Abstract This chapter is intended to be a didactical introduction to the Hybridizable Discontinuous Galerkin (HDG) method, including the formulation and its implementation. The Laplace and Stokes equations are considered as representative problems with self-adjoint operators, accounting for the incompressibility in the second one.

1 Introduction

Even though the Hybridizable Discontinuous Galerkin (HDG) method is a novel method proposed just a few years ago (see Cockburn et al. (2009, 2008)), it has nowadays been successfully applied to all kinds of problems, especially in the field of Computational Fluid Dynamics (CFD); see, for instance, Cockburn et al. (2011); Nguyen et al. (2010, 2011) for its application to the Stokes and Navier–Stokes equations, or Kirby et al. (2011); Giorgiani et al. (2013b); Huerta et al. (2013) for an efficiency study in front of Continuous Finite Elements (CFE) in the context of elliptic problems and wave problems.

HDG inherits all the advantages of high-order Discontinuous Galerkin (DG) methods (see for instance Cockburn (2004); Hesthaven and Warburton (2002); Peraire and Persson (2008); Montlaur et al. (2008)) that have made them so popular in CFD in the last decade, such as local conservation of quantities of interest, intrinsic stabilization thanks to a proper definition of numerical fluxes at element boundaries, suitability for code vectorization and parallel computation, and suitability for adaptivity.

But, HDG outperforms other DG methods for problems involving self-adjoint operators, due to two main peculiarities: hybridization and superconvergence properties. The hybridization process drastically reduces the number of degrees of freedom in the discrete problem, similarly to static condensation in the context of high-order CFE, see for instance Giorgiani et al. (2013b). More precisely, in a Laplace equation the unknowns reduce to the approximation of the trace of the solution at the mesh skeleton, i.e. the sides (or faces in 3D) of the mesh; and in incompressible flow problems, the final unknowns correspond to just the trace of...
the velocity at the mesh skeleton plus one scalar representing the mean of the pressure at every element. On other hand, HDG is based on a mixed formulation that, differently to CFE or other DG methods, is stable even when all variables (primal unknowns and derivatives) are approximated with polynomials of the same degree $k$. Consequently, convergence of order $k + 1$ in $L^2$ norm is proved not only for the primal unknown, but also for its derivatives. In addition, a simple element-by-element postprocess of the derivatives leads to a superconvergent approximation of the primal variables, with convergence of order $k + 2$ in $L^2$ norm. The superconvergent solution can also be used to compute an efficient error estimator and define an adaptivity procedure as proposed by Giorgiani et al. (2013a, 2014).

This document presents an introduction to HDG methods. The presentation aims to be didactical (including implementation) more than exhaustive, thus it does not cover completely the current state-of-the art of HDG methods and its applications. The Laplace and Stokes equations are considered as representative problems with self-adjoint operators, accounting for the incompressibility in the second one.

## 2 Laplace equation

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain with boundary $\partial \Omega$. The following problem is considered,

$$
\begin{align*}
-\nabla \cdot (\nu \nabla u) &= f & \text{in } \Omega \\
-\nu \nabla u \cdot n &= g & \text{on } \Gamma_N \\
u &= u_D & \text{on } \Gamma_D
\end{align*}
$$

where $u$ is the solution, $\nu$ is a material coefficient, $f$ is a given source term, $u_D$ are prescribed values on the Dirichlet boundary $\Gamma_D$, and $g$ is a prescribed flux on the Neumann boundary $\Gamma_N$, with $\Gamma_D \cup \Gamma_N = \partial \Omega$.

The domain $\Omega$ is now assumed to be split by a finite element mesh with $n_{\text{el}}$ disjoint elements $K_i$, such that

$$
\Omega \subset \bigcup_{i=1}^{n_{\text{el}}} K_i,
$$

$q = K_i \cap K_j = \emptyset$ for $i \neq j$.

The union of all $n_{\text{fc}}$ faces $\Gamma_i$ (sides for 2D) is denoted as

$$
\Gamma := \bigcup_{i=1}^{n_{\text{el}}} \partial K_i = \bigcup_{f=1}^{n_{\text{fc}}} \Gamma_f.
$$

The key idea of the HDG method is introducing a new unknow, $\hat{u}$, corresponding to the solution $u$ on the mesh skeleton $\Gamma$, see Figure 1.
Figure 1. The left picture shows a representation of an HDG discretization on a finite element mesh. Elemental variables ($u$ and $q$) are element-by-element approximated from nodal values (black dots). Trace variables ($\hat{u}$) are approximated on the mesh skeleton $\Gamma$ in red. The right picture shows a representation of the local problem: a pure Dirichlet problem taking $\hat{u}$ as data on the boundary of the element (in red). It is solved in each element to express the elemental variables, $u$ and $q$, in black, in terms of the trace $\hat{u}$.

The new unknown $\hat{u}$, which is usually referred to as trace variable, allows stating a Dirichlet problem in each element, that is, the so-called local problems,

\[
\begin{align*}
\nabla \cdot q &= f \quad \text{in } K_i, \\
q + \nu \nabla u &= 0 \quad \text{in } K_i, \\
u &= \hat{u} \quad \text{on } \partial K_i
\end{align*}
\]

for $i = 1, \ldots, n_{el}$.

(2)

The solution of the local problem, in each element, allows expressing $u$, and also the flux $q$, in terms of the trace variable $\hat{u}$. Thus, now the actual unknown of the problem is the trace $\hat{u}$. It is then determined, closing the problem, imposing the conservativity conditions,

\[
[q \cdot n] = 0 \quad \text{on } \Gamma \setminus \partial \Omega, \quad q \cdot n = g \quad \text{on } \Gamma_N,
\]

(3)

and the Dirichlet boundary conditions

\[
\hat{u} = \mathbb{P}_2(u_D) \quad \text{on } \Gamma_D,
\]

(4)

where $\mathbb{P}_2(u_D)$ is the $L_2$ projection of the Dirichlet data $u_D$ onto the finite element space on the boundary. The jump $[\cdot]$ operator is defined at a face $\Gamma_f$ as

\[
[\hat{\circledast}] = \hat{\circledast}_{L(f)} + \hat{\circledast}_{R(f)} \quad \text{on } \Gamma_f,
\]

where $R(f)$ and $L(f)$ are the numbers of the left and right elements sharing the face, that is, $\Gamma_f = K_{L(f)} \cap K_{R(f)}$, and the subindex $\hat{\circledast}$ denotes the value of function $\hat{\circledast}$ from element $K_i$. In particular, $[q \cdot n] = q_{L(f)} \cdot n_{L(f)} + q_{R(f)} \cdot n_{R(f)} = (q_{L(f)} - q_{R(f)}) \cdot n_{L(f)}$.
It is important noting that the continuity of the solution $u$ across $\Gamma$ is imposed by the Dirichlet boundary condition in the local problems and the fact that $\hat{u}$ is single valued on $\Gamma$.

The discretization of the local problems and the global equations leads to the HDG discrete problem: find $u_h \in \mathcal{V}^h$, $q_h \in [\mathcal{V}^h]^d$ and $\hat{u}_h \in \Lambda^h$ such that $\hat{u}_h = \mathbb{P}_2(u_D)$ on $\Gamma_D$ and

$$
\begin{align*}
\int_{K_i} v \nabla \cdot q_h \, dV + \int_{\partial K_i} \nu v (u_h - \hat{u}_h) \, dS &= \int_{K_i} v f \, dV \\
\int_{K_i} q_h \cdot w \, dV - \int_{\partial K_i} \nu u_h \nabla \cdot w \, dV + \int_{\partial K_i} \nu \hat{u}_h w \cdot n \, dS &= 0
\end{align*}
$$

for $i = 1, \ldots, n_{el}$, and

$$
\begin{align*}
\int_{\Gamma_f} \hat{v} \|q_h \cdot n\| \, dS + 2 \int_{\Gamma_f} \tau \hat{v} \left( \{\nu u_h\} - \{\nu\} \hat{u}_h \right) \, dS &= 0, \\
\int_{\Gamma_f} \hat{v} q_h \cdot n \, dS + \int_{\Gamma_f} \tau \nu \hat{v} (u_h - \hat{u}_h) &= \int_{\Gamma_f} \hat{v} g \, dS,
\end{align*}
$$

$\forall \, v \in \mathcal{V}^h$, $w \in [\mathcal{V}^h]^d$ and $\hat{v} \in \Lambda^h$ such that $\hat{v} = 0$ on $\Gamma_D$, where $\{\cdot\}$ is the mean operator on the interior faces, $\{\cdot\} = \frac{1}{2} (\circlearrowleft L(f) + \circlearrowleft R(f))$ on $\Gamma_f$.

The discrete spaces for elemental variables, $u$ and $q$, and for the trace variable, $\hat{u}$, are

$$
\begin{align*}
\mathcal{V}^h := \{ v \in L^2(\Omega) : v|_{K_i} \in \mathcal{P}_k(K_i) \text{ for } i = 1, \ldots, n_{el} \} \\
\Lambda^h := \{ \hat{v} \in L^2(\Gamma) : \hat{v}|_{\Gamma_f} \in \mathcal{P}_k(\Gamma_f) \text{ for } f = 1, \ldots, n_{fc} \}
\end{align*}
$$

where $\mathcal{P}_k$ denotes the space of polynomials of degree less or equal to $k$.

Remark 2.1. The parameter $\tau$ is a non-negative stabilisation parameter usually taken of order $O(1)$. For each element, it may be taken as a positive constant on all faces, or positive on one arbitrary face and zero at the rest (single face). Both options lead in practice to stable and optimally convergent solutions, with super-convergent post-processed solutions. See section 2.1 and, for instance, Giorgiani et al. (2013a), Cockburn et al. (2008) for details on the influence of this parameter on the solution behaviour.

Remark 2.2. Different degree of approximation can be considered in each element with a straight-forward implementation. Based on this advantage of HDG, and the superconvergence properties that will be commented later, Giorgiani et al. (2013a) proposed an automatic degree-adaptive procedure.
Equations (5) correspond to the discretization of the local problem in each element. The first equation can be derived from the first equation in (2) applying integration by parts, replacing the flux by the numerical flux

\[ \hat{q} := q + \tau \nu (u - \hat{u}) n, \]

and undoing the integration by parts. In fact, it can also be interpreted as the weighted residual of the PDE plus an stabilization term (that is zero for the analytical solution) with the parameter \( \tau \). The second equation in (5) is obtained from the weak form of the second equation in (2), applying integration by parts and replacing the boundary condition \( u = \hat{u} \) on the element boundary.

**Figure 2.** Representation of the connectivity matrix for faces, \( F \): the three faces for the \( i \)-th element, \( K_i \), correspond to faces \( F_{i1}, F_{i2} \) and \( F_{i3} \).

The discretization of the local problem (5), for each element, can also be written in matrix form as

\[
\begin{align*}
A_{u_{i1}}^{K_i} u^i + A_{u_{i2}}^{K_i} q^i + A_{u_{i3}}^{K_i} \Lambda^i &= f_{u}^{K_i} \\
A_{q_{i1}}^{K_i} u^i + A_{q_{i2}}^{K_i} q^i + A_{q_{i3}}^{K_i} \Lambda^i &= 0
\end{align*}
\]

(9)

where \( u^i \) and \( q^i \) are the vectors of nodal values of \( u \) and \( q \) in element \( K_i \), and \( \Lambda^i \) is the vector of nodal values of \( \hat{u} \) on the \( n \) faces of the element (\( n = 3 \) for triangles, \( n = 4 \) for tetrahedra or quads, etc). That is,

\[
\Lambda^i := \begin{bmatrix} \hat{u}_{F_{i1}} \\ \vdots \\ \hat{u}_{F_{in}} \end{bmatrix},
\]

(10)

where \( \hat{u}_{F_{ij}} \) denotes the nodal values of \( \hat{u} \) on face \( \Gamma_{ij} \), and \( F_{ij} \) is the number of the \( j \)-th face of element \( K_i \); see an example in Figure 2. Note also that the subindexes in the \( A \) matrices refer to the space for the weighting function and the test function.
System (9) can be solved for $u^i$ and $q^i$ in each element, obtaining the so-called local solver in the element $K_i$

$$u^i = U^{K_i}A^i + f_{U}^{K_i}, \quad q^i = Q^{K_i}A^i + f_{Q}^{K_i}, \quad (11)$$

with

$$\begin{bmatrix}
U^{K_i} \\
Q^{K_i}
\end{bmatrix} = -A^{-1} \begin{bmatrix}
A_{u\bar{u}}^{K_i} \\
A_{q\bar{u}}^{K_i}
\end{bmatrix}, \quad \begin{bmatrix}
f_{U}^{K_i} \\
f_{Q}^{K_i}
\end{bmatrix} = \bar{A}^{-1} \begin{bmatrix}
f_{u}^{K_i} \\
0
\end{bmatrix} \quad (12)$$

and

$$\bar{A} = \begin{bmatrix}
A_{u\bar{u}}^{K_i} & A_{u\bar{q}}^{K_i} \\
A_{q\bar{u}}^{K_i} & A_{q\bar{q}}^{K_i}
\end{bmatrix}$$

That is, for each element, the elemental values of the solution, $u^i$ and $q^i$, can be explicitly expressed in terms of the trace on its faces, $\Lambda^i$.

On other hand, the equations (6) correspond to the discretization of the global equations, (3), to determine the trace variable $\hat{u}$, imposing in weak form continuity of the normal flux and Neumann boundary conditions. They can also be written as

$$\sum_{i=1}^{n_e} \int_{\partial K_i} (\tilde{v} q_h \cdot n + \tau \tilde{v} (u_h - \tilde{u}_h)) \, dS = \int_{\Gamma_N} \tilde{v} g \, dS, \quad (13)$$

where $n$ denotes the normal vector exterior to $K_i$. For an interior face $\Gamma_f$, the equation can be written in matrix form as

$$A_{u\bar{u}}^{f,L} \hat{u}_L(f) + A_{u\bar{q}}^{f,L} q_L(f) + A_{q\bar{u}}^{f,R} u_R(f) + A_{q\bar{q}}^{f,R} q_R(f) + A_{u\bar{u}}^{f} \hat{u}_f = 0. \quad (14)$$

Then, replacing the local solver (11), for the elements $K_{L(f)}$ and $K_{R(f)}$, in (14) for every face $\Gamma_f$, leads to a system of equations involving only the trace variables $\{\tilde{u}_f\}_{f=1}^{n_f}$.

![Figure 3](image-url) Example of face $\Gamma_1$ shared by 2 triangular elements.
For instance, the block equation corresponding to the face $\Gamma_1$, shared by elements $K_L$ and $K_R$, as represented in Figure 3 would be

$$\begin{bmatrix}
A_{uu}^{1,L} U_{KL} + A_{uq}^{1,L} Q_{KL} \\
A_{uu}^{1,R} U_{KR} + A_{uq}^{1,R} Q_{KR}
\end{bmatrix}
\begin{bmatrix}
\hat{u}_1 \\
\hat{u}_2 \\
\hat{u}_3
\end{bmatrix}
+ \begin{bmatrix}
A_{uu}^{1,L} \hat{f}_{KL} \\
A_{uu}^{1,R} \hat{f}_{KR}
\end{bmatrix}
\begin{bmatrix}
\hat{u}_1 \\
\hat{u}_4 \\
\hat{u}_5
\end{bmatrix}
+ \hat{u}_1 = -A_{uu}^{1,L} f_{KL} U_{KL} - A_{uq}^{1,L} f_{KL} Q_{KL} - A_{uu}^{1,R} f_{KR} U_{KR} - A_{uq}^{1,R} f_{KR} Q_{KR}
$$

The only lost of generalization in the example is the numbering of the faces in the two triangles. The structure of the equations would be the same for any interior face, just involving the faces in the two elements sharing the face. Note that the contributions from each element to the block equation can be easily identified. That is, the face equations can be in fact computed as an assembly of the contributions from each element sharing the face.

Thus, similarly to the assembly in CFE, the computation of the HDG system is implemented with a loop over elements. For each element, the matrices and vectors for the local solver (11) are computed, and the contribution to the equation (14) is assembled for each one of the faces of the element. Once the system is assembled for all elements, and Dirichlet boundary conditions (4) are imposed, the system can be solved. Then, given the trace variables $\{\hat{u}_f\}_{f=1}^n$, the solution, $u^*$ and $q^*$, can be computed for each element using (11).

A Matlab code with the same notation used in this document can be found at https://www.lacan.upc.edu/user/sonia-fernandez/

2.1 Convergence and postprocess for superconvergent approximation $u^*$

The HDG approximation, with degree $k$, provides an approximation with convergence of order $k + 1$ for both $u$ and for the approximation of the derivative $q$. In addition, the mean of the solution in the elements is superconvergent with order $k + 2$. See Cockburn et al. (2009) and Cockburn et al. (2008) for mathematical proofs. Thanks to these expectional convergence properties, a cheap element-by-element postprocess can be computed to get a new approximation $u_{h}^*$ with superconvergence of order $k + 2$.

The problem to be solved in each element is:

$$\nabla \cdot \nabla u_{h}^* = \nabla \cdot q_h \quad \text{in } K_i$$

$$\nabla u_{h}^* \cdot n = q_h \cdot n \quad \text{on } \partial K_i$$

$$\int_{K_i} u_{h}^* \, dV = \int_{K_i} u_h \, dV$$

Superconvergence is ensured when the stabilization parameter $\tau$ is defined in each element as null in all faces except one, with an arbitrary positive constant value. This choice is refered to by some authors as single face. In any
In practice, the superconvergence postprocess always provides a better approximation, with a convergence rate at least $k + 1$ and in most cases close to $k + 2$. The reader is invited to do some tests with the Matlab code available at [https://www.lacan.upc.edu/user/sonia-fernandez/](https://www.lacan.upc.edu/user/sonia-fernandez/).

Figure 4 shows an example on a coarse finite element mesh, with degree $k = 2$. Discontinuities can be clearly seen in $u_h$, whereas $u^*_h$ shows much smaller discontinuities, which can be understood as an indicator of improved accuracy. More precisely, in this example the postprocess reduces the $L_2$ error from $0.9 \times 10^{-2}$ to $0.7 \times 10^{-3}$.

[Giorgiani et al. (2013a, 2014)](https://www.lacan.upc.edu/user/sonia-fernandez/) proposed an automatic adaptive algorithm based on the superconvergent solution $u^*$. The error is estimated in each element simply as the difference of the two available approximations, $u$ and $u^*$, and the degree is each element is consequently adapted.

### 2.2 Sparsity pattern of the HDG matrix and computational efficiency

Compared to other DG methods (such as Interior Penalty Method (IPM), Compact Discontinuous Galerkin (CDG), among others), the HDG global system has much less degrees of freedom, mainly thanks to the static condensation of elemental variables in terms of trace variables. In fact, the final number of degrees of freedom is close to CFE for high degree, as can be observed in the example in Figure 5 and in the work by [Huerta et al. (2013)](https://www.lacan.upc.edu/user/sonia-fernandez/). Moreover, HDG matrices have a special block structure, because every interior face is connected to the same number of faces. For instance, in a mesh of triangles, every face is connected to itself and four more faces, corresponding to the faces in the two elements sharing...
Figures from Giorgiani et al. (2013b). The rows in red and blue correspond to the equations of the nodes marked with a red star (vertex node) and a blue star (side node), respectively. The stencil of these nodes is shown as black nodes on the mesh.

Thus, every block of rows, has five non-null blocks.

This special structure seems to be advantageous for linear solvers. A detailed comparison of HDG and CG in 2D and 3D can be found in the work by Kirby et al. (2011); Yakovlev et al. (2015), and also in Giorgiani et al. (2013b). Numerical experiments show that the computational time for assembly of HDG is greater than for CFE, but it may be compensated by a smaller CPU time in the linear solver, and by the higher accuracy (in part thanks to superconvergence) of HDG. Thus,
HDG exhibits computational efficiency similar to CFE (in terms of CPU time for a given level of accuracy) but with the classical advantages of DG methods, such as easy adaptivity, suitability for parallel computing, stability through numerical fluxes and conservativity.

3 Incompressible flow

The following Stokes problem is considered,

\[ -\nabla \cdot (\nu \nabla \mathbf{u}) + \nabla p = f \quad \text{in } \Omega \]
\[ \nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega \]
\[ (\nu \nabla \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n} = g \quad \text{on } \Gamma_N \]
\[ \mathbf{u} = \mathbf{u}_D \quad \text{on } \Gamma_D \]

(15)

where \( \mathbf{u} \) is the velocity, \( p \) is the pressure, \( \nu \) is the material viscosity, \( \mathbf{I} \) is the second-order identity tensor, \( f \) is an external force, \( \mathbf{u}_D \) are prescribed values on the Dirichlet boundaries \( \Gamma_D \), and \( g \) is a prescribed traction on the Neumann boundary \( \Gamma_N \).

The domain \( \Omega \) is again assumed to be covered by a finite element mesh, with the same notation as in the previous section. And, following the HDG rationale, problem (15) is split in a set of local problems, for each one of the elements, and some global equations defined on element faces. The local problems are pure Dirichlet problems. That is, for each element

\[ L - \nabla \mathbf{u} = 0, \quad \nabla \cdot (-\nu L + p \mathbf{I}) = f \quad \text{and} \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } K_i \]
\[ \mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \partial K_i \]
\[ \int_{\partial K_i} p \, dS = \rho_i. \]

(16a)
(16b)
(16c)

The variable \( L \) is the gradient of \( \mathbf{u} \), allowing the splitting of the PDE in two first order PDEs, \( \hat{\mathbf{u}} \) is the trace of \( \mathbf{u} \) at the mesh faces \( \Gamma \), and \( \rho_i \) is the mean of the pressure at the boundary of the element, which is only a scalar for each element. The new variables, \( \hat{\mathbf{u}} \) and \( \{\rho_i\}_{i=1}^{n_{el}} \), are assumed to be data in the local problems. Note that the Stokes problem with only Dirichlet data does not have a unique solution, and (16c) closes the problem, setting the mean of the pressure on the element boundary to \( \rho_i \).

The local problems (16) can be solved element-by-element to determine \( \mathbf{u}, L \) and \( p \), given \( \hat{\mathbf{u}} \) and \( \rho_i \). Thus, now the problem reduces to find \( \hat{\mathbf{u}} \) and \( \{\rho_i\}_{i=1}^{n_{el}} \) with the global equations

\[ (\nu L + p \mathbf{I}) \cdot \mathbf{n} = 0 \quad \text{on } \Gamma \setminus \partial \Omega, \quad (\nu L + p \mathbf{I}) \cdot \mathbf{n} = g \quad \text{on } \Gamma_N, \]
\[ \int_{\partial K_i} \hat{\mathbf{u}} \cdot \mathbf{n} \, dS = 0 \quad \text{for } i = 1, \ldots, n_{el}, \]

(17a)
(17b)

(10)
\[ \hat{u} = P_2(u_D) \quad \text{on } \Gamma_D. \quad (17c) \]

Equation (17a) is the conservativity condition, imposing equilibrium of the traction on element faces and on the Neumann boundary. Equation (17b) imposes the incompressibility condition on the boundary of the elements, ensuring well-posedness of the Dirichlet local problems (16).

If the problem is a pure Dirichlet problem, i.e. \( \Gamma_N = \emptyset \), the solution of (15) is determined up to a constant for the pressure. In this case, an additional constraint for the pressure should be imposed at the global level, for instance, setting the mean of the pressure on the boundary of the first element to 0, that is \( \rho_1 = 0 \), or to a given constant.

The discretization spaces in (7) are now considered for elemental variables \( u \), \( L \) and \( p \), and for the trace variable \( \hat{u} \), respectively. The discretization of the local problems (16) and the global equations (17) leads to the complete HDG formulation detailed next.

The HDG local problem for each element \( K_i \) is: given \( \hat{u}_h \in [\Lambda^h]^d \) and \( \rho_h \in \mathbb{R} \), find \( u_h \in [P_k(K_i)]^d \), \( L_h \in [P_k(K_i)]^{d \times d} \) and \( p_h \in P_k(K_i) \) such that

\[
\int_{K_i} (-\nabla \cdot (\nu L_h) + \nabla p_h) \cdot v \, dV + \int_{\partial K_i} \tau \nu (u_h - \hat{u}_h) \cdot v \, dS = \int_{K_i} f \cdot v \, dV
\]

\[
\int_{K_i} L_h : Q \, dV + \int_{K_i} (\nabla \cdot Q) \cdot u_h \, dV - \int_{\partial K_i} (Q \cdot n) \cdot \hat{u}_h \, dS = 0
\]

\[
\int_{K_i} u_h \cdot \nabla q \, dV - \int_{\partial K_i} (\hat{u}_h \cdot n) q \, dS = 0
\]

\[
\frac{1}{|\partial K_i|} \int_{\partial K_i} p_h \, dS = \rho_i,
\]

for all \( v \in [P_k(K_i)]^d \), \( Q \in [P_k(K_i)]^{d \times d} \) and \( q \in P_k(K_i) \). The first equation in (18) can be derived from the first equation in (16a) by applying integration by parts, replacing the velocity gradient by the numerical velocity gradient,

\[ \hat{L} := L + \tau (\hat{u} - u) \otimes n, \]

and undoing the integration by parts. The stabilization parameter \( \tau \) can be taken as \( \tau = 1 \), see Nguyen et al. (2009); Giorgiani et al. (2014) for details. The second and third equations are obtained from the weak form of the second and third equations in (16a) by simply applying integration by parts and replacing the boundary condition (16c) on the element boundary.

The discretization of the local problem leads now to a system of equations of
the form

\[
\begin{bmatrix}
A_{uu}^i & A_{uL}^i & A_{up}^i & 0 \\
A_{Lu}^i & A_{LL}^i & 0 & 0 \\
A_{pu}^i & 0 & 0 & A_{pp}^i \\
0 & 0 & A_{pp}^i & 0 \\
\end{bmatrix}
\begin{bmatrix}
u^i \\
L^i \\
p^i \\
\lambda \\
\end{bmatrix}
= \begin{bmatrix}
\nu^i \\
L^i \\
p^i \\
\lambda \\
\end{bmatrix} + \begin{bmatrix}
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
+ \begin{bmatrix}
\rho_i \\
\rho_i \\
\rho_i \\
\rho_i \\
\end{bmatrix}
\begin{bmatrix}
A_{uu}^i \\
A_{uL}^i \\
A_{up}^i \\
A_{pp}^i \\
\end{bmatrix}
\begin{bmatrix}
\nu^f \\
L^f \\
p^f \\
\lambda \\
\end{bmatrix}
\]

(20)

where the vectors \(\nu^i\), \(L^i\) and \(p^i\) are elemental vectors of nodal values, and \(\Lambda^i\) is a vector with the nodal values of the trace of the velocity \(\nu^f\) for the faces of the element. Note that the constraint for the mean of the pressure, \(\frac{1}{|\partial K_i|} \int_{\partial K_i} p \, dS = \rho_i\), is imposed with a Lagrange multiplier \(\lambda\).

This system can be solved for each element, leading to the so-called local solver in the element, that is, an explicit expression of the elemental variables \(\nu^i\), \(L^i\) and \(p^i\), in terms of the trace of the velocity at the faces \(\Lambda^f\) and the mean of the pressure \(\rho_i\).

\[
\begin{align*}
\nu^i &= S_{\nu}^i \Lambda^f + r_{\nu}^i \rho_i + f_{\nu}^i, \\
L^i &= S_{L}^i \Lambda^f + r_{L}^i \rho_i + f_{L}^i, \\
p^i &= S_{p}^i \Lambda^f + r_{p}^i \rho_i + f_{p}^i,
\end{align*}
\]

(21)

where the matrices \(S_{\nu}^i\) and the vectors \(r_{\nu}^i\), \(f_{\nu}^i\) depend on the matrices and vectors in (20).

The HDG global problem corresponding to the discretization of (17) with the numerical velocity gradient (19) is: find \(\hat{u}_h \in [\Lambda^h]^d\) and \(\rho_h \in \mathbb{R}\) for \(i = 1, \ldots, n_{el}\) satisfying

\[
\sum_{i=1}^{n_{el}} \int_{\partial K_i} \hat{v} \cdot ((-\nu L_h + p_h I) \cdot n + \tau \nu \hat{\nu} \cdot (u_h - \hat{u}_h)) \, dS = \int_{\Gamma_N} \hat{v} \cdot g \, dS,
\]

(22)

for all \(\hat{v} \in [\Lambda^h]^d\) such that \(\hat{v} = 0\) on \(\Gamma_D\).

\[
\int_{\partial K_i} \hat{u}_h \cdot n \, dS = 0 \quad \text{for } i = 1, \ldots, n_{el}
\]

(23)

and

\[
\hat{u}_h = \mathbb{P}_2(u_D) \quad \text{on } \Gamma_D,
\]

(24)

where \(u_h\), \(L_h\) and \(p_h\) are solutions to the local problem for each element, i.e., the solution of (18). Equations (22) and (23) can be written in matrix form as

\[
\begin{align*}
A_{uu}^{f,L} u_L(f) + A_{uL}^{f,L} L_L(f) + A_{up}^{f,L} p_L(f) \\
+ A_{uu}^{f,R} u_R(f) + A_{uL}^{f,R} L_R(f) + A_{up}^{f,R} p_R(f) \quad + A_{\hat{u}\hat{u}}^{f} \hat{u}^f = 0
\end{align*}
\]

(25)
for $f = 1, \ldots, n_{fc}$, and

$$A_{\mu^i}^i \mathbf{A}^i = 0$$ \hspace{1cm} (26)

for $i = 1, \ldots, n_{el}$. Replacing the local solver – that is, equation (21) for the elements $K_L(f)$ and $K_R(f)$ – in (25) and (26), and applying the Dirichlet boundary condition (24), the global problem leads to a system of equations involving only the trace variable $\{\hat{u}^f\}^i_{f=1}$ and the mean of the pressure on the boundary of the elements $\{\rho_i\}^i_{i=1}$. After solving this global system, elemental variables can be obtained by simply plugging in the solution, $\{\hat{u}^f\}^i_{f=1}$ and $\{\rho_i\}^i_{i=1}$, in the local solver for each element.

Analogously to Laplace, an element-by-element postprocessing provides a superconvergent solution, $\mathbf{u}_*^h$, with order $k + 2$ in $L_2$ norm: given $\mathbf{u}_h \in [P_k(K_i)]^d$ and $L_h \in [P_{k+1}(K_i)]^d$, find $\mathbf{u}_*^h \in [P_{k+1}(K_i)]^d$ such that

$$\int_{K_i} \nabla \mathbf{u}_*^h : \nabla \mathbf{v} \, dV = \int_{K_i} L_h : \nabla \mathbf{v} \, dV \quad \forall \mathbf{v} \in [P_{k+1}(K_i)]^d,$$

$$\int_{K_i} \mathbf{u}_*^h \, dV = \int_{K_i} \mathbf{u}_h \, dV.$$ \hspace{1cm} (27)

A Matlab code implementing HDG for Stokes can be found at

https://www.lacan.upc.edu/user/sonia-fernandez/

### 3.1 Matrix structure and computational efficiency

The static condensation of elemental variables is even more advantageous in incompressible flow problems, since in this case the degrees of freedom corresponding to the pressure are reduced to just one scalar per element, $\rho_i$, regardless the degree of approximation $k$. Figure 6 shows an example of the sparsity pattern of the global matrix for CFE, with a Taylor-Hook approximation (with degree $k-1$ for pressure and $k$ for velocity) and static condensation, and for HDG with degree $k = 5$.

Looking to the blue block, corresponding to velocity unknowns, we can again observe that for the same mesh and same degree, HDG has more degrees of freedom, but it also exhibits the nice block structure that is advantageous for linear solvers. Now, in addition, the number of degrees of freedom for pressure is drastically reduced in HDG, with just one unknown $\rho_i$ per element, for any degree of approximation $k$. Moreover, with these approximations the HDG solution is expected to be more accurate than the CFE one, since the velocity $\mathbf{u}_*^h$ converges with order $k + 2$, in front of the order $k$ in CFE, and the pressure $p_\mathbf{u}$ converges with order $k + 1$, instead of $k$ for CFE. A critical comparison, in terms of CPU time for similar accuracy, for incompressible flow can be found in [Paipuri et al. (2018)].
3.2 Some comments on Navier-Stokes

Similarly to Stokes, the non-linear Navier-Stokes equations can be solved in an efficient way, just applying the same static condensation tricks for the linearized system in the non-linear solver iterations, see for instance Paipuri et al. (2018) for details.

The stabilization parameter $\tau$ can play an important role in the presence of sharp fronts for high Reynolds number, see for instance Paipuri et al. (2018); Gia- comini et al. (2020).

Bibliography


