Sonia Fernández-Méndez

Universitat Politècnica de Catalunya (UPC-BarcelonaTech), Laboratori de Càlcul Numèric (LaCàN), <https://www.lacan.upc.edu/user/sonia-fernandez/>

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, specially 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 $\mathcal{L}_2$ 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 $\mathcal{L}_2$ 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,

$$\begin{aligned} -\nabla \cdot (\nu \nabla u) &= f && \text{in } \Omega \\ -\nu \nabla u \cdot \mathbf{n} &= g && \text{on } \Gamma_N \\ u &= u_D && \text{on } \Gamma_D \end{aligned} \quad (1)$$

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

The domain Ω is now assumed to be split by a finite element mesh with n_{e1} disjoint elements K_i , such that

$$\overline{\Omega} \subset \bigcup_{i=1}^{n_{e1}} \overline{K_i}, \quad K_i \cap K_j = \emptyset \text{ for } i \neq j.$$

The union of all n_{fc} faces Γ_i (sides for 2D) is denoted as

$$\Gamma := \bigcup_{i=1}^{n_{e1}} \partial K_i = \bigcup_{f=1}^{n_{fc}} \Gamma_f.$$

The key idea of the HDG method is introducing a new unknown, $\hat{u}$, corresponding to the solution u on the mesh skeleton Γ , 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 Γ 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*,

$$\left. \begin{aligned} \nabla \cdot \mathbf{q} &= f && \text{in } K_i \\ \mathbf{q} + \nu \nabla u &= 0 && \text{in } K_i \\ u &= \hat{u} && \text{on } \partial K_i \end{aligned} \right\} \text{ for } i = 1, \dots, n_{\text{el}}. \quad (2)$$

The solution of the local problem, in each element, allows expressing u , and also the flux $\mathbf{q}$, 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*,

$$[[\mathbf{q} \cdot \mathbf{n}]] = 0 \quad \text{on } \Gamma \setminus \partial\Omega, \quad \mathbf{q} \cdot \mathbf{n} = g \quad \text{on } \Gamma_N, \quad (3)$$

and the Dirichlet boundary conditions

$$\hat{u} = \mathbb{P}_2(u_D) \quad \text{on } \Gamma_D, \quad (4)$$

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

$$[[\odot]] = \odot_{L(f)} + \odot_{R(f)} \quad \text{on } \Gamma_f,$$

where $R(f)$ and $L(f)$ are the numbers of the left and right elements sharing the face, that is, $\Gamma_f = \overline{K_{L(f)}} \cap \overline{K_{R(f)}}$, and the subindex $\odot_i$ denotes the value of function $\odot$ from element K_i . In particular, $[[\mathbf{q} \cdot \mathbf{n}]] = \mathbf{q}_{L(f)} \cdot \mathbf{n}_{L(f)} + \mathbf{q}_{R(f)} \cdot \mathbf{n}_{R(f)} = (\mathbf{q}_{L(f)} - \mathbf{q}_{R(f)}) \cdot \mathbf{n}_{L(f)}$.

It is important noting that the continuity of the solution u across Γ is imposed by the Dirichlet boundary condition in the local problems (2) and the fact that $\hat{u}$ is single valued on Γ .

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

$$\begin{aligned} \int_{K_i} v \nabla \cdot \mathbf{q}_h dV + \int_{\partial K_i} \tau \nu v (u_h - \hat{u}_h) dS &= \int_{K_i} v f dV \\ \int_{K_i} \mathbf{q}_h \cdot \mathbf{w} dV - \int_{K_i} \nu u_h \nabla \cdot \mathbf{w} dV + \int_{\partial K_i} \nu \hat{u}_h \mathbf{w} \cdot \mathbf{n} dS &= 0 \end{aligned} \quad (5)$$

for $i = 1, \dots, n_{e1}$, and

$$\begin{aligned} \int_{\Gamma} \hat{v} [\![\mathbf{q}_h \cdot \mathbf{n}]\!] dS + 2 \int_{\Gamma} \tau \hat{v} (\{\nu u_h\} - \{\nu\} \hat{u}_h) dS &= 0, \\ \int_{\Gamma_N} \hat{v} \mathbf{q}_h \cdot \mathbf{n} dS + \int_{\Gamma_N} \tau \nu \hat{v} (u_h - \hat{u}_h) &= \int_{\Gamma_N} \hat{v} g dS, \end{aligned} \quad (6)$$

$\forall v \in \mathcal{V}^h$, $\mathbf{w} \in [\mathcal{V}^h]^d$ and $\hat{v} \in \Lambda^h$ such that $\hat{v} = 0$ on Γ_D , where $\{\cdot\}$ is the mean operator on the interior faces,

$$\{\odot\} = \frac{1}{2} (\odot_{L(f)} + \odot_{R(f)}) \quad \text{on } \Gamma_f.$$

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

$$\begin{aligned} \mathcal{V}^h &:= \{v \in \mathcal{L}^2(\Omega) : v|_{K_i} \in \mathcal{P}_k(K_i) \text{ for } i = 1, \dots, n_{e1}\} \\ \Lambda^h &:= \{\hat{v} \in \mathcal{L}^2(\Gamma) : \hat{v}|_{\Gamma_f} \in \mathcal{P}_k(\Gamma_f) \text{ for } f = 1, \dots, n_{fc}\}, \end{aligned} \quad (7)$$

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

Remark 2.1. The parameter τ is a non-negative stabilisation parameter usually taken of order $\mathcal{O}(1)$. 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 superconvergent 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

$$\hat{\mathbf{q}} := \mathbf{q} + \tau \nu (u - \hat{u}) \mathbf{n}, \quad (8)$$

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 τ . 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, $\mathbf{F}$: 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

$$\begin{cases} \mathbf{A}_{uu}^{K_i} \mathbf{u}^i + \mathbf{A}_{uq}^{K_i} \mathbf{q}^i + \mathbf{A}_{u\hat{u}}^{K_i} \Lambda^i = \mathbf{f}_u^{K_i} \\ \mathbf{A}_{qu}^{K_i} \mathbf{u}^i + \mathbf{A}_{qq}^{K_i} \mathbf{q}^i + \mathbf{A}_{q\hat{u}}^{K_i} \Lambda^i = \mathbf{0} \end{cases} \quad (9)$$

where $\mathbf{u}^i$ and $\mathbf{q}^i$ are the vectors of nodal values of u and $\mathbf{q}$ in element K_i , and Λ^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,

$$\Lambda^i := \begin{bmatrix} \hat{\mathbf{u}}^{\mathbf{F}_{i1}} \\ \vdots \\ \hat{\mathbf{u}}^{\mathbf{F}_{in}} \end{bmatrix}, \quad (10)$$

where $\hat{\mathbf{u}}^f$ denotes the nodal values of $\hat{u}$ on face Γ_f , and $\mathbf{F}_{ij}$ 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 $\mathbf{A}$ matrices refer to the space for the weighting function and the test function.

System (9) can be solved for $\mathbf{u}^i$ and $\mathbf{q}^i$ in each element, obtaining the so-called *local solver* in the element K_i

$$\mathbf{u}^i = \mathbf{U}^{K_i} \boldsymbol{\Lambda}^i + \mathbf{f}_U^{K_i}, \quad \mathbf{q}^i = \mathbf{Q}^{K_i} \boldsymbol{\Lambda}^i + \mathbf{f}_Q^{K_i}, \quad (11)$$

with

$$\begin{bmatrix} \mathbf{U}^{K_i} \\ \mathbf{Q}^{K_i} \end{bmatrix} = -\mathbb{A}^{-1} \begin{bmatrix} \mathbf{A}_{u\hat{u}}^{K_i} \\ \mathbf{A}_{q\hat{u}}^{K_i} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{f}_U^{K_i} \\ \mathbf{f}_Q^{K_i} \end{bmatrix} = \mathbb{A}^{-1} \begin{bmatrix} \mathbf{f}_u^{K_i} \\ \mathbf{0} \end{bmatrix} \quad (12)$$

and

$$\mathbb{A} = \begin{bmatrix} \mathbf{A}_{uu}^{K_i} & \mathbf{A}_{uq}^{K_i} \\ \mathbf{A}_{qu}^{K_i} & \mathbf{A}_{qq}^{K_i} \end{bmatrix}$$

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

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

$$\sum_{i=1}^{n_{el}} \int_{\partial K_i} (\hat{v} \mathbf{q}_h \cdot \mathbf{n} + \tau \hat{v} (u_h - \hat{u}_h)) dS = \int_{\Gamma_N} \hat{v} g dS, \quad (13)$$

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

$$\mathbf{A}_{\hat{u}u}^{f,L} \mathbf{u}^{L(f)} + \mathbf{A}_{\hat{u}q}^{f,L} \mathbf{q}^{L(f)} + \mathbf{A}_{\hat{u}u}^{f,R} \mathbf{u}^{R(f)} + \mathbf{A}_{\hat{u}q}^{f,R} \mathbf{q}^{R(f)} + \mathbf{A}_{\hat{u}\hat{u}}^f \hat{\mathbf{u}}^f = 0. \quad (14)$$

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

Figure 3. Example of face Γ_1 shared by 2 triangular elements.

For instance, the block equation corresponding to the face Γ_1 , shared by elements K_L and K_R , as represented in Figure 3, would be

$$\begin{aligned} & \left[\mathbf{A}_{\hat{u}u}^{1,L} \mathbf{U}^{K_L} + \mathbf{A}_{\hat{u}q}^{1,L} \mathbf{Q}^{K_L} \right] \begin{bmatrix} \hat{\mathbf{u}}^1 \\ \hat{\mathbf{u}}^2 \\ \hat{\mathbf{u}}^3 \end{bmatrix} + \left[\mathbf{A}_{\hat{u}u}^{1,R} \mathbf{U}^{K_R} + \mathbf{A}_{\hat{u}q}^{1,R} \mathbf{Q}^{K_R} \right] \begin{bmatrix} \hat{\mathbf{u}}^1 \\ \hat{\mathbf{u}}^4 \\ \hat{\mathbf{u}}^5 \end{bmatrix} \\ & + \mathbf{A}_{\hat{u}\hat{u}}^1 \hat{\mathbf{u}}^1 = -\mathbf{A}_{\hat{u}u}^{1,L} \mathbf{f}_U^{K_L} - \mathbf{A}_{\hat{u}q}^{1,L} \mathbf{f}_Q^{K_L} - \mathbf{A}_{\hat{u}u}^{1,R} \mathbf{f}_U^{K_R} - \mathbf{A}_{\hat{u}q}^{1,R} \mathbf{f}_Q^{K_R} \end{aligned}$$

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 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 $\{\hat{\mathbf{u}}^f\}_{f=1}^{n_{fc}}$, the solution, $\mathbf{u}^i$ and $\mathbf{q}^i$, 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 $\mathbf{q}$. 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 exceptional 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:

$$\begin{aligned} \nabla \cdot \nabla u_h^* &= \nabla \cdot \mathbf{q}_h && \text{in } K_i \\ \nabla u_h^* \cdot \mathbf{n} &= \mathbf{q}_h \cdot \mathbf{n} && \text{on } \partial K_i \\ \int_{K_i} u_h^* dV &= \int_{K_i} u_h dV \end{aligned}$$

Superconvergence is ensured when the stabilization parameter τ is defined in each element as null in all faces except one, with an arbitrary positive constant value. This choice is referred to by some authors as *single face*. In any

Figure 4. Example of an HDG solution u_h (left) and u_h^* (right) on a coarse mesh of triangles with degree 2. The improvement in the postprocessed superconvergent solution can be clearly seen.

case, 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/>

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 $\mathcal{L}_2$ error from $0.9 \cdot 10^{-2}$ to $0.7 \cdot 10^{-3}$.

Giorgiani et al. (2013a, 2014) 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 of 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). 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

Figure 5. 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.

the face, as can be seen in figure 3. 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,

$$\begin{aligned} -\nabla \cdot (\nu \nabla \mathbf{u}) + \nabla p &= \mathbf{f} & \text{in } \Omega \\ \nabla \cdot \mathbf{u} &= 0 & \text{in } \Omega \\ (-\nu \nabla \mathbf{u} + p \mathbf{I}) \cdot \mathbf{n} &= \mathbf{g} & \text{on } \Gamma_N \\ \mathbf{u} &= \mathbf{u}_D & \text{on } \Gamma_D \end{aligned} \quad (15)$$

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

The domain Ω 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

$$\mathbf{L} - \nabla \mathbf{u} = 0, \quad \nabla \cdot (-\nu \mathbf{L} + p \mathbf{I}) = \mathbf{f} \quad \text{and} \quad \nabla \cdot \mathbf{u} = 0 \quad \text{in } K_i \quad (16a)$$

$$\mathbf{u} = \hat{\mathbf{u}} \quad \text{on } \partial K_i \quad (16b)$$

$$\frac{1}{|\partial K_i|} \int_{\partial K_i} p \, dS = \rho_i. \quad (16c)$$

The variable $\mathbf{L}$ 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 Γ , and ρ_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 $\{\rho_i\}_{i=1}^{n_{e1}}$, 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 ρ_i .

The local problems (16) can be solved element-by-element to determine $\mathbf{u}$, $\mathbf{L}$ and p , given $\hat{\mathbf{u}}$ and ρ_i . Thus, now the problem reduces to find $\hat{\mathbf{u}}$ and $\{\rho_i\}_{i=1}^{n_{e1}}$ with the global equations

$$\llbracket (-\nu \mathbf{L} + p \mathbf{I}) \cdot \mathbf{n} \rrbracket = 0 \quad \text{on } \Gamma \setminus \partial\Omega, \quad (-\nu \mathbf{L} + p \mathbf{I}) \cdot \mathbf{n} = \mathbf{g} \quad \text{on } \Gamma_N, \quad (17a)$$

$$\int_{\partial K_i} \hat{\mathbf{u}} \cdot \mathbf{n} \, dS = 0 \quad \text{for } i = 1, \dots, n_{e1}, \quad (17b)$$

$$\hat{\mathbf{u}} = \mathbb{P}_2(\mathbf{u}_D) \quad \text{on } \Gamma_D. \quad (17c)$$

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 $\mathbf{u}$, $\mathbf{L}$ and p , and for the trace variable $\hat{\mathbf{u}}$, 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 $\hat{\mathbf{u}}_h \in [\Lambda^h]^d$ and $\rho_i \in \mathbb{R}$, find $\mathbf{u}_h \in [\mathcal{P}_k(K_i)]^d$, $\mathbf{L}_h \in [\mathcal{P}_k(K_i)]^{d \times d}$ and $p_h \in \mathcal{P}_k(K_i)$ such that

$$\begin{aligned} \int_{K_i} (-\nabla \cdot (\nu \mathbf{L}_h) + \nabla p_h) \cdot \mathbf{v} \, dV + \int_{\partial K_i} \tau \nu (\mathbf{u}_h - \hat{\mathbf{u}}_h) \cdot \mathbf{v} \, dS &= \int_{K_i} \mathbf{f} \cdot \mathbf{v} \, dV \\ \int_{K_i} \mathbf{L}_h : \mathbf{Q} \, dV + \int_{K_i} (\nabla \cdot \mathbf{Q}) \cdot \mathbf{u}_h \, dV - \int_{\partial K_i} (\mathbf{Q} \cdot \mathbf{n}) \cdot \hat{\mathbf{u}}_h \, dS &= 0 \\ \int_{K_i} \mathbf{u}_h \cdot \nabla q \, dV - \int_{\partial K_i} (\hat{\mathbf{u}}_h \cdot \mathbf{n}) q \, dS &= 0 \\ \frac{1}{|\partial K_i|} \int_{\partial K_i} p_h \, dS &= \rho_i, \end{aligned} \quad (18)$$

for all $\mathbf{v} \in [\mathcal{P}_k(K_i)]^d$, $\mathbf{Q} \in [\mathcal{P}_k(K_i)]^{d \times d}$ and $q \in \mathcal{P}_k(K_i)$. 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,

$$\hat{\mathbf{L}} := \mathbf{L} + \tau(\hat{\mathbf{u}} - \mathbf{u}) \otimes \mathbf{n}, \quad (19)$$

and undoing the integration by parts. The stabilization parameter τ 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

$$\begin{bmatrix} \mathbf{A}_{uu}^i & \mathbf{A}_{uL}^i & \mathbf{A}_{up}^i & \mathbf{0} \\ \mathbf{A}_{Lu}^i & \mathbf{A}_{LL}^i & \mathbf{0} & \mathbf{0} \\ \mathbf{A}_{pu}^i & \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^i{}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{\rho p}^i & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^i \\ \mathbf{L}^i \\ \mathbf{p}^i \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{f}_u^i \\ \mathbf{f}_L^i \\ \mathbf{f}_p^i \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{1} \end{bmatrix} \rho_i - \begin{bmatrix} \mathbf{A}_{u\hat{u}}^i \\ \mathbf{A}_{L\hat{u}}^i \\ \mathbf{A}_{p\hat{u}}^i \\ \mathbf{0} \end{bmatrix} \mathbf{\Lambda}^i \quad (20)$$

where the vectors $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$ are elemental vectors of nodal values, and $\mathbf{\Lambda}^i$ is a vector with the nodal values of the trace of the velocity $\hat{\mathbf{u}}^f$ 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 λ .

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 $\mathbf{u}^i$, $\mathbf{L}^i$ and $\mathbf{p}^i$, in terms of the trace of the velocity at the faces $\mathbf{\Lambda}^i$ and the mean of the pressure ρ_i .

$$\begin{aligned} \mathbf{u}^i &= \mathbf{S}_u^i \mathbf{\Lambda}^i + \mathbf{r}_u^i \rho_i + \mathbf{f}_U^i, \\ \mathbf{L}^i &= \mathbf{S}_L^i \mathbf{\Lambda}^i + \mathbf{r}_L^i \rho_i + \mathbf{f}_L^i, \\ \mathbf{p}^i &= \mathbf{S}_p^i \mathbf{\Lambda}^i + \mathbf{r}_p^i \rho_i + \mathbf{f}_p^i, \end{aligned} \quad (21)$$

where the matrices $\mathbf{S}_*^i$ and the vectors $\mathbf{r}_*^i$, $\mathbf{f}_*^i$ 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 $\hat{\mathbf{u}}_h \in [\Lambda^h]^d$ and $\rho^i \in \mathbb{R}$ for $i = 1, \dots, n_{e1}$ satisfying

$$\sum_{i=1}^{n_{e1}} \int_{\partial K_i} \hat{\mathbf{v}} \cdot ((-\nu \mathbf{L}_h + p_h \mathbf{I}) \cdot \mathbf{n} + \tau \nu \hat{\mathbf{v}} \cdot (\mathbf{u}_h - \hat{\mathbf{u}}_h)) \, dS = \int_{\Gamma_N} \hat{\mathbf{v}} \cdot \mathbf{g} \, dS, \quad (22)$$

for all $\hat{\mathbf{v}} \in [\Lambda^h]^d$ such that $\hat{\mathbf{v}} = 0$ on Γ_D ,

$$\int_{\partial K_i} \hat{\mathbf{u}}_h \cdot \mathbf{n} \, dS = 0 \quad \text{for } i = 1, \dots, n_{e1} \quad (23)$$

and

$$\hat{\mathbf{u}}_h = \mathbb{P}_2(\mathbf{u}_D) \quad \text{on } \Gamma_D, \quad (24)$$

where $\mathbf{u}_h$, $\mathbf{L}_h$ 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

$$\begin{aligned} & \mathbf{A}_{\hat{u}u}^{f,L} \mathbf{u}^{L(f)} + \mathbf{A}_{\hat{u}L}^{f,L} \mathbf{L}^{L(f)} + \mathbf{A}_{\hat{u}p}^{f,L} \mathbf{p}^{L(f)} \\ & + \mathbf{A}_{\hat{u}u}^{f,R} \mathbf{u}^{R(f)} + \mathbf{A}_{\hat{u}L}^{f,R} \mathbf{L}^{R(f)} + \mathbf{A}_{\hat{u}p}^{f,R} \mathbf{p}^{R(f)} + \mathbf{A}_{\hat{u}\hat{u}}^f \hat{\mathbf{u}}^f = 0 \end{aligned} \quad (25)$$

for $f = 1, \dots, n_{fc}$, and

$$\mathbf{A}_{\rho\hat{\mathbf{u}}}^i \mathbf{\Lambda}^i = \mathbf{0} \quad (26)$$

for $i = 1, \dots, n_{e1}$. 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 $\{\hat{\mathbf{u}}^f\}_{f=1}^{n_{fc}}$ and the mean of the pressure on the boundary of the elements $\{\rho_i\}_{i=1}^{n_{e1}}$. After solving this global system, elemental variables can be obtained by simply plugging in the solution, $\{\hat{\mathbf{u}}^f\}_{f=1}^{n_{fc}}$ and $\{\rho_i\}_{i=1}^{n_{e1}}$, 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 $\mathcal{L}_2$ norm: given $\mathbf{u}_h \in [\mathcal{P}_k(K_i)]^d$ and $\mathbf{L}_h \in [\mathcal{P}_k(K_i)]^{d \times d}$, find $\mathbf{u}_h^* \in [\mathcal{P}_{k+1}(K_i)]^d$ such that

$$\begin{aligned} \int_{K_i} \nabla \mathbf{u}_h^* : \nabla \mathbf{v} \, dV &= \int_{K_i} \mathbf{L}_h : \nabla \mathbf{v} \, dV \quad \forall \mathbf{v} \in [\mathcal{P}_{k+1}(K_i)]^d, \\ \int_{K_i} \mathbf{u}_h^* \, dV &= \int_{K_i} \mathbf{u}_h \, dV. \end{aligned} \quad (27)$$

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, ρ_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 ρ_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 u_h^* converges with order $k + 2$, in front of the order k in CFE, and the pressure p_h 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).

Figure 6. Figures from Paipuri et al. (2018). Example of sparsity pattern for HDG and CFE (CG).

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 linealized system in the non-linear solver iterations, see for instance Paipuri et al. (2018) for details.

The stabilization parameter τ can play an important role in the presence of sharp fronts for high Reynolds number, see for instance Paipuri et al. (2018); Giacomini et al. (2020).

Bibliography

- B. Cockburn. Discontinuous Galerkin methods for computational fluid dynamics. In *Encyclopedia of Computational Mechanics*, volume 3 (Fluids), chapter 4. Wiley, New York, 2004.
- B. Cockburn, B. Dong, and J. Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Math. Comp.*, 77(264): 1887–1916, 2008. ISSN 0025-5718.
- 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(2):1319–1365, 2009.
- B. Cockburn, J. Gopalakrishnan, N. C. Nguyen, J. Peraire, and F.-J. Sayas. Analysis of HDG methods for Stokes flow. *Math. Comp.*, 80(274):723–760, 2011.

- M. Giacomini, R. Sevilla, and A. Huerta. Tutorial on Hybridizable Discontinuous Galerkin (HDG) formulation for incompressible flow problems. In L. De Lorenzis and A. Düster, editors, *Modeling in Engineering Using Innovative Numerical Methods for Solids and Fluids*, volume 599 of *CISM International Centre for Mechanical Sciences*, pages 163–201. Springer International Publishing, 2020. doi: 10.1007/978-3-030-37518-8_5.
- G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable Discontinuous Galerkin p-adaptivity for wave propagation problems. *Int. J. Numer. Methods Fluids*, 72(12):1244–1262, 2013a.
- G. Giorgiani, D. Modesto, S. Fernández-Méndez, and A. Huerta. High-order continuous and discontinuous Galerkin methods for wave problems. *Int. J. Numer. Methods Fluids*, 73(10):883–903, 2013b.
- G. Giorgiani, S. Fernández-Méndez, and A. Huerta. Hybridizable Discontinuous Galerkin with degree adaptivity for the incompressible Navier–Stokes equations. *Comput. Fluids*, 98:196–208, 2014.
- J.S Hesthaven and T. Warburton. Nodal high-order methods on unstructured grids: I. Time-domain solution of Maxwell’s equations. *J. Comput. Phys.*, 181(1):186–221, 2002.
- A. Huerta, A. Angeloski, X. Roca, and J. Peraire. Efficiency of high-order elements for continuous and discontinuous Galerkin methods. *Int. J. Numer. Methods Eng.*, 96(9):529–560, 2013.
- R. Kirby, S. J. Sherwin, and B. Cockburn. To CG or to HDG: A comparative study. *J. Sci. Comput.*, 51(1):183–212, 2011. ISSN 0885-7474.
- A. Montlaur, S. Fernández-Méndez, and A. Huerta. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. *Int. J. Numer. Methods Fluids*, 57(9):1071–1092, 2008.
- 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(9):3232–3254, 2009.
- N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 230(4):1147–1170, 2011.
- N.C. Nguyen, J. Peraire, and B. Cockburn. A hybridizable discontinuous Galerkin method for Stokes flow. *Comput. Methods Appl. Mech. Eng.*, 199(9-12):582–597, 2010. ISSN 0045-7825.
- Mahendra Paipuri, Sonia Fernández-Méndez, and Carlos Tiago. Comparison of high-order continuous and hybridizable discontinuous Galerkin methods in incompressible fluid flow problems. *Mathematics and Computers in Simulation*, 153:35–58, 2018. doi: <https://doi.org/10.1016/j.matcom.2018.05.012>.
- J. Peraire and P. O. Persson. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput.*, 30(4):1806–1824, 2008. ISSN 1064-8275.

Sergey Yakovlev, David Moxey, Robert Kirby, and Spencer Sherwin. To cg or to hdg: A comparative study in 3d. *Journal of Scientific Computing*, 67, 07 2015. doi: 10.1007/s10915-015-0076-6.