbol{u}_h^{m,n}, \delta p_h^{m,n}, \delta \hat{\boldsymbol{u}}_h^{m,n}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$ such that

$$\begin{aligned}
&(\delta \mathbf{L}_h^{m,n}, \mathbf{G})_{\mathcal{T}_h} + (\delta \boldsymbol{u}_h^{m,n}, \nabla \cdot \mathbf{G})_{\mathcal{T}_h} - \left\langle \delta \hat{\boldsymbol{u}}_h^{m,n}, \mathbf{G}\boldsymbol{n} \right\rangle_{\partial \mathcal{T}_h} = r^1(\mathbf{G}),\\
&(\nu \delta \mathbf{L}_h^{m,n} - \delta p_h^{m,n} \mathbf{I} - \delta \boldsymbol{u}_h^{m,n} \otimes \boldsymbol{u}_h^m - \boldsymbol{u}_h^m \otimes \delta \boldsymbol{u}_h^{m,n}, \nabla \boldsymbol{v})_{\mathcal{T}_h}\\
&\quad + \left\langle (-\nu \delta \mathbf{L}_h^{m,n} + \delta p_h^{m,n} \mathbf{I} + \delta \hat{\boldsymbol{u}}_h^{m,n} \otimes \hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m \otimes \delta \hat{\boldsymbol{u}}_h^{m,n}) \boldsymbol{n}, \boldsymbol{v} \right\rangle_{\partial \mathcal{T}_h}\\
&\quad + \left\langle \partial_1 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m) \delta \boldsymbol{u}_h^{m,n} + \partial_2 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m) \delta \hat{\boldsymbol{u}}_h^{m,n}, \boldsymbol{v} \right\rangle_{\partial \mathcal{T}_h} = r^2(\boldsymbol{v}),\\
&\beta(\delta p_h^{m,n} - \delta p_h^{m,n-1}, q)_{\mathcal{T}_h} - (\delta \boldsymbol{u}_h^{m,n}, \nabla q)_{\mathcal{T}_h} + \left\langle \delta \hat{\boldsymbol{u}}_h^{m,n} \cdot \boldsymbol{n}, q \right\rangle_{\partial \mathcal{T}_h} = r^3(q)\\
&\left\langle (-\nu \delta \mathbf{L}_h^{m,n} + \delta p_h^{m,n} \mathbf{I} + \delta \hat{\boldsymbol{u}}_h^{m,n} \otimes \hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m \otimes \delta \hat{\boldsymbol{u}}_h^{m,n}) \boldsymbol{n} \right\rangle_{\partial \mathcal{T}_h}\\
&\quad + \left\langle \partial_1 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m) \delta \boldsymbol{u}_h^{m,n} + \partial_2 \boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m) \delta \hat{\boldsymbol{u}}_h^{m,n}, \boldsymbol{\mu} \right\rangle_{\partial \mathcal{T}_h} = r^4(\boldsymbol{\mu})
\end{aligned} \tag{28}$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0})$. Here β is a given positive constant. Note also that $(\delta p_h^{m,n}, 1)_{\mathcal{T}_h} = 0$ is always satisfied since we start with $\delta p_h^{m,0} = 0$.

We stop the iterations when the error in the pressure increment is less than a prescribed tolerance $\varepsilon_{\mathrm{tol}}$, that is, when

$$\left\| \beta\left(\delta p_h^{m,n} - \delta p_h^{m,n-1}\right) \right\|_{\mathcal{T}_h} < \varepsilon_{\mathrm{tol}}. \tag{29}$$

The final iterate of (28) is then an approximate solution of the original system (21). It remains to describe how to solve the linear system (28).

Toward that end, we observe that the first three equations in (28) define the following local solver: The local solver $\mathcal{L}_h^{\mathrm{B}}$ maps any given $(r, \boldsymbol{\eta}, \varphi) \in L_h \times \boldsymbol{M}_h \times P_h$, where L_h denotes the space of linear functionals $r := (r^1, r^2, r^3)$ over $\mathbf{G}_h \times \boldsymbol{V}_h \times P_h$, to the function $(\mathfrak{L}_h, \mathfrak{u}_h, \mathfrak{p}_h) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h$ satisfying

$$\begin{aligned}
&(\mathfrak{L}_h, \mathbf{G})_K + (\mathfrak{u}_h, \nabla\cdot\mathbf{G})_K = l_K^1(\mathbf{G}),\\
&(\nu\mathfrak{L}_h - \mathfrak{p}_h\mathbf{I} - \mathfrak{u}_h \otimes \boldsymbol{u}_h^m - \boldsymbol{u}_h^m \otimes \mathfrak{u}_h, \nabla\boldsymbol{v})_K + \left\langle(-\nu\mathfrak{L}_h + \mathfrak{p}_h\mathbf{I})\boldsymbol{n} + \partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\mathfrak{u}_h, \boldsymbol{v}\right\rangle_{\partial K} = l_K^2(\boldsymbol{v}),\\
&\beta(\mathfrak{p}_h, q)_{\mathcal{T}_h} - (\mathfrak{u}_h, \nabla q)_K = l_K^{3,\beta}(q)
\end{aligned} \tag{30}$$

for all $(\mathbf{G}, \boldsymbol{v}, q) \in \mathbf{P}_k(K) \times \boldsymbol{P}_k(K) \times P_k(K)$, where

$$\begin{aligned}
&l_K^1(\mathbf{G}) \equiv r_K^1(\mathbf{G}) + \langle\boldsymbol{\eta}, \mathbf{G}\boldsymbol{n}\rangle_{\partial K},\\
&l_K^2(\boldsymbol{v}) \equiv r_K^2(\boldsymbol{v}) - \left\langle(\boldsymbol{\eta}\otimes\hat{\boldsymbol{u}}_h^m + \hat{\boldsymbol{u}}_h^m\otimes\boldsymbol{\eta})\boldsymbol{n}, \boldsymbol{v}\right\rangle_{\partial K} - \left\langle\partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\boldsymbol{\eta}, \boldsymbol{v}\right\rangle_{\partial K},\\
&l_K^{3,\beta}(q) \equiv r_K^3(q) + \beta(\varphi, q)_{\mathcal{T}_h}
\end{aligned}$$

for each element $K \in \mathcal{T}_h$.

Again, setting

$$\begin{aligned}
&(\mathfrak{L}_h^r, \mathfrak{u}_h^r, \mathfrak{p}_h^r) := \mathcal{L}_h^{\mathrm{B}}(r, \mathbf{0}, 0),\\
&(\mathfrak{L}_h^{\boldsymbol{\eta}}, \mathfrak{u}_h^{\boldsymbol{\eta}}, \mathfrak{p}_h^{\boldsymbol{\eta}}) := \mathcal{L}_h^{\mathrm{B}}(\mathbf{0}, \boldsymbol{\eta}, 0),\\
&(\mathfrak{L}_h^{\varphi}, \mathfrak{u}_h^{\varphi}, \mathfrak{p}_h^{\varphi}) := \mathcal{L}_h^{\mathrm{B}}(\mathbf{0}, \mathbf{0}, \varphi),
\end{aligned}$$

we obtain the following result proceeding; see [39] for details.

Lemma 4.2. *Let $(\delta\mathbf{L}_h^{m,n}, \delta\boldsymbol{u}_h^{m,n}, \delta p_h^{m,n}, \delta\hat{\boldsymbol{u}}_h^{m,n})$ solve the linear system (28). Then we have*

$$\begin{aligned}
\delta\mathbf{L}_h^{m,n} &= \mathfrak{L}_h^r + \mathfrak{L}_h^{\delta p_h^{m,n-1}} + \mathfrak{L}_h^{\delta\boldsymbol{\lambda}_h^{m,n}},\\
\delta\boldsymbol{u}_h^{m,n} &= \mathfrak{u}_h^r + \mathfrak{u}_h^{\delta p_h^{m,n-1}} + \boldsymbol{u}_h^{\delta\boldsymbol{\lambda}_h^{m,n}},\\
\delta p_h^{m,n} &= \mathfrak{p}_h^r + \mathfrak{p}_h^{\delta p_h^{m,n-1}} + \mathfrak{p}_h^{\delta\boldsymbol{\lambda}_h^{m,n}},\\
\delta\hat{\boldsymbol{u}}_h^{m,n} &= \delta\boldsymbol{\lambda}_h^{m,n},
\end{aligned} \tag{31}$$

where $\delta\boldsymbol{\lambda}_h^{m,n} \in \boldsymbol{M}_h(\mathbf{0})$ is the solution of the following weak formulation

$$c_h(\delta\boldsymbol{\lambda}_h^{m,n}, \boldsymbol{\mu}) = f_h(\boldsymbol{\mu}), \quad \forall\boldsymbol{\mu} \in \boldsymbol{M}_h(\mathbf{0}). \tag{32}$$

Here the forms are given by

$$\begin{aligned}
c_h(\boldsymbol{\eta}, \boldsymbol{\mu}) &= \left\langle(-\nu\mathfrak{L}_h^{\boldsymbol{\eta}} + \mathfrak{p}_h^{\boldsymbol{\eta}}\mathbf{I} + \boldsymbol{\eta}\otimes\hat{\mathbf{u}}_h^m + \hat{\mathbf{u}}_h^m\otimes\boldsymbol{\eta})\boldsymbol{n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} + \left\langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\mathfrak{u}_h^{\boldsymbol{\eta}} + \partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\boldsymbol{\eta}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h},\\
f_h(\boldsymbol{\mu}) &= r^4(\boldsymbol{\mu}) + \left\langle\left(\nu\left(\mathfrak{L}_h^r + \mathfrak{L}_h^{\delta p_h^{m,n-1}}\right) - \left(\mathfrak{p}_h^r + \mathfrak{p}_h^{\delta p_h^{m,n-1}}\right)\mathbf{I}\right)\boldsymbol{n}, \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h} - \left\langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \hat{\boldsymbol{u}}_h^m)\left(\mathfrak{u}_h^r + \mathfrak{u}_h^{\delta p_h^{m,n-1}}\right), \boldsymbol{\mu}\right\rangle_{\partial\mathcal{T}_h}
\end{aligned}$$

for all η, $\mu \in M_h$.

The weak formulation (32) gives rise to a matrix system of the form

$$C^m\delta\Lambda^{m,n} = F^{m,n}, \tag{33}$$

where $\delta\Lambda^{m,n}$ represents the vectors of degrees of freedom of $\delta\lambda_h^{m,n}$. It is important to note that the matrix C^m does not change during the augmented Lagrangian iteration as it is formed once for every Newton–Raphson iteration. The matrix C^m has the same size and sparsity structure as the matrix A^m of (27), so that the system (33) is smaller than (27). However, this sub-timestepping approach requires a number of sub-iterations to converge to the solution of the system (27).

5. Numerical results

In this section, we present numerical results to assess the performance of the HDG method for several problems including the Kovasznay flow [30], lid-driven cavity flow [25], channel expansion flow [42], Taylor vortex flow [44], and natural convective flow in a cavity [24]. We also examine the convergence and accuracy properties of the method for a wide range of Reynolds numbers and for various polynomial degrees.

We have implemented both the solution strategies. The first strategy requires only one solution of the system (27), while the second strategy requires a number of sub-iterations to converge. However, the number of sub-iterations depends only on β and the timestep Δt^n, *not* on the mesh size h and polynomial degree k. In our experiments we select $\beta = 0.01$ and observe that the number of sub-iterations required is about 10 in all cases. Moreover, we use a sparse direct solver for solving the resulting linear systems.

In all the presented examples, the stabilization tensor $\mathbf{S}$ has the form

$$\mathbf{S} = \begin{pmatrix} \tau & 0 \\ 0 & \tau \end{pmatrix},$$

where τ is some positive constant defined on $\mathcal{E}_h$ and typically chosen such that

$$\tau \approx \frac{\nu}{\ell} + |\boldsymbol{u}|. \tag{34}$$

Here ℓ is a representative length scale and $|\boldsymbol{u}|$ is the magnitude of the velocity. This choice is based on dimensional analysis which shows that τ has the same dimensions as the ratio ν/ℓ and the velocity magnitude: the former accounts for the diffusion effect and the latter accounts for the convection effect. We refer to [37,38] for a detailed discussion on the choice of the stabilization parameter τ for convection–diffusion problems and to [14] for the Stokes equations.

5.1. Kovasznay flow

We consider the Kovasznay flow [30] with the analytical solution

$$\begin{aligned} u_1 &= 1 - \exp(\lambda x_1)\cos(2\pi x_2), \\ u_2 &= \frac{\lambda}{2\pi}\exp(\lambda x_1)\sin(2\pi x_2), \\ p &= -\frac{1}{2}\exp(2\lambda x_1) + C, \end{aligned}$$

where $\lambda = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$ and $Re = \frac{1}{\nu}$ is the Reynolds number. We take Dirichlet boundary conditions for the velocity as the restriction of the exact solution to the domain boundary and the Reynolds number $Re = 20$ so that $\nu = 0.05$. The computational domain is $\Omega = (0,2) \times (-0.5, 1.5)$.

In our experiments, we consider triangular meshes that are obtained by splitting a regular $n \times n$ Cartesian grid into a total of $2n^2$ triangles, giving uniform element sizes of $h = 2/n$. On these meshes, we consider polynomials of degree k to represent all the approximate variables using a nodal basis within each element, with the nodes uniformly distributed.

We present the error and order of convergence in L^2-norm in Table 2 for $\tau = 1$. We observe that all the approximate variables converge optimally with order $k + 1$ for $k = 1, 2, 3$. It is also seen that the postprocessed velocity $\boldsymbol{u}_h^*$ converges with order $k + 2$, which is one order higher than the original velocity $\boldsymbol{u}_h$. We further show plots of streamline of the approximate velocity and the postprocessed velocity in Fig. 1 for $k = 1$ on the grid 8 × 8. We see that the approximate velocity can be significantly

Table 2
History of convergence of the HDG method for $\tau = 1$ for the Kovasznay problem with Reynolds number $Re = 20$.

Degree	Mesh	$\Vert\boldsymbol{u}-\boldsymbol{u}_h\Vert_{\mathcal{T}_h}$		$\Vert p-p_h\Vert_{\mathcal{T}_h}$		$\Vert\mathbf{L}-\mathbf{L}_h\Vert_{\mathcal{T}_h}$		$\Vert\boldsymbol{u}-\boldsymbol{u}_h^\star\Vert_{\mathcal{T}_h}$	
k	n	Error	Order	Error	Order	Error	Order	Error	Order
1	4	3.88e − 1	–	3.27e − 1	–	3.82e − 0	–	3.35e − 1	–
	8	1.37e − 1	1.51	8.03e − 2	2.02	1.59e − 0	1.27	8.26e − 2	2.02
	16	2.96e − 2	2.20	1.81e − 2	2.15	4.90e − 1	1.70	1.13e − 2	2.87
	32	7.06e − 3	2.07	4.14e − 3	2.13	1.39e − 1	1.81	1.57e − 3	2.85
	64	1.73e − 3	2.03	9.96e − 4	2.06	3.74e − 2	1.90	2.09e − 4	2.91
2	4	2.09e − 1	–	1.48e − 1	–	2.16e − 0	–	1.38e − 1	–
	8	1.66e − 2	3.66	9.02e − 3	4.03	2.37e − 1	3.19	8.28e − 3	4.05
	16	1.90e − 3	3.13	9.32e − 4	3.27	3.37e − 2	2.81	5.47e − 4	3.92
	32	2.30e − 4	3.04	1.12e − 4	3.06	4.62e − 3	2.87	3.75e − 5	3.87
	64	2.85e − 5	3.02	1.38e − 5	3.02	6.08e − 4	2.93	2.46e − 6	3.93
3	4	2.45e − 2	–	1.57e − 2	–	3.00e − 1	–	1.42e − 2	–
	8	1.55e − 3	3.98	7.93e − 4	4.31	2.47e − 2	3.60	5.68e − 4	4.64
	16	9.30e − 5	4.06	5.01e − 5	3.98	1.73e − 3	3.83	1.89e − 5	4.91
	32	5.74e − 6	4.02	3.18e − 6	3.98	1.16e − 4	3.90	6.37e − 7	4.89
	64	3.57e − 7	4.01	2.00e − 7	3.99	7.52e − 6	3.95	2.07e − 8	4.94

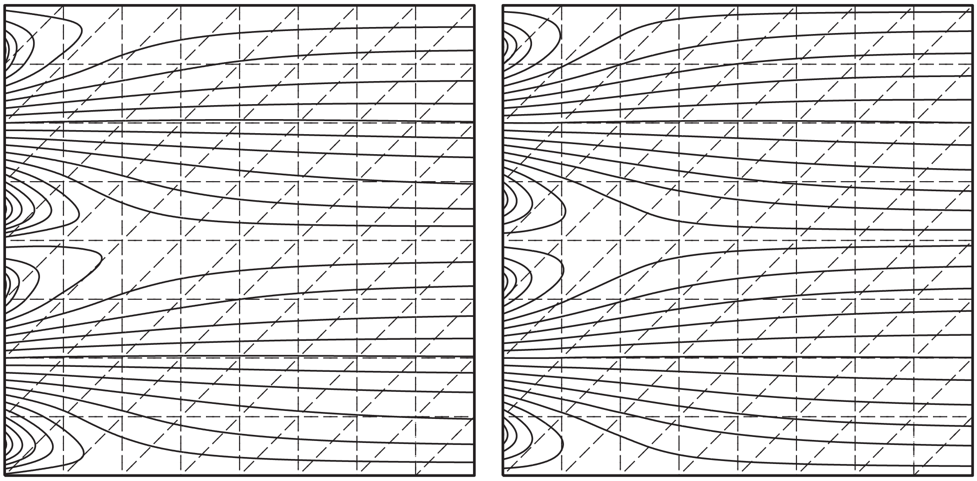

Fig. 1. Streamline of the approximate velocity $\boldsymbol{u}_h$ (left) and the postprocessed velocity $\boldsymbol{u}_h^*$ (right) for the Kovasznay problem using $k = 1$ on the grid 8 × 8.

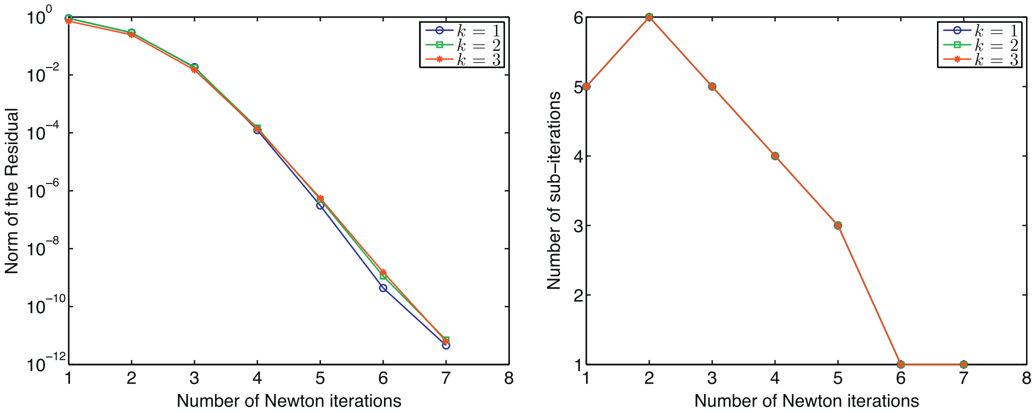

Fig. 2. Newton convergence of the HDG method (left) and the number of sub-iterations versus the number of Newton iterations (right) on the grid 16 × 16.

improved by the local postprocessing since the postprocessed velocity $\boldsymbol{u}_h^*$ is clearly superior to the original velocity $\boldsymbol{u}_h$. Moreover, unlike the original velocity, the postprocessed velocity is exactly divergence-free and $\boldsymbol{H}$(div)-conforming.

Using the augmented Lagrangian method (solution strategy B) we show the Newton convergence and the number of sub-iterations required in Fig. 2. Here the norm of the residual is defined as the two-norm of the right-hand-side vector of (27) or (28). We observe that the Newton method converges very rapidly to the norm of the residual of 10^{-10} in 7 iterations. Furthermore, the number of sub-iterations is small (less than 7) and tends to decrease with the Newton steps. Note that using the solution strategy A, we obtain the same results in terms of the Newton convergence and accuracy. As the linear systems are solved using direct methods, the solution strategy A is faster than the solution strategy B since it requires only one linear solve per Newton step. We shall thus consider the solution strategy A for the other examples.

In summary, the HDG method yields the $(k + 1)$ th-order of convergence for the pressure and velocity gradient, and in conjunction with the local postprocessing produces the $(k + 2)$ th-order of convergence for the velocity (when $k \geqslant 1$). In contrast, for many other DG methods, the approximate pressure and velocity gradient converges with order k, while the approximate velocity converges with order $k + 2$. Moreover, the HDG method has less the globally coupled degrees of freedom than other DG methods since it solves for the approximate trace of the velocity and the mean of the pressure.

5.2. Lid-driven cavity flow

The lid-driven cavity flow has been widely used as a validation case for numerical methods of the incompressible Navier–Stokes equations. The problem has simple geometry and boundary conditions. The standard case is fluid contained in a square domain $\Omega = (0, 1) \times (0, 1)$ with homogeneous Dirichlet boundary conditions on all sides except on the upper side, where the velocity is prescribed as (1,0). Despite its simple geometry, the lid-driven cavity problem poses some difficulties for any numerical method due to the singularity of the solution at the upper corners, the rapid change of the flow at high Reynolds number, and the appearance of rotating vortices with significantly different sizes.

Fig. 3 shows a computational grid 32 × 32 on which our numerical solutions are obtained for Reynolds numbers $Re = 100$, 200, 400, 1000, 2500, and $Re = 5000$. The number of elements is thus 2048. The grid is refined along the wall in order to capture the boundary layer and the complex fluid structure near the wall. The minimum grid spacing is $h_{\min} = 3.8 \times 10^{-3}$ and the maximum grid spacing is $h_{\max} = 1.0 \times 10^{-1}$. The steady-state solution is computed by the HDG method described in Section 3. The approximate solution for the lowest Reynolds $Re = 100$ is computed from the solution of the corresponding Stokes problem as initial guess, meanwhile the approximate solution for a higher Reynolds number is computed from the approximate solution for a lower Reynolds numbers as initial guess. The stabilization parameter $\tau = 1$ is used for all cases.

We next show the profiles of the velocity along the centerlines for $Re = 1000$ and $Re = 5000$ in Fig. 4. The horizontal velocity profiles exhibit a kink near the upper wall $y = 1$, while a similar behavior is observed for the vertical velocity profiles near the right wall $x = 1$. Such behavior has been reported in [3,25]. Moreover, we observe that our numerical results are in very good agreement with the results previously reported in [25]. Note for $Re = 1000$ that the results for $k = 1$ are very good despite the fact that our computational grid 32 × 32 is quite coarse. However, for $Re = 5000$, the velocity profiles for $k = 1$ are slightly different near the wall due to the boundary layer effect for higher Reynolds number. We also present in Fig. 5 the pressure profiles along the centerlines for $Re = 1000$ and $Re = 5000$. We observe a similar trend of convergence: the approximate pressure converges rapidly with increasing k. Unlike the case of the velocity profiles which have larger errors near the wall than at the center of the domain, the pressure profiles have larger error at the center of the domain than at the wall. This is because the grid is refined along the wall in order to resolve the boundary layer.

Finally, we display in Fig. 6 the streamline contours of the postprocessed velocity $\boldsymbol{u}_h^*$ and in Fig. 7 the approximate pressure for $Re = 1000$ and $Re = 5000$. We clearly observe the typical structure of the steady-state solution for the lid-driven cavity flow [3,25]: there are various secondary vortices near the corners and the size of the secondary vortices increases with the Reynolds number. Moreover, these structures remain clearly observed for $k = 1$ even though our mesh 32 × 32 is significantly

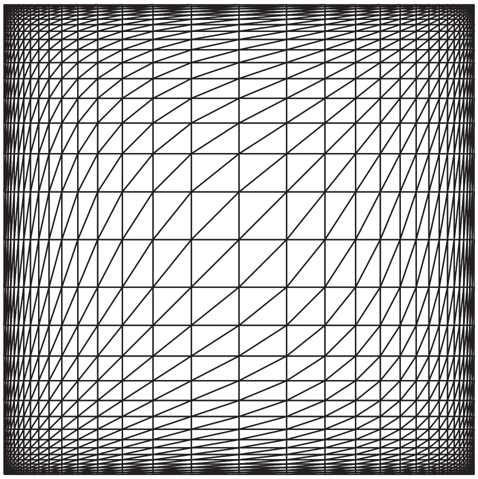

Fig. 3. Finite element mesh for the lid-driven cavity flow.

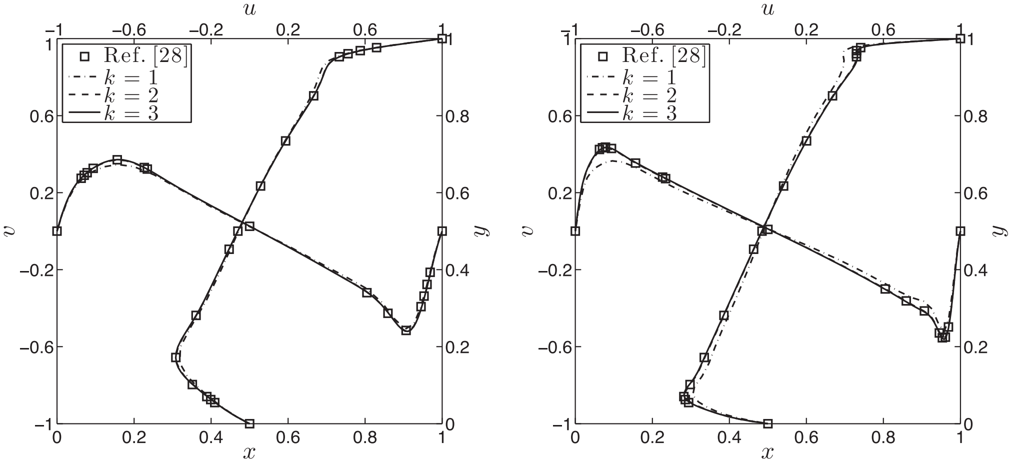

Fig. 4. The u- and v- components of the approximate velocity $\boldsymbol{u}_h$ along the horizontal and vertical centerlines for $Re = 1000$ (left) and $Re = 5000$ (right) on the grid 32 × 32 for the lid-driven cavity flow.

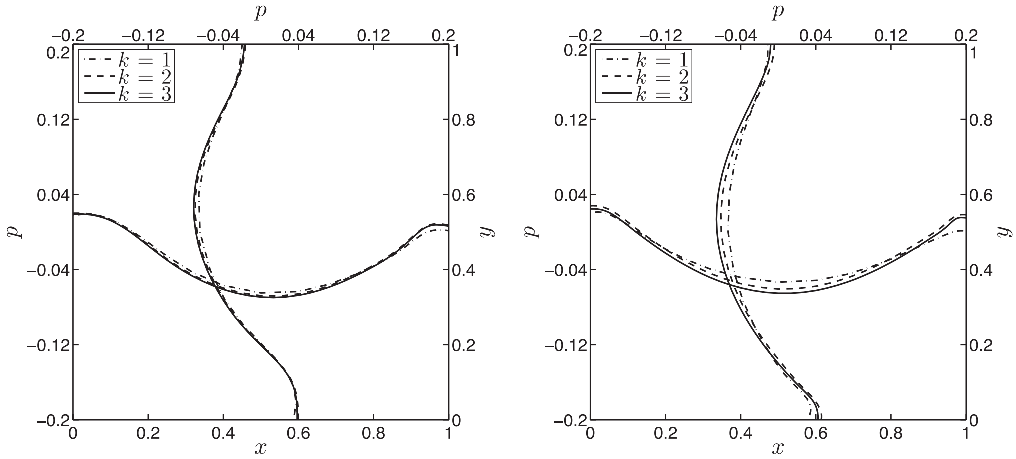

Fig. 5. The pressure along the horizontal and vertical centerlines for *Re* = 1000 (left) and *Re* = 5000 (right) on the grid 32 × 32 for the lid-driven cavity flow.

coarser than the meshes (129 × 129 and 257 × 257) used in the previous calculation [25]. However, we note that our local postprocessing is not as effective as it was in the previous example in improving the accuracy of the velocity approximation. In particular, the postprocessed velocity looks very similar to the original velocity. This is due to the presence of singularities at the corners as well as to the relatively high Reynolds numbers.

5.3. Laminar flow in a channel expansion

Laminar flow in a channel expansion has also been studied by many authors as a test for numerical schemes. We choose the same geometry and boundary conditions given in [42] in order to compare our results with those presented in [42]. The channel geometry and computational mesh are shown in Fig. 8. The governing equations are given by

$$\begin{aligned}
&-\frac{1}{Re}\Delta\boldsymbol{u}+\nabla p+\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u})=\boldsymbol{0}, \quad \text{in } \Omega,\\
&\nabla\cdot\boldsymbol{u}=0, \quad \text{in } \Omega,\\
&\boldsymbol{u}=(0,0), \quad \text{on } \Gamma_{\text{wall}},\\
&\boldsymbol{u}=(1-y^2,0), \quad \text{on } \Gamma_{\text{in}},\\
&\frac{1}{Re}\nabla\boldsymbol{u}\boldsymbol{n}-p\boldsymbol{n}=(0,0), \quad \text{on } \Gamma_{\text{out}}.
\end{aligned}$$

These equations are nondimensionalized with respect to the inlet channel half-width H and the maximum velocity at inflow U_0. The Reynolds number is defined as $Re = U_0H/\nu$. The approximate solution of the corresponding Stokes problem is used as an initial solution in the HDG method for solving the governing equations. The numerical results are obtained for $Re = 109.5$ with the stabilization parameter being set to $\tau = 1$.

We present in Fig. 9 the contour of the approximate pressure and its zoom near the step corner for $k = 2$ and $k = 3$, and in Fig. 10 the contour of the streamline of the postprocessed velocity. We observe that the high pressure at the step and the effect of recirculation on the vorticity distribution in the corner are all consistent with physical intuition. Moreover, we see that the reattachment point is located at the point $(x_r, y_r) = (5, -2)$. This figure is the same as the value reported in [42]. We also find that the streamfunction attains its minimum at the center of the vortex, which is $(x_m, y_m) = (1.5, -1.4)$. This figure agrees with the experimental value and numerical value obtained using finite differences [22]. In Fig. 11 we display the streamwise velocity profiles at several locations downstream of the step. The agreement with [42] is very good.

5.4. Taylor vortex problem

The Taylor vortex problem [44] is a well-known example of the unsteady incompressible Navier–Stokes equations. The exact solution is

$$\begin{aligned}
u_x &= -\cos(\pi x)\sin(\pi y)\exp\left(\frac{-2\pi^2 t}{Re}\right),\\
u_y &= \sin(\pi x)\cos(\pi y)\exp\left(\frac{-2\pi^2 t}{Re}\right),\\
p &= -\frac{1}{4}(\cos(2\pi x)+\cos(2\pi y))\exp\left(\frac{-4\pi^2 t}{Re}\right),
\end{aligned}$$

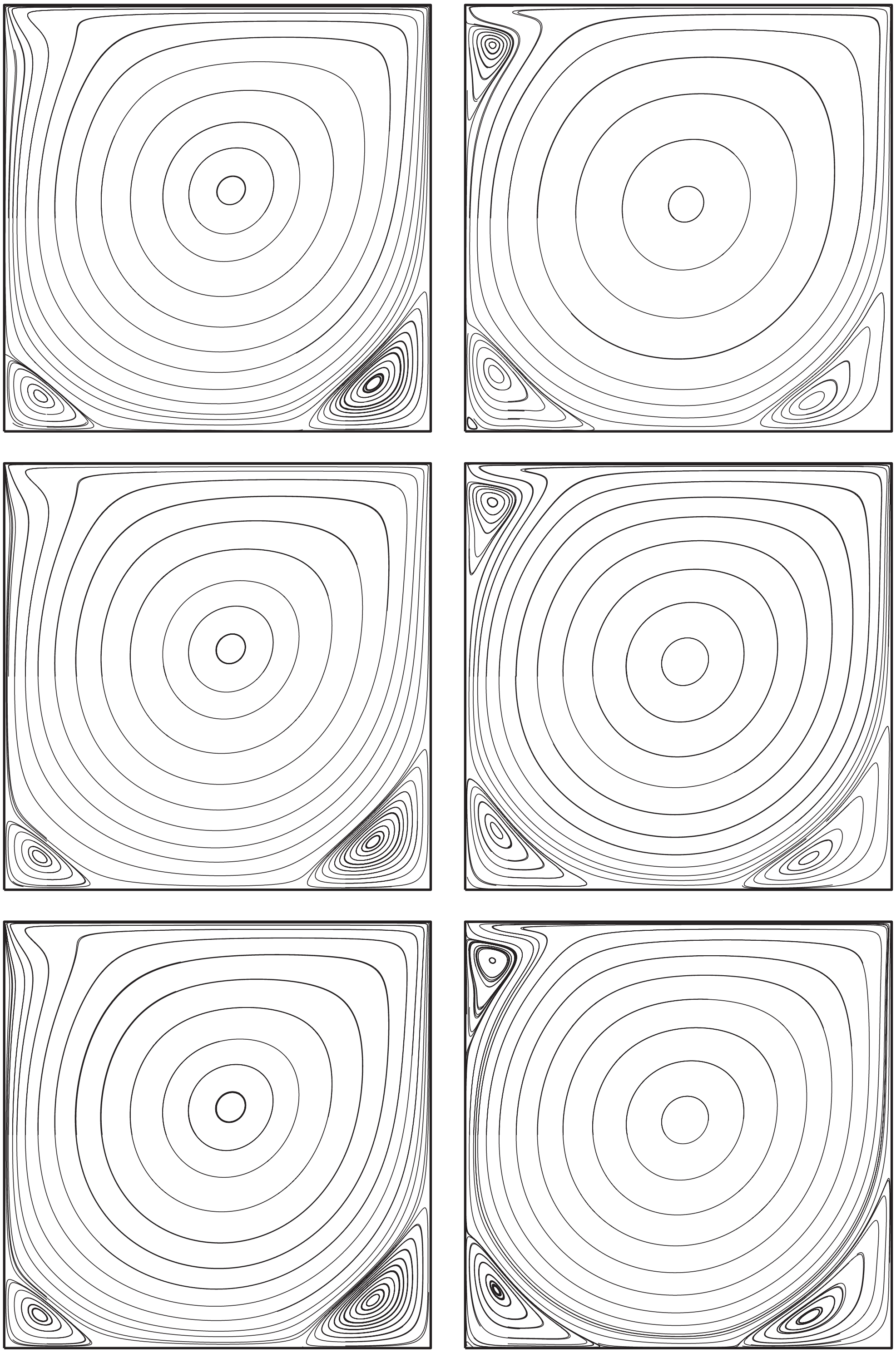

Fig. 6. Streamline of the postprocessed velocity $\boldsymbol{u}_h^*$ for $Re = 1000$ (left) and $Re = 5000$ (right) using $k = 1$ (top), $k = 2$ (middle), and $k = 3$ (bottom) on the grid 32×32 for the lid-driven cavity flow.

where $Re = 1/\nu$ is the Reynolds number. We consider the above problem on $\Omega = (0,1)^2$ with Reynolds number $Re = 20$ and final time $T = 1$. We take the Dirichlet boundary condition for the velocity as the restriction of the exact solution to the domain boundary and the initial condition as an instantiation of the exact solution at $t = 0$.

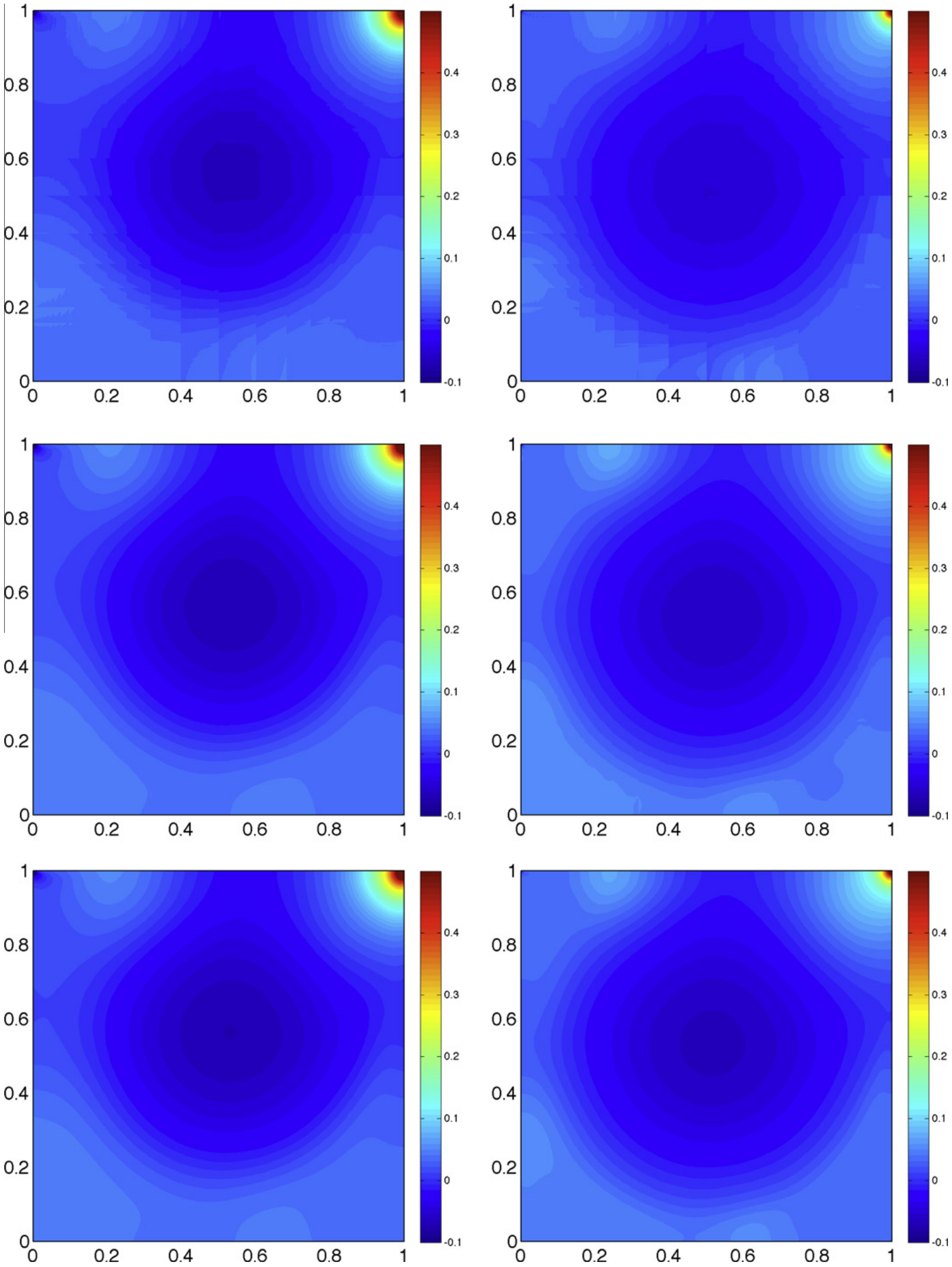

Fig. 7. Plots of the approximate pressure for Re = 1000 (left) and Re = 5000 (right) using k = 1 (top), k = 2 (middle), and k = 3 (bottom) on the grid 32 × 32 for the lid-driven cavity flow.

We consider triangular meshes that are obtained by splitting a regular $n \times n$ Cartesian grid into a total of $2n^2$ triangles, giving uniform element sizes of $h = 1/n$. We use the third-order backward difference formula (BDF3) for the temporal discretization. The stabilization parameter τ is set to 1 on $\partial\mathcal{T}_h$.

We first look at the convergence and accuracy in terms of both k and h refinements. For this purpose, we select a small constant timestep of $\Delta t = 0.005$, so that the spatial error is dominant and the temporal error is negligible. We present in Table 3 the history of convergence of the HDG method at the final time $t = 1$. We observe that the approximate velocity, pressure, and velocity gradient converge with the optimal order $k + 1$ for k = 1, 2, 3. The fact that the HDG method yields

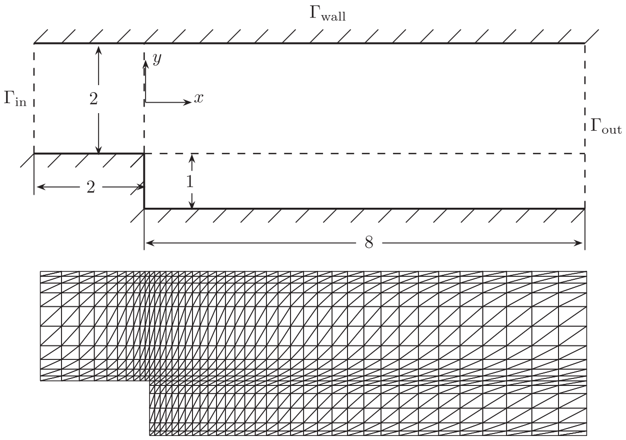

Fig. 8. Geometry configuration and computational mesh for the channel expansion.

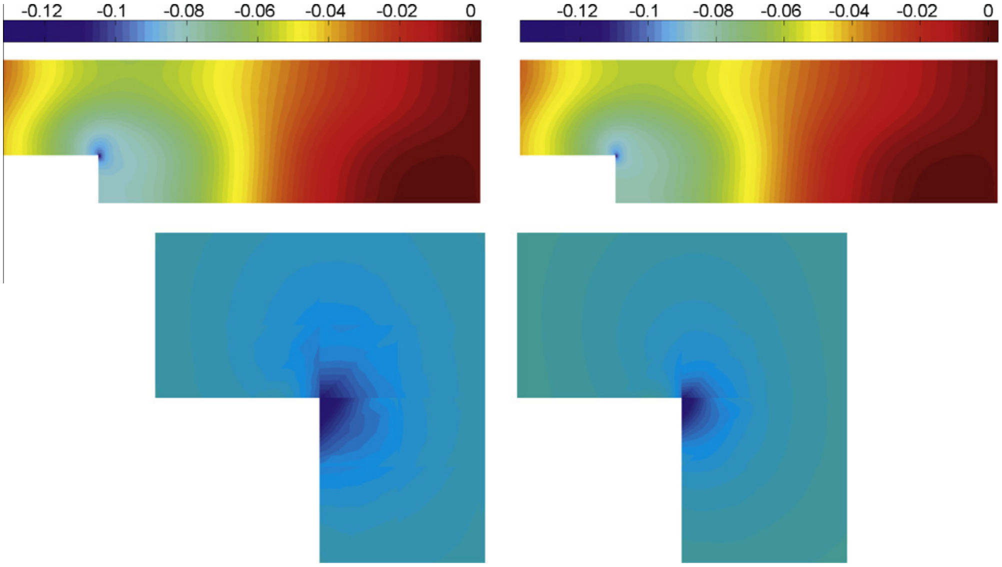

Fig. 9. Plots of the approximate pressure and its zoom near the step corner for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow. The improved resolution obtained with $k = 3$ is apparent from the figure.

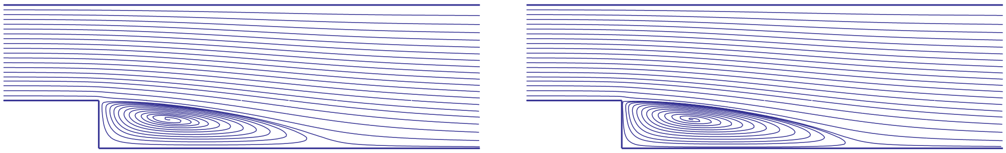

Fig. 10. Plots of the postprocessed streamline for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow.

optimal convergence for both the approximate pressure and velocity gradient is a very important advantage since all known DG methods provide suboptimal convergence of order k for the approximate pressure and velocity gradient.

Equally important is the fact that the postprocessed velocity $\boldsymbol{u}_h^*$ converges with the order $k + 2$, which is one order higher than the original approximate velocity $\boldsymbol{u}_h$. Furthermore, we emphasize that $\boldsymbol{u}_h^*$ is an exactly divergence-free and $\boldsymbol{H}$(div)-conforming velocity field. To visualize the effect of the local postprocessing, we show in Fig. 12 the plots of the approximate

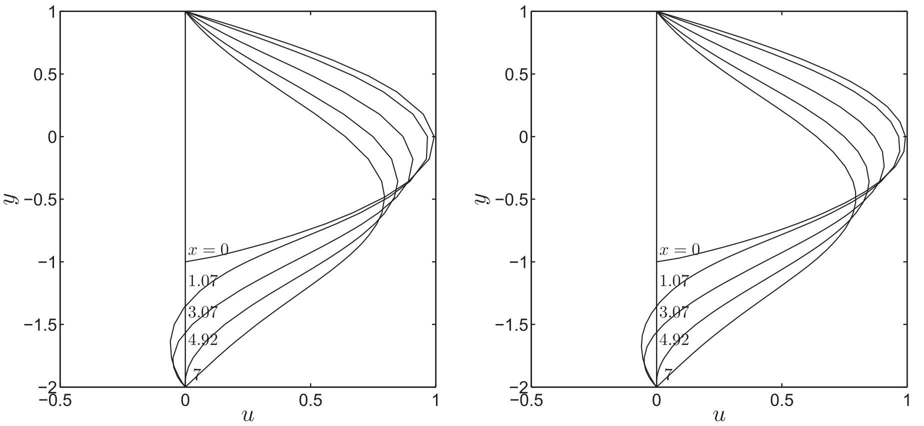

Fig. 11. A plot of the streamwise velocity profiles for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow.

Table 3
History of convergence of the HDG method for the Taylor vortex problem when the velocity Dirichlet condition is applied on the entire boundary.

Degree	Mesh	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{\mathcal{T}_h}$		$\|p-p_h\|_{\mathcal{T}_h}$		$\|\mathbf{L}-\mathbf{L}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{u}-\boldsymbol{u}_h^{\star}\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	error	Order	Error	Order	Error	Order
1	4	4.73e − 2	–	3.44e − 2	–	3.29e − 1	–	3.40e − 2	–
	8	1.27e − 2	1.89	8.59e − 3	2.00	1.26e − 1	1.39	8.04e − 3	2.08
	16	2.94e − 3	2.11	2.14e − 3	2.01	3.85e − 2	1.71	1.34e − 3	2.59
	32	6.95e − 4	2.08	5.38e − 4	1.99	1.07e − 2	1.84	1.89e − 4	2.82
	64	1.70e − 4	2.03	1.36e − 4	1.99	2.85e − 3	1.91	2.50e − 5	2.92
2	4	1.14e − 2	–	6.67e − 3	–	1.04e − 1	–	8.35e − 3	–
	8	1.26e − 3	3.17	8.43e − 4	2.98	1.72e − 2	2.60	6.12e − 4	3.77
	16	1.51e − 4	3.06	1.07e − 4	2.98	2.60e − 3	2.73	4.07e − 5	3.91
	32	1.87e − 5	3.01	1.33e − 5	3.00	3.64e − 4	2.84	2.70e − 6	3.91
	64	2.33e − 6	3.00	1.67e − 6	3.00	4.85e − 5	2.91	1.76e − 7	3.94
3	2	1.81e − 3	–	1.00e − 3	–	2.01e − 2	–	1.22e − 3	–
	4	1.08e − 4	4.06	7.00e − 5	3.84	1.72e − 3	3.54	4.67e − 5	4.70
	8	6.59e − 6	4.04	4.33e − 6	4.01	1.29e − 4	3.74	1.63e − 6	4.84
	16	4.08e − 7	4.01	2.68e − 7	4.01	8.92e − 6	3.85	5.48e − 8	4.89
	32	2.55e − 8	4.00	1.67e − 8	4.00	5.88e − 7	3.92	1.82e − 9	4.91

velocity and the postprocessed velocity for $k = 2$ on the grid $h = 1/2$. We observe that the local postprocessing does provide a significant improvement in the approximation of the velocity field, since $\boldsymbol{u}_h^*$ is clearly superior to $\boldsymbol{u}_h$.

Moreover, since the local postprocessing is performed at the element level and only at the timestep where higher accuracy is desired, it adds very little to the overall computational cost. As a result, with the HDG method, the $(k + 2)$-convergent velocity, $(k + 1)$-convergent pressure, and $(k + 1)$-convergent velocity gradient can be computed at the cost of a DG approximation using polynomials of degree k.

5.5. Testing other boundary conditions

We now look at the effect of applying different boundary conditions on different parts of the boundary on the accuracy of the HDG method. We use the Taylor vortex problem presented above for our investigation. We denote by Γ_{left}, Γ_{bottom}, Γ_{right}, and Γ_{top} the left, bottom, right, and top sides of the domain, respectively. We consider six cases of boundary conditions as shown in Table 4. The first three cases correspond to a Neumann condition on one side and three velocity Dirichlet conditions on the other three sides, while the last three cases correspond to two Neumann conditions on two sides and two velocity Dirichlet conditions on the other two sides.

We present the results in Table 5 for the first three cases and in Table 6 for the last three cases. We observe that all the approximate variables converge with the optimal order $k + 1$ for all six cases. Furthermore, the postprocessed velocity converges with order $k + 2$ for the first case, but with order $k + 1$ for the other cases. This can be explained by noting that the gradient-pressure condition in the first case is compatible with the weak formulation of the HDG method, whereas the vorticity-pressure and stress-pressure conditions in the other cases are not.

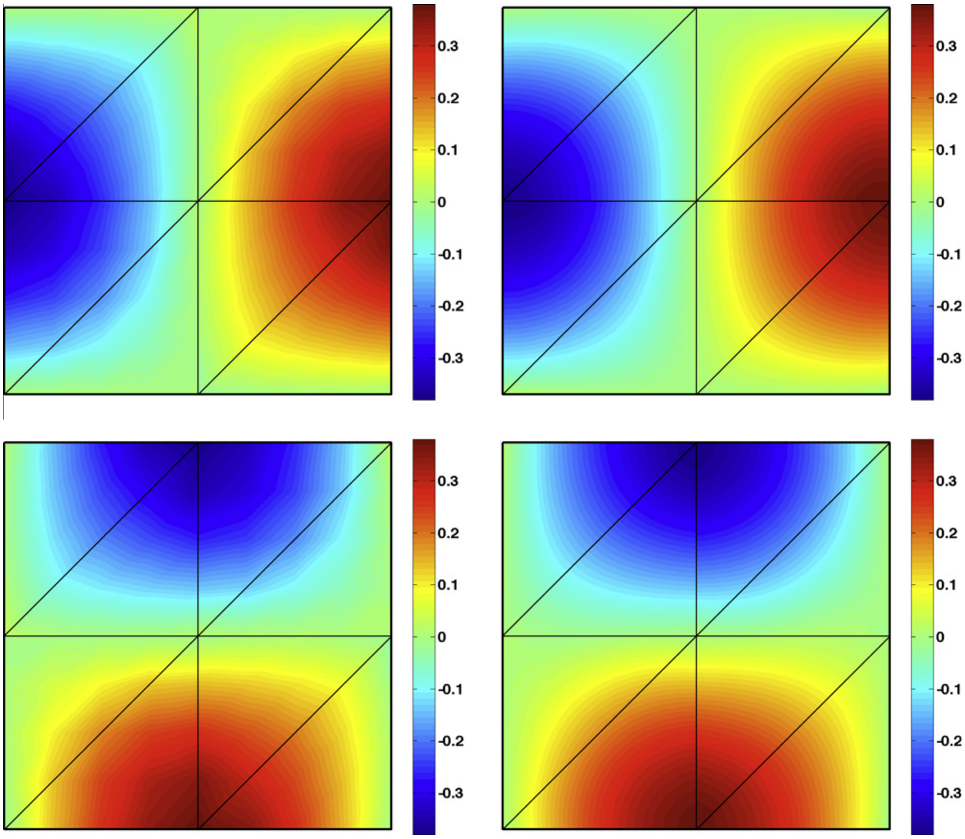

Fig. 12. The approximate velocity $\boldsymbol{u}_h$ (left) and the postprocessed velocity $\boldsymbol{u}_h^*$ (right) obtained using $k = 2$ on the grid $h = 1/2$ for the Taylor vortex problem. The horizontal velocity is placed at the top, while the vertical velocity at the bottom.

Table 4
Six cases of boundary conditions for the Taylor vortex problem.

Case	Γ_{left}	Γ_{bottom}	Γ_{right}	Γ_{top}
1	Velocity	Gradient-pressure	Velocity	Velocity
2	Velocity	Velocity	Vorticity-pressure	Velocity
3	Velocity	Velocity	velocity	Stress
4	Velocity	Gradient-pressure	Vorticity-pressure	Velocity
5	Velocity	Gradient-pressure	velocity	Stress
6	Velocity	Velocity	Vorticity-pressure	Stress

5.6. The effect of Fixing $\Delta t/h$

We revisit the Taylor vortex example to illustrate the effect of taking a fixed ratio $\Delta t/h$. However, for this purpose we consider the Reynolds number $Re = 100$ and the final time $T = 10$. We use the BDF3 scheme for the temporal discretization and take the timestep to be $\Delta t = 10h$, so that $\Delta t/h = 10$. We present in Table 7 the history of convergence of the HDG method at the final time. Unlike the previous results obtained by taking Δt very small, the convergence rates are affected by the accuracy order of the BDF3 method when we fix the ratio $\Delta t/h$. More specifically, for $k = 1$ and $k = 2$, all the approximate variables converge with the optimal order $k + 1$. However, for $k = 3$ and $k = 3$, all the approximate variables converge with order 3 only, which is the same order of accuracy as the BDF3 scheme. These results indicate that when $\Delta t/h$ is kept fixed we should use a $k + 1$-order time-stepping method in order to obtain optimal convergence properties. Moreover, as expected, the order of convergence of the postprocessed velocity is also affected in the same way. In particular, the postprocessed converges with order at most 3 for all k. As a result, it converges with order $k + 2$ for only $k = 1$. Therefore, the time-stepping method must be accurate with order $k + 2$ in order to obtain super-convergence of the postprocessed velocity.

Table 5
History of convergence of the HDG method for the Taylor vortex problem for the first three cases of boundary conditions shown in Table 4.

Degree	Mesh	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{\mathcal{T}_h}$		$\|p-p_h\|_{\mathcal{T}_h}$		$\|\mathbf{L}-\mathbf{L}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{u}-\boldsymbol{u}_h^\star\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order
Case 1									
1	2	5.11e − 2	–	3.85e − 2	–	3.53e − 1	–	3.97e − 2	–
	4	1.26e − 2	2.02	9.05e − 3	2.09	1.29e − 1	1.45	7.91e − 3	2.33
	8	2.98e − 3	2.08	2.17e − 3	2.06	3.92e − 2	1.72	1.43e − 3	2.47
	16	7.00e − 4	2.09	5.40e − 4	2.01	1.09e − 2	1.85	2.11e − 4	2.76
	32	1.70e − 4	2.04	1.35e − 4	2.00	2.87e − 3	1.92	2.83e − 5	2.89
2	2	1.28e − 2	–	7.56e − 3	–	1.19e − 1	–	1.06e − 2	–
	4	1.33e − 3	3.26	8.53e − 4	3.15	1.77e − 2	2.75	7.43e − 4	3.83
	8	1.52e − 4	3.13	1.05e − 4	3.02	2.62e − 3	2.75	4.36e − 5	4.09
	16	1.87e − 5	3.02	1.32e − 5	2.99	3.66e − 4	2.84	2.75e − 6	3.99
	32	2.33e − 6	3.00	1.66e − 6	2.99	4.86e − 5	2.91	1.77e − 7	3.96
Case 2									
1	2	4.96e − 2	–	4.07e − 2	–	3.36e − 1	–	3.80e − 2	–
	4	1.35e − 2	1.88	9.13e − 3	2.16	1.30e − 1	1.37	9.37e − 3	2.02
	8	3.06e − 3	2.14	2.26e − 3	2.01	3.95e − 2	1.72	1.60e − 3	2.55
	16	7.14e − 4	2.10	5.71e − 4	1.99	1.10e − 2	1.84	2.52e − 4	2.67
	32	1.75e − 4	2.03	1.44e − 4	1.99	2.92e − 3	1.92	4.85e − 5	2.37
2	2	1.20e − 2	–	7.74e − 3	–	1.10e − 1	–	9.63e − 3	–
	4	1.29e − 3	3.21	8.95e − 4	3.11	1.81e − 2	2.60	6.90e − 4	3.80
	8	1.54e − 4	3.07	1.12e − 4	3.00	2.72e − 3	2.73	5.03e − 5	3.78
	16	1.90e − 5	3.02	1.39e − 5	3.01	3.75e − 4	2.86	4.25e − 6	3.56
	32	2.36e − 6	3.01	1.72e − 6	3.01	4.94e − 5	2.92	3.80e − 7	3.48
Case 3									
1	2	5.03e − 2	–	4.29e − 2	–	3.48e − 1	–	3.88e − 2	–
	4	1.20e − 2	2.07	1.14e − 2	1.92	1.29e − 1	1.43	6.85e − 3	2.50
	8	2.98e − 3	2.01	2.69e − 3	2.08	4.00e − 2	1.69	1.41e − 3	2.28
	16	7.63e − 4	1.97	6.45e − 4	2.06	1.12e − 2	1.84	3.67e − 4	1.95
	32	1.96e − 4	1.96	1.59e − 4	2.02	2.97e − 3	1.92	1.01e − 4	1.86
2	2	1.08e − 2	–	7.25e − 3	–	1.09e − 1	–	7.84e − 3	–
	4	1.25e − 3	3.12	8.28e − 4	3.13	1.77e − 2	2.62	5.8e − 4	3.76
	8	1.54e − 4	3.02	1.08e − 4	2.94	2.66e − 3	2.73	5.12e − 5	3.50
	16	1.95e − 5	2.98	1.39e − 5	2.95	3.71e − 4	2.84	6.22e − 6	3.04
	32	2.47e − 6	2.98	1.78e − 6	2.97	4.92e − 5	2.91	8.37e − 7	2.89

5.7. Natural convective flow in a cavity

Finally, we consider the natural convective flow in a laterally heated cavity presented in [24]. The geometry and boundary conditions are depicted in Fig. 13. We investigate the numerical solution of this problem by the HDG method for the Prandtl number $Pr = 0$ and Grashof number $Gr = 10^6$. In this case the governing equations are given by

$$\begin{aligned}
&\frac{\partial \boldsymbol{u}}{\partial t} - \Delta\boldsymbol{u} + \nabla p + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{u}) = Gr\theta\,\boldsymbol{e}_y, \quad \text{in } \Omega\times(0,T],\\
&\nabla\cdot\boldsymbol{u} = 0, \quad \text{in } \Omega\times(0,T],\\
&\boldsymbol{u} = \boldsymbol{0}, \quad \text{on } \partial\Omega\times(0,T],\\
&\boldsymbol{u} = \boldsymbol{0}, \quad \text{on } \Omega\times\{t=0\},
\end{aligned}$$

where $\theta = 1 - x/4$ is the temperature distribution in the rectangular cavity and $\boldsymbol{e}_y = (0,1)$ is the unit vector in the vertical direction. Here the final time is $T = 0.1$.

In this example, since the magnitude of the velocity scales like $\sqrt{Gr}$, we choose the stabilization parameter $\tau = \sqrt{Gr}$, in agreement with (34). Furthermore, we use the second-order backward difference formula (BDF2) for the temporal discretization and take the timestep to be $\Delta t = 2\times10^{-4}$. The finite element mesh shown in Fig. 13 is uniform with $h = 0.1$, giving the cell Peclet number about 100. Because this mesh is relatively coarse for the Grashof number $Gr = 10^6$, we compute the numerical solution by using high polynomial degrees $k = 3$ and $k = 4$.

We present in Fig. 14 the postprocessed approximate velocity for $k = 3, 4$ at the spatial point $(2, 0.8)$. We observe the onset of oscillatory instability in the flow and that the flow becomes periodic unstable after $t = 0.08$. Note that for $Pr = 0$ considered here, the temperature is not perturbed at all. Hence, the instability is of purely hydrodynamic origin. Fig. 15 shows the streamline of the approximate velocity at different time levels. We see that the periodic unstable flows consists of three primary convective rolls. As seen from the streamline patterns, the oscillations of the streamlines are most noticeable between

Table 6
History of convergence of the HDG method for the Taylor vortex problem for the last three cases of boundary conditions shown in Table 4.

Degree	Mesh	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{\mathcal{T}_h}$		$\|p-p_h\|_{\mathcal{T}_h}$		$\|\mathbf{L}-\mathbf{L}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{u}-\boldsymbol{u}_h^{\star}\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order
Case 4									
1	2	7.17e − 2	–	3.63e − 2	–	3.85e − 1	–	6.44e − 2	–
	4	1.55e − 2	2.21	8.31e − 3	2.13	1.32e − 1	1.54	1.22e − 2	2.40
	8	3.55e − 3	2.13	2.13e − 3	1.97	4.01e − 2	1.72	2.42e − 3	2.33
	16	8.33e − 4	2.09	5.44e − 4	1.97	1.11e − 2	1.85	5.01e − 4	2.27
	32	2.04e − 4	2.03	1.38e − 4	1.98	2.93e − 3	1.92	1.15e − 4	2.12
2	2	1.52e − 2	–	6.60e − 3	–	1.20e − 1	–	1.36e − 2	–
	4	1.60e − 3	3.25	8.28e − 4	3.00	1.85e − 2	2.70	1.16e − 3	3.55
	8	1.73e − 4	3.21	1.05e − 4	2.99	2.75e − 3	2.75	9.44e − 5	3.62
	16	2.03e − 5	3.09	1.32e − 5	2.98	3.78e − 4	2.86	8.50e − 6	3.47
	32	2.47e − 6	3.04	1.67e − 6	2.99	4.96e − 5	2.93	8.42e − 7	3.34
Case 5									
1	2	5.61e − 2	–	3.75e − 2	–	3.72e − 1	–	4.64e − 2	–
	4	1.57e − 2	1.84	9.93e − 3	1.92	1.37e − 1	1.45	1.23e − 2	1.92
	8	4.20e − 3	1.90	2.44e − 3	2.03	4.20e − 2	1.70	3.28e − 3	1.90
	16	1.05e − 3	2.00	6.03e − 4	2.01	1.16e − 2	1.86	8.09e − 4	2.02
	32	2.64e − 4	1.99	1.51e − 4	2.00	3.04e − 3	1.93	2.04e − 4	1.99
2	2	1.46e − 2	–	7.16e − 3	–	1.28e − 1	–	1.28e − 2	–
	4	1.31e − 3	3.48	8.31e − 4	3.11	1.81e − 2	2.82	7.15e − 4	4.16
	8	1.57e − 4	3.07	1.04e − 4	2.99	2.68e − 3	2.76	5.90e − 5	3.60
	16	2.05e − 5	2.94	1.35e − 5	2.96	3.73e − 4	2.85	8.82e − 6	2.74
	32	2.67e − 6	2.94	1.72e − 6	2.97	4.94e − 5	2.92	1.31e − 6	2.75
Case 6									
1	2	8.32e − 2	–	3.97e − 2	–	3.90e − 1	–	7.76e − 2	–
	4	1.57e − 2	2.41	1.07e − 2	1.90	1.37e − 1	1.50	1.24e − 2	2.65
	8	3.72e − 3	2.07	2.7e − 3	1.98	4.24e − 2	1.70	2.67e − 3	2.21
	16	9.04e − 4	2.04	6.71e − 4	2.01	1.16e − 2	1.87	6.10e − 4	2.13
	32	2.27e − 4	1.99	1.68e − 4	1.99	3.03e − 3	1.93	1.53e − 4	2.00
2	2	1.27e − 2	–	7.06e − 3	–	1.17e − 1	–	1.06e − 2	–
	4	1.35e − 3	3.23	8.04e − 4	3.14	1.87e − 2	2.65	7.95e − 4	3.74
	8	1.61e − 4	3.07	1.05e − 4	2.94	2.79e − 3	2.75	6.98e − 5	3.51
	16	2.14e − 5	2.91	1.38e − 5	2.92	3.87e − 4	2.85	1.08e − 5	2.70
	32	3.01e − 6	2.83	1.83e − 6	2.91	5.51e − 5	2.81	1.91e − 6	2.49

Table 7
History of convergence of the HDG method for the Taylor vortex problem using the BDF3 scheme with $\Delta t/h = 10$.

Degree	Mesh	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{\mathcal{T}_h}$		$\|p-p_h\|_{\mathcal{T}_h}$		$\|\mathbf{L}-\mathbf{L}_h\|_{\mathcal{T}_h}$		$\|\boldsymbol{u}-\boldsymbol{u}_h^{\star}\|_{\mathcal{T}_h}$	
k	$1/h$	Error	Order	Error	Order	Error	Order	Error	Order
1	2	1.91e − 2	–	1.09e − 2	–	1.66e − 1	–	1.57e − 2	–
	4	8.29e − 3	1.20	2.63e − 3	2.05	9.50e − 2	0.81	7.66e − 3	1.04
	8	1.67e − 3	2.31	6.89e − 4	1.93	3.57e − 2	1.41	1.42e − 3	2.43
	16	3.38e − 4	2.31	1.79e − 4	1.94	1.24e − 2	1.52	2.44e − 4	2.55
	32	7.09e − 5	2.25	4.64e − 5	1.95	3.90e − 3	1.67	3.67e − 5	2.73
	64	1.62e − 5	2.13	1.19e − 5	1.97	1.12e − 3	1.80	5.15e − 6	2.84
2	2	5.16e − 3	–	1.71e − 3	–	5.66e − 2	–	4.55e − 3	–
	4	6.54e − 4	2.98	2.43e − 4	2.81	1.32e − 2	2.10	5.46e − 4	3.06
	8	6.44e − 5	3.34	3.31e − 5	2.88	2.52e − 3	2.39	4.07e − 5	3.74
	16	7.45e − 6	3.11	4.35e − 6	2.93	4.38e − 4	2.52	3.54e − 6	3.52
	32	9.00e − 7	3.05	5.65e − 7	2.94	6.84e − 5	2.68	3.25e − 7	3.45
	64	1.11e − 7	3.02	7.27e − 8	2.96	9.80e − 6	2.80	3.31e − 8	3.30
3	2	1.06e − 3	–	5.01e − 4	–	1.42e − 2	–	9.85e − 4	–
	4	1.18e − 4	3.17	7.62e − 5	2.72	1.96e − 3	2.85	1.14e − 4	3.12
	8	1.39e − 5	3.08	1.01e − 5	2.91	2.21e − 4	3.15	1.37e − 5	3.05
	16	1.79e − 6	2.96	1.32e − 6	2.94	2.55e − 5	3.12	1.78e − 6	2.95
	32	2.27e − 7	2.97	1.68e − 7	2.97	3.03e − 6	3.07	2.27e − 7	2.97
4	2	3.99e − 4	–	4.73e − 4	–	5.58e − 3	–	3.97e − 4	–
	4	1.09e − 4	1.88	7.41e − 5	2.68	1.39e − 3	2.01	1.09e − 4	1.87
	8	1.36e − 5	3.00	1.01e − 5	2.88	1.75e − 4	2.99	1.36e − 5	3.00
	16	1.78e − 6	2.94	1.32e − 6	2.94	2.28e − 5	2.94	1.78e − 6	2.94
	32	2.27e − 7	2.97	1.68e − 7	2.97	2.91e − 6	2.97	2.27e − 7	2.97

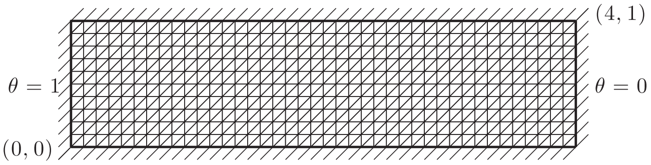

Fig. 13. Geometry, boundary conditions, and finite element mesh for the natural convective flow in a laterally heated cavity.

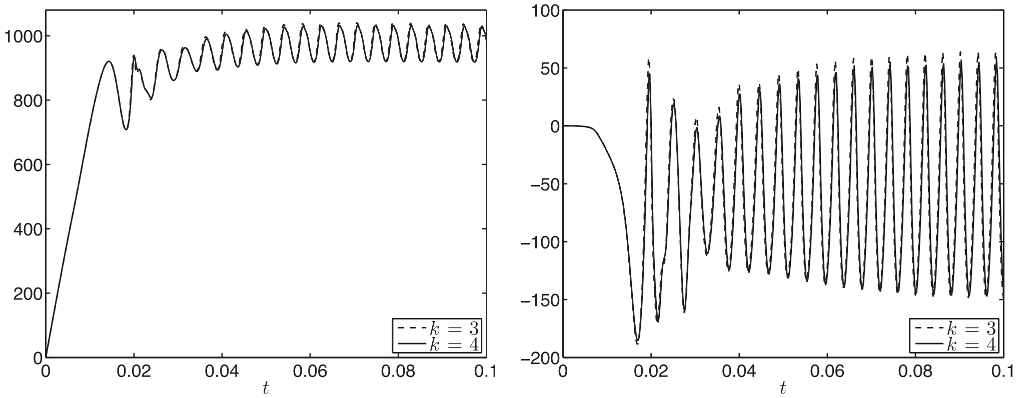

Fig. 14. The postprocessed approximate velocity for $k = 3, 4$ at the spatial point (2,0.8) as a function of t for the natural convective flow. The horizontal velocity is shown on the left, while the vertical velocity on the right.

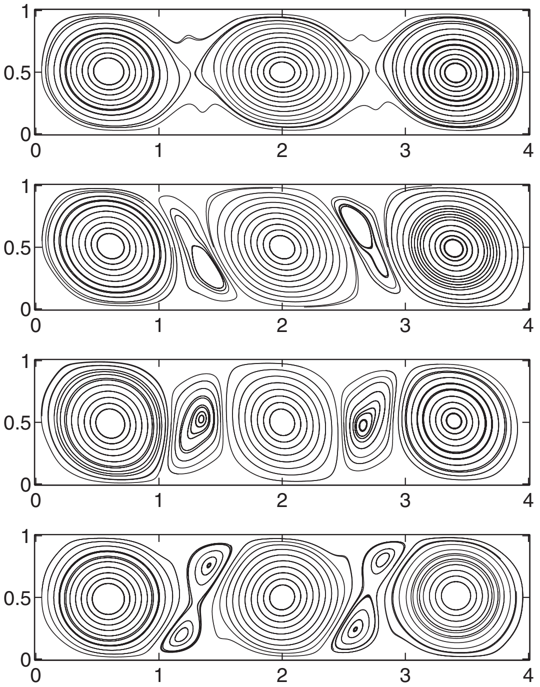

Fig. 15. Streamline of the postprocessed approximate velocity for $k = 4$ at time $t = 0.095$ (top), $t = 0.096$ (second), $t = 0.097$ (third), and $t = 0.098$ (bottom) for the natural convective flow.

the primary rolls. This indicates that the oscillatory instability is caused by a hydrodynamic interaction between the rolls. These general observations agree well with the previous calculations based on the standard Galerkin and finite-volume methods [24].

6. Conclusions

We have presented a hybridizable discontinuous Galerkin method for the numerical solution of the incompressible Navier–Stokes equations. As indicated in the Introduction, the method holds important advantages over many existing DG methods in terms of the reduction of the number of globally-coupled degrees of freedom, in the convergence and accuracy properties of the approximation and in the ability to handle a wide variety of boundary conditions. The numerical results show that the HDG method is efficient for solving the steady and unsteady incompressible Navier–Stokes equations.

We are currently developing efficient iterative methods for solving the linear system arising from application of the Newton–Raphson procedure. The problems being addressed include natural convection flows, boundary layers, stability and transition in general curved geometries. The theoretical analysis of the method also constitutes the subject of ongoing work.

Acknowledgements

J. Peraire and N. C. Nguyen would like to acknowledge the Singapore-MIT Alliance and the Air Force Office of Scientific Research under the MURI project on Biologically Inspired Flight for partially supporting this work. B. Cockburn would like to acknowledge the National Science Foundation for partially supporting this work through Grant DMS-0712955.

References

[1] D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems, SIAM J. Numer. Anal. 39 (5) (2001) 1749–1779.

[2] G.A. Baker, W.N. Jureidini, O.A. Karakashian, Piecewise solenoidal vector fields and the Stokes problem, SIAM J. Numer. Anal. 27 (1990) 1466–1485.

[3] F. Bassi, A. Crivellini, D.A. Di Pietro, S. Rebay, An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equation, J. Comput. Phys. 218 (2) (2006) 794–815.

[4] J. Carrero, B. Cockburn, D. Schötzau, Hybridized globally divergence-free LDG methods. I. The Stokes problem, Math. Comput. 75 (2006) 533–563.

[5] B. Cockburn, Discontinuous Galerkin methods, ZAMM Z. Angew. Math. Mech. 83 (2003) 731–754.

[6] B. Cockburn, Discontinuous Galerkin Methods for Computational Fluid Dynamics, in: R. de Borst, E. Stein, T.J.R. Hughes (Eds.), Encyclopedia of Computational Mechanics, vol. 3, John Wiley & Sons, Ltd., England, 2004, pp. 91–123.

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

[8] B. Cockburn, B. Dong, J. Guzmán, M. Restelli, R. Sacco, Superconvergent and optimally convergent LDG-hybridizable discontinuous Galerkin methods for convection–diffusion–reaction problems, SIAM J. Sci. Comput. 31 (2009) 3827–3846.

[9] B. Cockburn, J. Gopalakrishnan, Incompressible finite elements via hybridization. Part I: the Stokes system in two space dimensions, SIAM J. Numer. Anal. 43 (4) (2005) 1627–1650.

[10] B. Cockburn, J. Gopalakrishnan, Incompressible finite elements via hybridization. Part II: the stokes system in three space dimensions, SIAM J. Numer. Anal. 43 (4) (2005) 1651–1672.

[11] B. Cockburn, J. Gopalakrishnan, The derivation of hybridizable discontinuous Galerkin methods for Stokes flow, SIAM J. Numer. Anal. 47 (2009) 1092–1125.

[12] B. Cockburn, J. Gopalakrishnan, J. Guzmán, A new elasticity element made for enforcing weak stress symmetry, Math. Comput. 79 (2010) 1331–1349.

[13] 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 (2009) 1319–1365.

[14] B. Cockburn, J. Gopalakrishnan, N.C. Nguyen, J. Peraire, F.-J. Sayas, Analysis of an HDG method for Stokes flow, Math. Comput. Posted on: 2nd September, 2010. PII S 0025-5718-2010-02410-X-1, in press.

[15] B. Cockburn, J. Gopalakrishnan, F.-J. Sayas, A projection-based error analysis of HDG methods, Math. Comput. 79 (2010) 1351–1367.

[16] B. Cockburn, J. Guzmán, S.-C. Soon, H.K. Stolarski, An analysis of the embedded discontinuous Galerkin method for second-order elliptic problems, SIAM J. Numer. Anal. 47 (2009) 2686–2707.

[17] B. Cockburn, J. Guzmán, H. Wang, Superconvergent discontinuous Galerkin methods for second-order elliptic problems, Math. Comput. 78 (2009) 1–24.

[18] B. Cockburn, G. Kanschat, D. Schötzau, A locally conservative LDG method for the incompressible Navier–Stokes equations, Math. Comput. 74 (2005) 1067–1095.

[19] B. Cockburn, G. Kanschat, D. Schötzau, A note on discontinuous Galerkin divergence-free solutions of the Navier–Stokes equations, J. Sci. Comput. 31 (2007) 61–73.

[20] B. Cockburn, G. Kanschat, D. Schötzau, C. Schwab, Local discontinuous Galerkin methods for the Stokes system, SIAM J. Numer. Anal. 40 (1) (2002) 319–343.

[21] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.

[22] M.K. Denham, M.A. Patrick, Laminar flow over a downstream-facing step in a two-dimensional flow channel, Trans. Inst. Chem. Eng. 52 (1974) 361–367.

[23] M. Fortin, R. Glowinski, Augmented Lagrangian methods, Studies in Mathematics and its Applications, vol. 15, North-Holland Publishing Co., Amsterdam, 1983 (Applications to the numerical solution of boundary value problems. Translated from the French by B. Hunt and D.C. Spicer).

[24] A. Yu. Gelfgat, P.Z. Bar-Yoseph, A.L. Yarin, Stability of multiple steady states of convection in laterally heated cavities, J. Fluid Mech. 388 (1999) 315–334.

[25] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[26] S. Güzey, B. Cockburn, H. Stolarski, The embedded discontinuous Galerkin methods: application to linear shells problems, Int. J. Numer. Meth. Eng. 70 (2007) 757–790.

[27] T.J.R. Hughes, G. Scovazzi, P. Bochev, A. Buffa, A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method, Comput. Meth. Appl. Mech. Eng. 195 (2006) 2761–2787.

[28] O.A. Karakashian, W.N. Jureidini, A nonconforming finite element method for the stationary Navier–Stokes equations, SIAM J. Numer. Anal. 35 (1998) 93–120.

[29] O.A. Karakashian, T. Katsaounis, A discontinuous Galerkin method for the incompressible Navier–Stokes equations, in: B. Cockburn, G.E. Karniadakis, C.-W. Shu (Eds.), Discontinuous Galerkin Methods. Theory, Computation and Applications, Lect. Notes Comput. Sci. Eng., vol. 11, Springer Verlag, Berlin, 2000, pp. 157–166.

[30] L.I.G. Kovasznay, Laminar flow behind two-dimensional grid, Proc. Cambridge Philos. Soc. 44 (1948) 58–62.

[31] R.J. Labeur, G.N. Wells, A Galerkin interface stabilisation method for the advection–diffusion and incompressible Navier–Stokes equations, Comput. Meth. Appl. Mech. Eng. 196 (2007) 4985–5000.

[32] J.-G. Liu, C.-W. Shu, A high-order discontinuous Galerkin method for 2D incompressible flows, J. Comput. Phys. 160 (2000) 577–596.

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

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

[35] J.-C. Nédélec, Mixed finite elements in R^3, Numer. Math. 35 (1980) 315–341.

[36] J.-C. Nédélec, A new family of mixed finite elements in R^3, Numer. Math. 50 (1986) 57–81.

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

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

[39] N.C. Nguyen, J. Peraire, B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow, Comput. Meth. Appl. Mech. Eng. 199 (2010) 582–597.

[40] N.C. Nguyen, J. Peraire, B. Cockburn, A comparison of HDG methods for Stokes flow, J. Sci. Comput. 45 (2010) 215–237.

[41] N.C. Nguyen, J. Peraire, B. Cockburn, Hybridizable discontinuous Galerkin methods, in: Proceedings of the International Conference on Spectral and High Order Methods, Trondheim, Norway, June 2009, Lecture Notes in Computational Science and Engineering, 76 (2011) 63–84.

[42] A.T. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion, J. Comput. Phys. 54 (1984) 468–488.

[43] J. Peraire, N.C. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier–Stokes equations (AIAA Paper 2010-363), in: Proceedings of the 48th AIAA Aerospace Sciences Meeting and Exhibit, Orlando, Florida, January 2010.

[44] 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 (2007) 391–407.

[45] S.-C. Soon, B. Cockburn, H.K. Stolarski, A hybridizable discontinuous Galerkin method for linear elasticity Internat, J. Numer. Meth. Eng. 80 (2009) 1058–1092.

[46] M.P. Ueckermann, P.F.J. Lermusiaux, High order schemes for 2D unsteady biogeochemical ocean models, Ocean Dynam. in press, doi:10.1007/s10236-010-0351-x.

[47] J. Wang, X. Ye, New finite element methods in computational fluid dynamics by *H*(div) elements, SIAM J. Numer. Anal. 45 (2007) 1269–1286.

[48] J. Wang, X. Wang, X. Ye, Finite element methods for the Navier–Stokes equations by *H*(div) elements, J. Comput. Math. 26 (2008) 410–436.

[49] J. Wang, X. Wang, X. Ye, A robust numerical method for Stokes equations based on divergence-free *H*(div) finite element methods, SIAM J. Sci. Comput. 31 (2009) 2784–2802.