48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2010, Orlando, Florida

AIAA 2010-362

A Hybridizable Discontinuous Galerkin Method for the Incompressible Navier-Stokes Equations

N. C. Nguyen* and J. Peraire†
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
B. Cockburn‡
University of Minnesota, Minneapolis, MN 55455, USA

We present a hybridizable discontinuous Galerkin method for the numerical solution the incompressible Navier-Stokes equations. The method is devised by using the discontinuous Galerkin approximation with a special choice of the numerical traces and a fully implicit time-stepping method for temporal discretization. The HDG method possesses several unique features which distinguish themselves 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. Second, it allows for pressure, vorticity and stress boundary conditions to be prescribed on different parts of the boundary. Third, 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 $k \geq 0$ are used for all components of the approximate solution. And fourth, 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 $k \geq 1$ and with order 1 for $k = 0$ in the L^2-norm. We present extensive numerical results to demonstrate the accuracy and convergence properties of the method for a wide range of Reynolds numbers and for various polynomial degrees.

I. Introduction

This paper presents a brief summary of the methodology and numerical results of our recent work[35] on the development of a hybridizable discontinuous Galerkin (HDG) method for the incompressible Navier-Stokes equations. The HDG method presented in Ref. [35] is in turn an extension of our previous work[6,8,9,30–34] on convection-diffusion problems and Stokes flows. The method is devised by using the discontinuous Galerkin approximation with a special choice of the numerical traces (or numerical fluxes) for spatial discretization and a fully implicit time-stepping method for temporal discretization. This results in fully implicit, high-order accurate methods for simulation of incompressible flows. The HDG method not only retains all the well-known advantages of the DG methods over classical continuous Galerkin finite element, finite difference and finite volume methods (see Ref. [4,5,14] and references therein), but also has some attractive features that make them ideally suited for computational fluid dynamics. In particular, the HDG method for the incompressible Navier-Stokes equations have the following distinctive properties:

- **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

*Research Scientist, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Member.
†Professor, Department of Aeronautics and Astronautics, M.I.T., 77 Massachusetts Avenue, AIAA Associate Fellow.
‡Professor, School of Mathematics, University of Minnesota, Minneapolis.

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[16] 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 Stokes[1] and Navier-Stokes equations,[21,22] the family of local DG methods for the Navier-Stokes equations,[11] and the particular cases of those DG methods presented in Ref. [12, 39–41].

- **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 Ref. [9] 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, $\boldsymbol{H}(\mathrm{div})$-conforming, and converges with order $k+2$ for $k \geq 1$. 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. The approach also allows for pressure, vorticity and stress boundary conditions to be prescribed on different parts of the boundary. This is very useful when one would like to impose boundary conditions that are not necessarily compatible with the weak formulation used to define the numerical scheme.

In recent years, several discontinuous Galerkin (DG) methods[2,11,12,21,24,25,27,37,40] have been developed for numerically solving the incompressible Navier-Stokes equations. However, to the best knowledge of the authors, no other known DG method for the incompressible Navier-Stokes equations has all the above four properties of the HDG method. As we pointed out in Ref. [33], although there are DG methods which provide velocities that are divergence-free inside each of the element; however, they do not lie on $\boldsymbol{H}(\mathrm{div})$ since their normal component has no interelement continuity. Examples are the first DG method proposed for the Stokes system[1] and for the Navier-Stokes equations[21] and, more recently, the DG methods for the Stokes equations[26] and for the Navier-Stokes equations.[27] Note also that there are DG methods that do provide velocities that are divergence-free and belong to $\boldsymbol{H}(\mathrm{div})$. A wide family DG methods with this property were introduced in Ref. [11] for the Navier-Stokes equations, even though only a particular case was treated in detail therein. Other particular cases were developed later in Ref. [12, 39–41]; see also the DG method proposed in Ref. [3] for the Stokes equations. However, for these DG methods, their velocities converge with order at most $k+1$ for $k \geq 1$, and their pressure and velocity gradient converge with order at most k. Other DG methods[19,20,24] aim to reduce the globally coupled degrees in a DG discretization to the approximate trace of the field variables. However, unlike the HDG method, the approximate trace in these methods resides in a $C^0(\Omega)$ space. Therefore, none of these DG methods has the local conservativity and superconvergence properties of the HDG method. As a consequence, see Ref. [10], the approximate pressure and velocity gradient of these DG methods converge with order k and hence their approximate solution can not be postprocessed to yield better velocity approximation.

The paper is organized as follows. In Section 2 we introduce the HDG method for numerically solving the incompressible Navier-Stokes equations and extend the method to treat different boundary conditions involving derivatives of the velocity. In this section, we briefly describe the implementation of the HDG method and the local postprocessing proposed in Ref. [9]. In Section 3 we provide numerical results to assess the convergence and accuracy of the method. Finally, in Section 4 we present some concluding remarks.

II. HDG Method for the Incompressible Navier-Stokes Equations

A. Governing Equations

We consider the time-dependent incompressible Navier-Stokes equations

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

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

$$\begin{aligned} \mathbf{L} - \nabla\boldsymbol{u} &= 0, && \text{in } \Omega\times(0,T],\\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla\cdot(-\nu\mathbf{L} + p\mathbf{I} + \boldsymbol{u}\otimes\boldsymbol{u}) &= \boldsymbol{f}, && \text{in } \Omega\times(0,T],\\ \nabla\cdot\boldsymbol{u} &= 0, && \text{in } \Omega\times(0,T],\\ \boldsymbol{u} &= \boldsymbol{g}, && \text{on } \partial\Omega\times(0,T],\\ \boldsymbol{u} &= \boldsymbol{u}_0, && \text{on } \Omega\times\{t=0\}. \end{aligned} \tag{1}$$

where Ω is a polygonal domain in $\mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$, T is the final time, and ν is a kinematic viscosity. Here $\boldsymbol{u}$ 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 $\int_\Omega p = 0$.

To describe the HDG method for solving the above system, we follow the notation used in.[9,32] 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 $K^\pm$, 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.

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 $\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu}\in\boldsymbol{M}_h\ :\ \boldsymbol{\mu} = \mathbf{P}_\partial\boldsymbol{g}\text{ on }\partial\Omega\}$, where $\mathbf{P}_\partial$ 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 $\overline{q}|_{\partial K} = \frac{1}{|\partial K|}\int_{\partial K} q$. Obviously, we have $\overline{q} = q$ 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 $w\,v$ 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 $\mathbf{N}, \mathbf{Z} \in \mathbf{G}_h$. We then 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 $D \subset \mathbb{R}^{d-1}$.

B. Fully Discrete System

We directly discretize the system (1) in time by using the Backward-Euler method and in space by using the HDG method proposed in Ref. [9,32]. In particular, we first compute

$$(\boldsymbol{u}_h^0, \boldsymbol{v})_{\mathcal{T}_h} = (\boldsymbol{u}_0, \boldsymbol{v})_{\mathcal{T}_h}, \quad \forall \boldsymbol{v} \in \boldsymbol{V}_h,$$

at time level $t^0 = 0$. At time level $t^n = \sum_{j=1}^n \Delta t^j$ we seek an approximation $(\mathbf{L}_h^n, \boldsymbol{u}_h^n, p_h^n, \widehat{\boldsymbol{u}}_h^n, \overline{p}_h^n) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\boldsymbol{g}) \times \overline{\Psi}_h$ such that

$$\begin{aligned}
(\mathbf{L}_h^n, \mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h^n, \nabla\cdot\mathbf{G})_{\mathcal{T}_h} - \langle \widehat{\boldsymbol{u}}_h^n, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= 0,\\
\big(\frac{\boldsymbol{u}_h^n}{\Delta t^n}, \boldsymbol{v}\big)_{\mathcal{T}_h} + (\nu\mathbf{L}_h^n - p_h^n\mathbf{I} - \boldsymbol{u}_h^n\otimes\boldsymbol{u}_h^n, \nabla\boldsymbol{v})_{\mathcal{T}_h} &\\
+ \langle(-\nu\mathbf{L}_h^n + p_h^n\mathbf{I} + \widehat{\boldsymbol{u}}_h^n\otimes\widehat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n), \boldsymbol{v}\rangle_{\partial\mathcal{T}_h} &= (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h} + \big(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t^n}, \boldsymbol{v}\big)_{\mathcal{T}_h},\\
-(\boldsymbol{u}_h^n, \nabla q)_{\mathcal{T}_h} + \langle \widehat{\boldsymbol{u}}_h^n\cdot\boldsymbol{n}, q - \overline{q}\rangle_{\partial\mathcal{T}_h} &= 0,\\
\big\langle \overline{p}_h^n - \overline{p}_h^n, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= 0,\\
\langle(-\nu\mathbf{L}_h^n + p_h^n\mathbf{I} + \widehat{\boldsymbol{u}}_h^n\otimes\widehat{\boldsymbol{u}}_h^n)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n), \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h} &= 0,\\
\big\langle \widehat{\boldsymbol{u}}_h^n\cdot\boldsymbol{n}, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= 0,\\
(p_h^n, 1)_{\mathcal{T}_h} &= 0,
\end{aligned} \tag{2}$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}, \overline{\psi}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0}) \times \overline{\Psi}$. Here $\boldsymbol{s}_h(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n)$ is the *stabilization vector-valued function* which has the form

$$\boldsymbol{s}_h(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n) = \mathbf{S}(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n)(\boldsymbol{u}_h^n - \widehat{\boldsymbol{u}}_h^n), \tag{3}$$

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

The choice of the stabilization tensor $\mathbf{S}(\boldsymbol{u}_h^n, \widehat{\boldsymbol{u}}_h^n)$ is crucial since it does have an important effect on both the stability and accuracy of the method. Following Ref. [30,31], we select the stabilization tensor $\mathbf{S}$ of 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

$$\tau \approx \frac{\nu}{\ell} + |\boldsymbol{u}_h^n|. \tag{4}$$

Here ℓ is a representative length scale and $|\boldsymbol{u}_h^n|$ is the magnitude of the approximate velocity. This choice is based on dimensional analysis which requires that τ has the same dimension unit as the velocity and the ratio ν/ℓ.

We note that using higher-order backward different formulas (BDF) or diagonally implicit Runge-Kutta methods would yield a discrete system similar to (2). 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 (1). In fact, we employ the second-order and third-order BDF schemes to discretize the time derivative for our numerical examples presented in Section 3.

C. Implementation

In the remainder of this section, we shall drop the superscript n in the approximate solution to simplify the notation. We consider the Newton-Raphson method for solving the nonlinear system (2). Given the m^{th} current iterate $(\mathbf{L}_h^m, \boldsymbol{u}_h^m, p_h^m, \widehat{\boldsymbol{u}}_h^m, \overline{\rho}_h^m)$, we find an increment $(\delta\mathbf{L}_h^m, \delta\boldsymbol{u}_h^m, \delta p_h^m, \delta\widehat{\boldsymbol{u}}_h^m, \delta\overline{\rho}_h^m) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0}) \times \overline{\Psi}_h$ 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\widehat{\boldsymbol{u}}_h^m, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= r^1(\mathbf{G}),\\
\big(\frac{\delta\boldsymbol{u}_h^m}{\Delta t}, \boldsymbol{v}\big)_{\mathcal{T}_h} + (\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}&\\
+\langle(-\nu\delta\mathbf{L}_h^m + \delta p_h^m\mathbf{I} + \delta\widehat{\boldsymbol{u}}_h^m\otimes\widehat{\boldsymbol{u}}_h^m + \widehat{\boldsymbol{u}}_h^m\otimes\delta\widehat{\boldsymbol{u}}_h^m)\boldsymbol{n}, \boldsymbol{v}\rangle_{\partial\mathcal{T}_h}&\\
+\langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \widehat{\boldsymbol{u}}_h^m)\delta\boldsymbol{u}_h^m + \partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \widehat{\boldsymbol{u}}_h^m)\delta\widehat{\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\widehat{\boldsymbol{u}}_h^m\cdot\boldsymbol{n}, q\rangle_{\partial\mathcal{T}_h} &= r^3(q),\\
\big\langle\delta\overline{p}_h^m - \delta\overline{\rho}_h^m, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= r^4(\overline{\psi}),\\
\langle(-\nu\delta\mathbf{L}_h^m + \delta p_h^m\mathbf{I} + \delta\widehat{\boldsymbol{u}}_h^m\otimes\widehat{\boldsymbol{u}}_h^m + \widehat{\boldsymbol{u}}_h^m\otimes\delta\widehat{\boldsymbol{u}}_h^m)\boldsymbol{n}\rangle_{\partial\mathcal{T}_h}&\\
+\langle\partial_1\boldsymbol{s}_h(\boldsymbol{u}_h^m, \widehat{\boldsymbol{u}}_h^m)\delta\boldsymbol{u}_h^m + \partial_2\boldsymbol{s}_h(\boldsymbol{u}_h^m, \widehat{\boldsymbol{u}}_h^m)\delta\widehat{\boldsymbol{u}}_h^m, \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h} &= r^5(\boldsymbol{\mu}),\\
\big\langle\delta\widehat{\boldsymbol{u}}_h^m\cdot\boldsymbol{n}, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= r^6(\overline{\psi}),\\
(\delta p_h^m, 1)_{\mathcal{T}_h} &= 0,
\end{aligned}
\tag{5}
$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}, \overline{\psi}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0}) \times \overline{\Psi}$, where the right-hand side residuals are obtained by substituting the current iterate into (2). 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.

At first sight, the system (5) appears very expensive to solve since it involves too many unknowns. However, by using the hybridization technique proposed in Ref. [32] we can reduce it to solving a global linear system in terms of $(\delta\widehat{\boldsymbol{u}}_h^m, \delta\overline{\rho}_h^m)$ only. In particular, the first four equations of (5) define the local solver which can be used to eliminate $(\delta\mathbf{L}_h^m, \delta\boldsymbol{u}_h^m, \delta p_h^m)$ by inserting them into the last three equations of (5). This results in the reduced system in terms of $(\delta\widehat{\boldsymbol{u}}_h^m, \delta\overline{\rho}_h^m)$. Once $(\delta\widehat{\boldsymbol{u}}_h^m, \delta\overline{\rho}_h^m)$ is obtained by solving the global reduced system, $(\delta\mathbf{L}_h^m, \delta\boldsymbol{u}_h^m, \delta p_h^m)$ can be computed inexpensively in an element-by-element fashion. Since $\delta\widehat{\boldsymbol{u}}_h^m$ is defined on the element faces and $\delta\overline{\rho}_h^m$ has one degree of freedom per element, the HDG method reduces significantly the number of the globally coupled unknowns. In practice, we implement the HDG method by using the augmented Lagrangian approach[16] which involves a sub-iteration of solving for the degrees of freedom of $\delta\widehat{\boldsymbol{u}}_h^m$ only. We refer to Ref. [32] for detailed implementation aspects of the HDG method.

D. Local Postprocessing

We apply the element-by-element postprocessing proposed in Ref. [9] to obtain a new approximate velocity which is exactly divergence-free and $\boldsymbol{H}(div)$-conforming. It is important to note that we need to perform the local postprocessing only at those time levels for which an enhanced velocity approximation is desired.

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

$$
\langle(\boldsymbol{u}_h^\star - \widehat{\boldsymbol{u}}_h)\cdot\boldsymbol{n}, \mu\rangle_F = 0 \quad \forall\,\mu \in \mathcal{P}_k(F), \tag{6a}
$$

$$
\langle(\boldsymbol{n}\times\nabla)(\boldsymbol{u}_h^\star\cdot\boldsymbol{n}) - \boldsymbol{n}\times(\,\{\!\{\mathbf{L}_h^t\}\!\}\boldsymbol{n}), (\boldsymbol{n}\times\nabla)\mu\rangle_F = 0 \quad \forall\,\mu \in \mathcal{P}_{k+1}(F)^\perp, \tag{6b}
$$

for all faces F of K, and such that

$$
(\boldsymbol{u}_h^\star - \boldsymbol{u}_h, \nabla w)_K = 0 \quad \forall\, w \in \mathcal{P}_k(K), \tag{6c}
$$

$$
(\nabla\times\boldsymbol{u}_h^\star - \mathbf{w}_h, (\nabla\times\boldsymbol{v})\,\mathbf{B}_K)_K = 0 \quad \forall\,\boldsymbol{v} \in \boldsymbol{\mathcal{S}}_k(K). \tag{6d}
$$

In (6b),

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

$\mathbf{n}\times\nabla$ 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 (6d),

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

is the approximation to the vorticity and $\mathbf{B}_K$ is the so-called *symmetric bubble matrix* introduced in,[7] 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, $\boldsymbol{\mathcal{S}}_k(K) := \sum_{\ell=1}^{k} \boldsymbol{S}_\ell(K)$ where $\boldsymbol{S}_\ell$ is the space of vector-valued homogeneous polynomials $\boldsymbol{v}$ of degree ℓ such that $\boldsymbol{v}\cdot\boldsymbol{x} = 0$.[28, 29]

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 $n_1 a_2 - n_2 a_1$, if $\nabla\times\boldsymbol{u}$ is replaced by $\nabla\times\boldsymbol{u} := \partial_1 u_2 - \partial_2 u_1$, and if equation (6d) is replaced by

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

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

E. Other Boundary Conditions

We end this section by extending the method to treat boundary conditions of the form

$$\mathbf{B}\boldsymbol{n} = \boldsymbol{g}_N, \quad \text{on } \partial\Omega_N, \tag{7}$$

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 third and fourth examples in Table 1 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

$$\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu} \in \boldsymbol{M}_h \ : \ \boldsymbol{\mu} = \mathbf{P}_\partial \boldsymbol{g} \text{ on } \partial\Omega_D\}, \tag{8a}$$

and, for the fourth example in Table 1, as

$$\boldsymbol{M}_h(\boldsymbol{g}) = \{\boldsymbol{\mu} \in \boldsymbol{M}_h \ : \ \boldsymbol{\mu} = \mathbf{P}_{\partial\Omega_D}\boldsymbol{g} \text{ on } \partial\Omega_D \text{ and } \boldsymbol{\mu}\cdot\boldsymbol{n} = \mathsf{P}_{\partial\Omega_N}\boldsymbol{g}\cdot\boldsymbol{n} \text{ on } \partial\Omega_N\}, \tag{8b}$$

where $\mathbf{P}_{\partial\Omega_D}$ and $\mathsf{P}_{\partial\Omega_N}$ denote the L^2 projections on $\partial\Omega_D$ and $\partial\Omega_N$, respectively.

We then seek an approximation $(\mathbf{L}_h, \boldsymbol{u}_h, p_h, \widehat{\boldsymbol{u}}_h, \overline{\rho}_h) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\boldsymbol{g}) \times \overline{\Psi}_h$ such that

$$\begin{aligned}
(\mathbf{L}_h, \mathbf{G})_{\mathcal{T}_h} + (\boldsymbol{u}_h, \nabla\cdot\,\mathbf{G})_{\mathcal{T}_h} - \langle\widehat{\boldsymbol{u}}_h, \mathbf{G}\boldsymbol{n}\rangle_{\partial\mathcal{T}_h} &= 0,\\
\big(\frac{\boldsymbol{u}_h}{\Delta t}, \boldsymbol{v}\big)_{\mathcal{T}_h} + (\nu\mathbf{L}_h - p_h\mathbf{I} - \boldsymbol{u}_h\otimes\boldsymbol{u}_h, \nabla\boldsymbol{v})_{\mathcal{T}_h} &\\
+ \langle(-\nu\mathbf{L}_h + p_h\mathbf{I} + \widehat{\boldsymbol{u}}_h\otimes\widehat{\boldsymbol{u}}_h)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h), \boldsymbol{v}\rangle_{\partial\mathcal{T}_h} &= (\boldsymbol{f}, \boldsymbol{v})_{\mathcal{T}_h} + \big(\frac{\boldsymbol{u}_h^{n-1}}{\Delta t}, \boldsymbol{v}\big)_{\mathcal{T}_h},\\
-(\boldsymbol{u}_h, \nabla q)_{\mathcal{T}_h} + \langle\widehat{\boldsymbol{u}}_h\cdot\boldsymbol{n}, q - \overline{q}\rangle_{\partial\mathcal{T}_h} &= 0,\\
\big\langle\overline{p}_h - \overline{\rho}_h, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= 0,\\
\langle(-\nu\mathbf{L}_h + p_h\mathbf{I} + \widehat{\boldsymbol{u}}_h\otimes\widehat{\boldsymbol{u}}_h)\boldsymbol{n} + \boldsymbol{s}_h(\boldsymbol{u}_h, \widehat{\boldsymbol{u}}_h), \boldsymbol{\mu}\rangle_{\partial\mathcal{T}_h} + \Big\langle\widehat{\mathbf{B}}_h\boldsymbol{n}, \boldsymbol{\mu}\Big\rangle_{\partial\Omega_N} &= \langle\boldsymbol{g}_N, \boldsymbol{\mu}\rangle_{\partial\Omega_N},\\
\big\langle\widehat{\boldsymbol{u}}_h\cdot\boldsymbol{n}, \overline{\psi}\big\rangle_{\partial\mathcal{T}_h} &= 0,\\
(p_h, 1)_{\mathcal{T}_h} &= 0,
\end{aligned} \tag{9}$$

for all $(\mathbf{G}, \boldsymbol{v}, q, \boldsymbol{\mu}, \overline{\psi}) \in \mathbf{G}_h \times \boldsymbol{V}_h \times P_h \times \boldsymbol{M}_h(\mathbf{0}) \times \overline{\Psi}$. Here $\widehat{\mathbf{B}}_h$ is an approximate trace operator shown in Table 1. Recall that the superscript n was dropped for the sake of notation simplification.

Condition Type	$\mathbf{B}$	$\widehat{\mathbf{B}}_h$
stress+pressure	$-\nu(\mathbf{L}+\mathbf{L}^t)+p\mathbf{I}$	$-\nu(\mathbf{L}_h+\mathbf{L}_h^t)+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
stress*	$-\nu(\mathbf{L}+\mathbf{L}^t)$	$-\nu(\mathbf{L}_h+\mathbf{L}_h^t)+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
vorticity+pressure	$-\nu(\mathbf{L}-\mathbf{L}^t)+p\mathbf{I}$	$-\nu(\mathbf{L}_h-\mathbf{L}_h^t)+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
vorticity*,†	$-\nu(\mathbf{L}-\mathbf{L}^t)$	$-\nu(\mathbf{L}_h-\mathbf{L}_h^t)+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
gradient+pressure	$-\nu\mathbf{L}+p\mathbf{I}$	$-\nu\mathbf{L}_h+p_h\mathbf{I}+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$
gradient*	$-\nu\mathbf{L}$	$-\nu\mathbf{L}_h+\boldsymbol{s}_h(\boldsymbol{u}_h,\widehat{\boldsymbol{u}}_h)\otimes\boldsymbol{n}$

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.

III. Numerical Results

We present numerical results to assess the performance of the HDG method for several test cases including the Taylor vortex flow,[37] lid-driven cavity flow,[18] channel expansion flow,[36] and natural convective flow in a cavity.[17]

A. Taylor Vortex Problem

The Taylor vortex problem[37] is a well-known example of the unsteady incompressible Navier-Stokes equations. We use this example to examine the convergence properties of the HDG method since the exact solution is available as

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

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 $t=0$.

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 2 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 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 $\boldsymbol{H}(\mathrm{div})$-conforming velocity field. To visualize the effect of the local postprocessing, we show in Figure 1 the plots of the approximate 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.

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

Table 2. History of convergence of the HDG method for the Taylor vortex problem when the Dirichlet condition is applied on the entire boundary.

B. 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.

Figure 2 shows a computational mesh 32×32 on which our numerical solutions are obtained for Reynolds numbers $Re = 1000$ and $Re = 5000$. The grid is refined and thus dense along the wall in order to capture the boundary layer and the complex fluid structure near the wall. The stabilization parameter $\tau = 1$ is used for all cases.

We present the profiles of the velocity along the centerlines for $Re = 1000$ and $Re = 5000$ in Figure 3. 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 $x = 1$. We further show in Figure 4 the streamline contours of the postprocessed velocity $\boldsymbol{u}_h^*$ for $Re = 1000$ and $Re = 5000$. We clearly observe the typical structure of the steady-state solution for the lid-driven cavity flow: 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 coarser than the meshes (129×129 and 257×257) used in the previous calculation.[18] We note however that our local postprocessing is not as effective as it was in the previous example in improving the accuracy of the velocity approximation. This is due to the presence of singularities at the corners as well as to the relatively high Reynolds numbers.

C. 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[36] in order to compare our results with those presented in.[36] The channel geometry and computational mesh are shown in Figure 5. The governing equations are given by

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

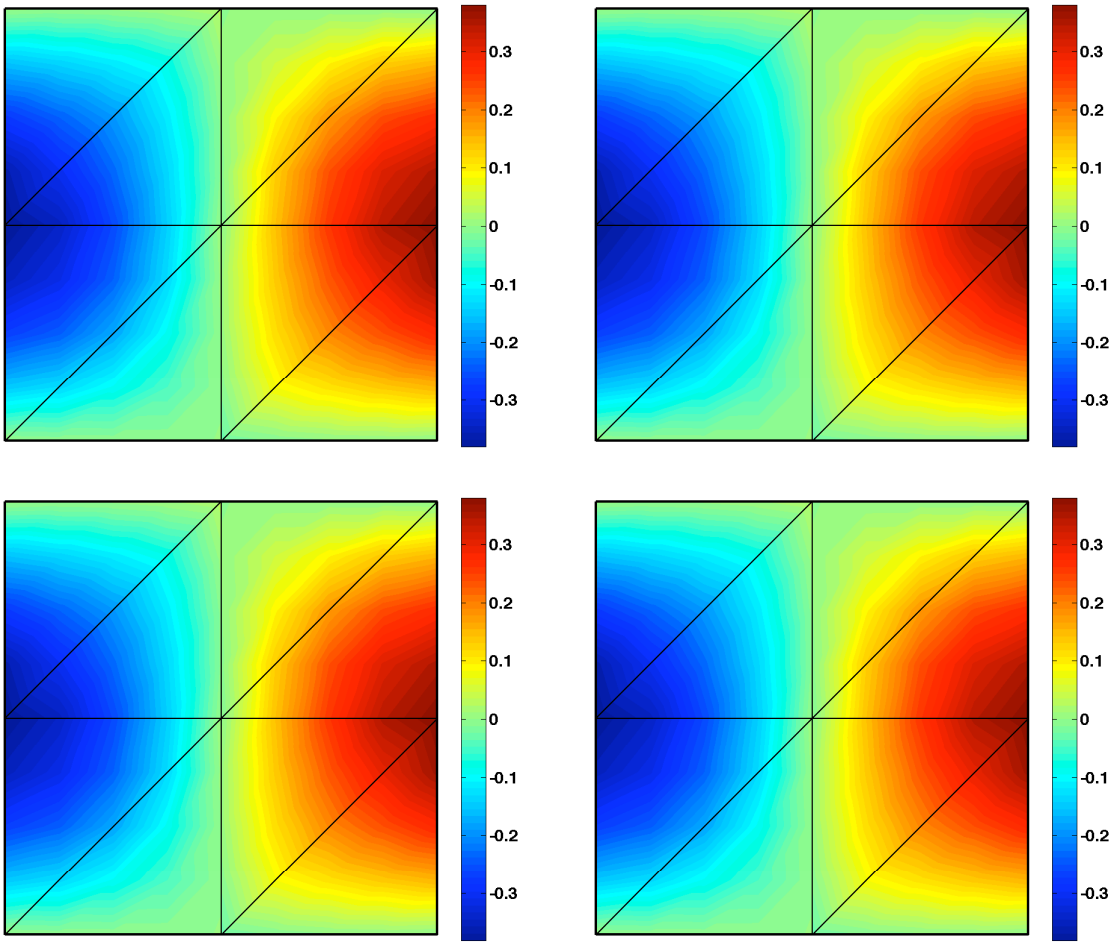

Figure 1. The approximate velocity $\boldsymbol{u}_h$ (left) and the postprocessed velocity $\boldsymbol{u}_h^*$ (right) obtained using $k = 2$ on the grid $h = 1/2$ for the Taylor vortex problem. The horizontal velocity is placed at the top, while the vertical velocity at the bottom.

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 $Re = U_0 H/\nu$. 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 Figure 6 the contour of the approximate pressure and its zoom near the step corner for $k = 2$ and $k = 3$, and in Figure 7 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.[36] 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.[15] In Figure 8 we display the streamwise velocity profiles at several locations downstream of the step. The agreement with[36] is very good.

D. Natural Convection in a Rectangular Cavity

Finally, we consider the natural convective flow in a laterally heated cavity presented in.[17] The geometry and boundary conditions are depicted in Figure 9. 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

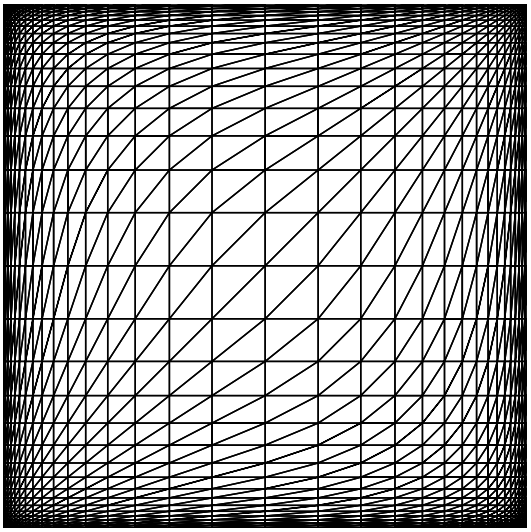

Figure 2. Finite element mesh for the lid-driven cavity flow.

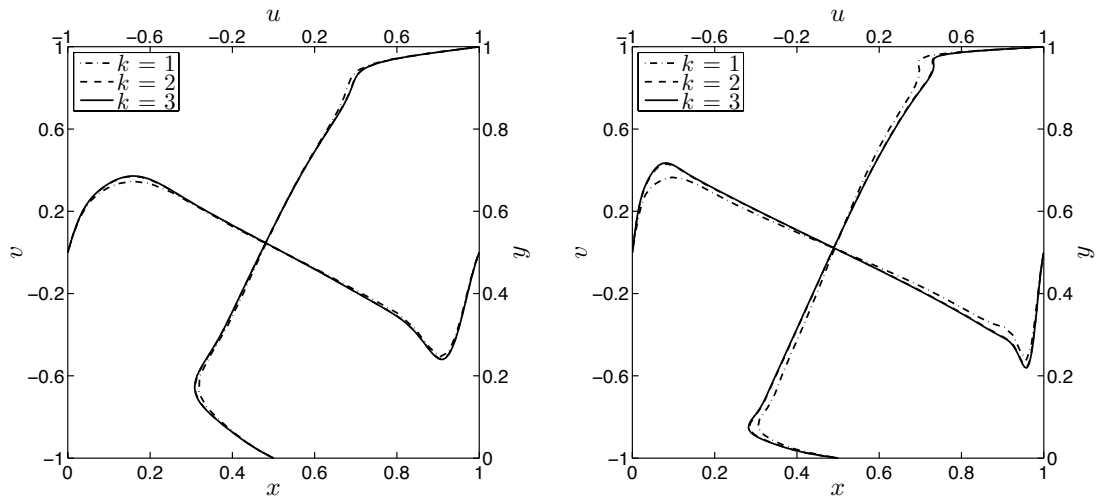

Figure 3. The u- and v- components of the approximate velocity $\boldsymbol{u}_h$ along the horizontal and vertical centerlines for $Re = 1000$ (left) and $Re = 5000$ (right) on the grid 32×32 for the lid-driven cavity flow.

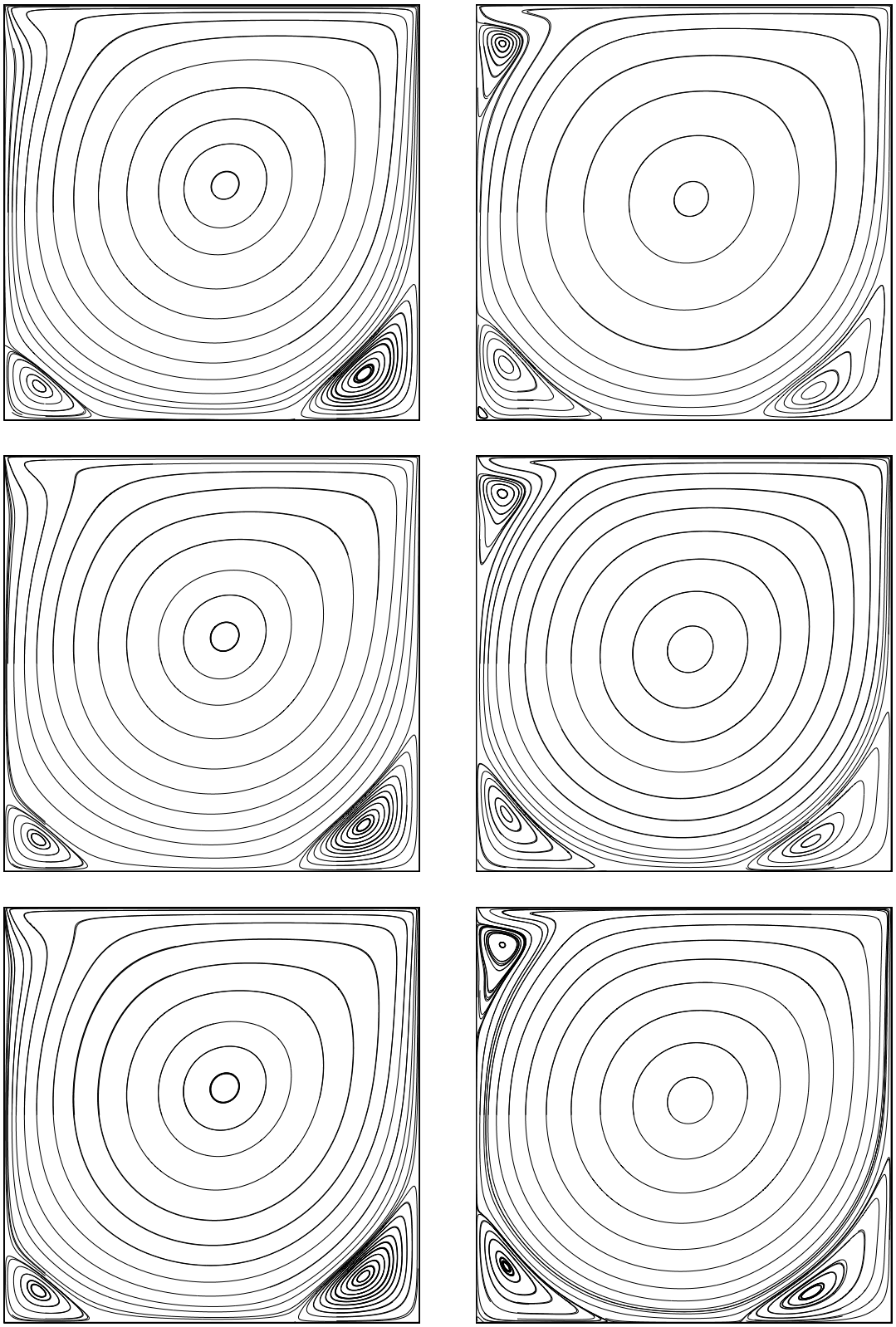

Figure 4. Streamline of the postprocessed velocity for $Re = 1000$ (left) and $Re = 5000$ (right) using $k = 1$ (top), $k = 2$ (middle), and $k = 3$ (bottom) on the grid 32×32 for the lid-driven cavity flow.

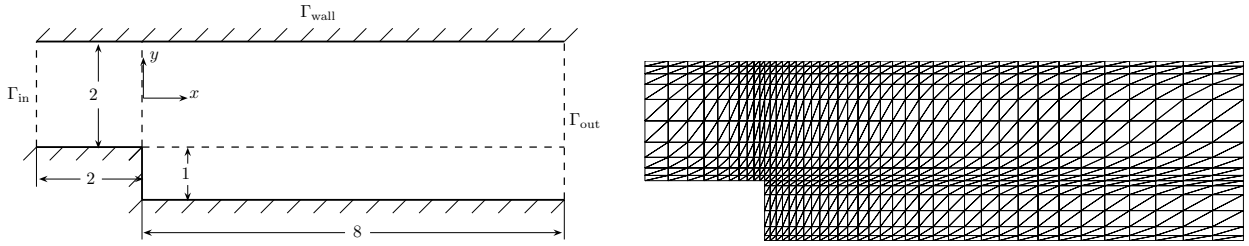

Figure 5. Geometry configuration and computational mesh for the channel expansion.

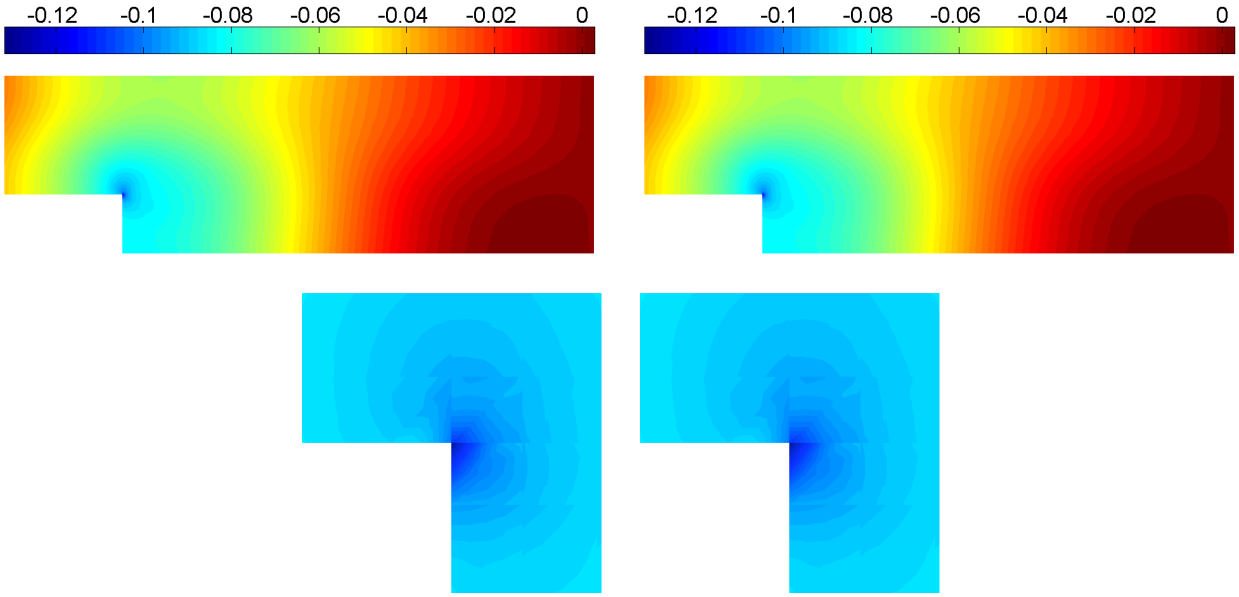

Figure 6. Plots of the approximate pressure and its zoom near the step corner for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow. It is clearly seen that the result for $k = 3$ is better than that for $k = 2$.

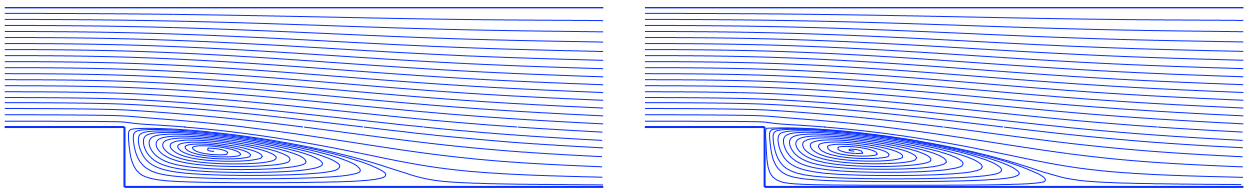

Figure 7. Plots of the postprocessed streamline for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow.

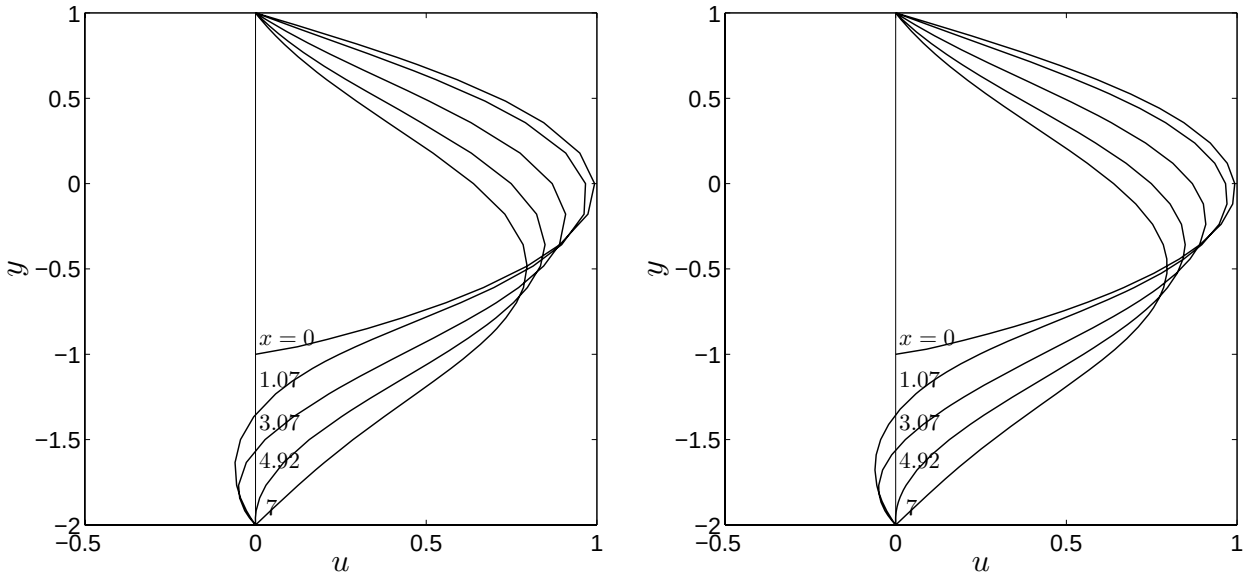

Figure 8. A plot of the streamwise velocity profiles for $k = 2$ (left) and $k = 3$ (right) for the channel expansion flow.

equations are given by

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

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 (4). Furthermore, we use the second-order backward difference formula (BDF2) for the temporal discretization and take the timestep to be $\Delta t = 2 \times 10^{-4}$. The finite element mesh shown in Figure 9 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 Figure 10 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. Figure 11 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 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.[17]

IV. 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 approxiamtion 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

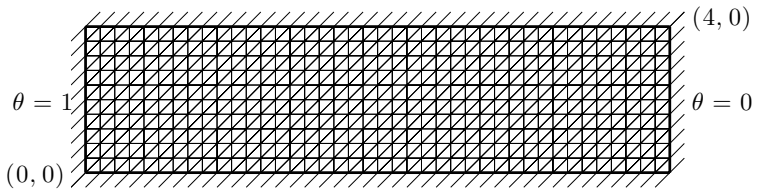

Figure 9. Geometry, boundary conditions, and finite element mesh for the natural convective flow in a laterally heated cavity.

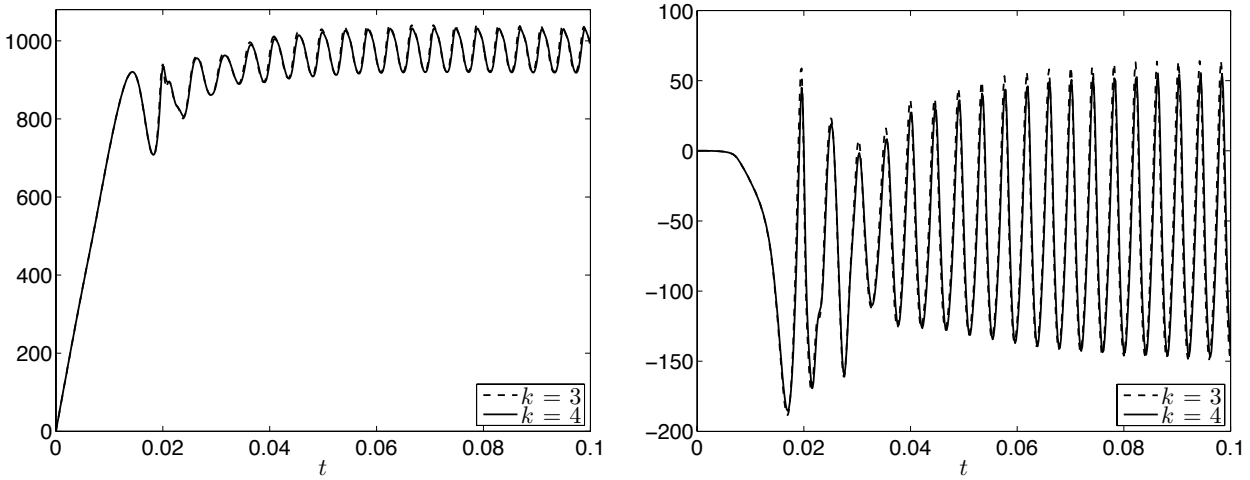

Figure 10. The postprocessed approximate velocity for $k = 3, 4$ at the spatial point $(2, 0.8)$ as a function of t for the natural convective flow. The horizontal velocity is shown on the left, and the vertical velocity on the right.

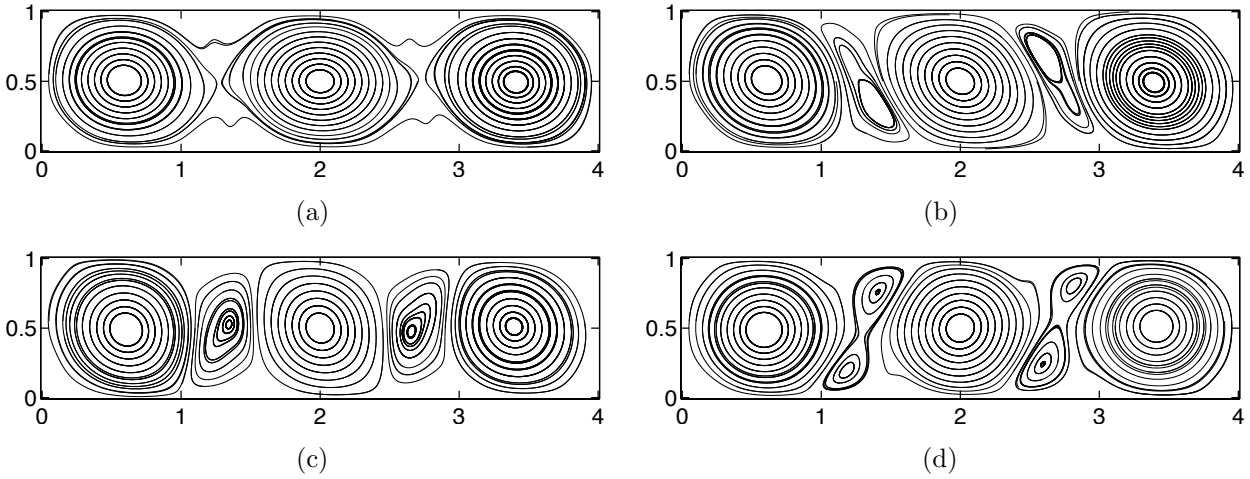

Figure 11. Streamline of the postprocessed approximate velocity for $k = 4$ at time $t = 0.095$ (a), $t = 0.096$ (b), $t = 0.097$(c), and $t = 0.098$ (d) for the natural convective flow.

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.

Acknowledgments

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

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

[3]J. Carrero, B. Cockburn, and D. Schötzau. Hybridized globally divergence-free LDG methods. I. The Stokes problem. *Math. Comp.*, 75:533–563, 2006.

[4]B. Cockburn. Discontinuous Galerkin methods. *ZAMM Z. Angew. Math. Mech.*, 83:731–754, 2003.

[5]B. Cockburn. Discontinuous Galerkin Methods for Computational Fluid Dynamics. In R. de Borst E. Stein and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 3, pages 91–123. John Wiley & Sons, Ltd., England, 2004.

[6]B. Cockburn and J. Gopalakrishnan. The derivation of hybridizable discontinuous Galerkin methods for Stokes flow. *SIAM J. Numer. Anal.*, 47:1092–1125, 2009.

[7]B. Cockburn, J. Gopalakrishnan, and J. Guzmán, *A new elasticity element made for enforcing weak stress symmetry. Math. Comp.*. To appear.

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

[9]B. Cockburn, J. Gopalakrishnan, N. C. Nguyen, J. Peraire, and F-J Sayas. Analysis of an HDG method for Stokes flow. *Math. Comp.*. To appear.

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

[11]B. Cockburn, G. Kanschat, and D. Schötzau. A locally conservative LDG method for the incompressible Navier-Stokes equations. *Math. Comp.*, 74:1067–1095, 2005.

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

[13]B. Cockburn, G. Kanschat, D. Schötzau, and C. Schwab. Local Discontinuous Galerkin Methods for the Stokes System. *SIAM J. Numer. Anal.*, 40(1):319–343, 2002.

[14]B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.*, 16:173–261, 2001.

[15]M. K. Denham, and M. A. Patrick. Laminar flow over a downstream-facing step in a two-dimensional flow channel. *Trans. Inst. Chemical Engineers*, 52:361–367, 1974.

[16]M. Fortin, and R. Glowinski. *Augmented Lagrangian methods*, volume 15 of *Studies in Mathematics and its Applications*. 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.

[17]A. Yu. Gelfgat, P. Z. Bar-Yoseph, and A. L. Yarin. Stability of multiple steady states of convection in laterally heated cavities. *Journal of Fluid Mechanics*, 388:315–334, 1999.

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

[19]S. Güzey, B. Cockburn, and H. Stolarski. The embedded discontinuous Galerkin methods: Application to linear shells problems, *Internat. J. Numer. Methods Engrg.*, 70 (2007), pp. 757-790.

[20]T. J. R. Hughes, G. Scovazzi, P. Bochev, and A. Buffa. A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method. *Comput. Meth. Appl. Mech. Engrg.*, 195:2761–2787, 2006.

[21]O. A. Karakashian and W.N. Jureidini. A nonconforming finite element method for the stationary Navier-Stokes equations. *SIAM J. Numer. Anal.*, 35:93–120, 1998.

[22]O. A. Karakashian and T. Katsaounis. A discontinuous Galerkin method for the incompressible Navier-Stokes equations. In B. Cockburn, G.E. Karniadakis, and C.-W. Shu, editors, *Discontinuous Galerkin Methods. Theory, Computation and Applications*, volume 11 of *Lect. Notes Comput. Sci. Engrg.*, pages 157–166, Berlin, February 2000. Springer Verlag.

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

[24]R. J. Labeur, and G. N. Wells. A Galerkin interface stabilisation method for the advection–diffusion and incompressible Navier-Stokes equations. *Comput. Meth. Appl. Mech. Engrg.*, 196:4985–5000, 2007.

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

[26]A. Montlaur, S. Fernández-Méndez and A. Huerta Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations *Internat. J. Numer. Methods Fluids*, 57:1071–1092, 2008.

[27]A. Montlaur, S. Fernández-Méndez, J. Peraire and A. Huerta Discontinuous Galerkin methods for the Navier-Stokes equations using solenoidal approximations *Internat. J. Numer. Methods Fluids*. To Appear.

[28]J.-C. Nédélec. Mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 35:315–341, 1980.

[29]J.-C. Nédélec. A new family of mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 50:57–81, 1986.

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

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

[32]N.C. Nguyen, J. Peraire, and B. Cockburn, A hybridizable discontinuous Galerkin method for Stokes flow. *Computer Methods in Applied Mechanics and Engineering*. To appear.

[33]N.C. Nguyen, J. Peraire, and B. Cockburn, A comparison of HDG methods for Stokes flow. Submitted.

[34]N.C. Nguyen, J. Peraire, and B. Cockburn, Hybridizable discontinuous Galerkin methods. Submitted.

[35]N.C. Nguyen, J. Peraire, and B. Cockburn, A hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. Submitted.

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

[37]K. Shahbazi, P. F Fischer, and C. R. Ethier. A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations. *J. Comput. Phys.*, 222:391–407, 2007.

[38]S.J. Sherwin, R.M. Kirby, J. Peiró, R.L. Taylor, and O.C. Zienkiewicz. *On 2D elliptic discontinuous Galerkin methods*, Internat. J. Numer. Methods Engrg. **65** (2006), no. 5, 752–784.

[39]J. Wang and X. Ye. New finite element methods in computational fluid dynamics by $H(\mathrm{div})$ elements *SIAM J. Numer. Anal.*, 45:1269–1286, 2007.

[40]J. Wang, X. Wang and X. Ye. Finite element methods for the Navier-Stokes equations by $H(\mathrm{div})$ elements *J. Comput. Math.*, 26:410–436, 2008.

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