On weak Dirichlet boundary conditions for elliptic problems in the continuous Galerkin method

Martin Vymazal[a,*], David Moxey[b], Spencer J. Sherwin[a], Chris D. Cantwell[a], Robert M. Kirby[c]

[a] *Department of Aeronautics, Imperial College London, London, UK*
[b] *College of Engineering, Mathematics and Physical Sciences, University of Exeter, UK*
[c] *School of Computing, Univ. of Utah, Salt Lake City, UT, USA*

Abstract

We combine continuous and discontinuous Galerkin methods in the setting of a model diffusion problem. Starting from a hybrid discontinuous formulation, we replace element interiors by more general subsets of the computational domain – groups of elements that support a piecewise-polynomial continuous expansion. This step allows us to identify a new weak formulation of Dirichlet boundary condition in the continuous framework. We show that the boundary condition leads to a stable discretization with a single parameter insensitive to mesh size and polynomial order of the expansion. The robustness of the approach is demonstrated on several numerical examples.

Keywords: High-order, Navier-Stokes equations, weakly imposed boundary conditions, fluids

1. Introduction

High-order methods, combined with unstructured grids, are now becoming increasingly popular in application areas such as computational fluid dynamics. They simultaneously provide geometric flexibility and high-fidelity of flow solutions, whilst being able to utilize modern computing hardware more effectively than traditional low-order methods [1]. These properties make high-order methods particularly attractive in various application areas such as Large-Eddy Simulations (LES) over complex industrial geometries [2], which can be used to gain detailed insight into flow physics. Simulations such as these, which are based on the incompressible Navier-Stokes equations, can be efficiently tackled using discretizations which employ high-order elements in space and a time-splitting scheme [3] that involves the solution of four scalar elliptic equations for pressure and velocity components respectively. The cost of one time step in this scheme is then largely determined by the amount of work needed to obtain the pressure field, which is defined as a solution to a (frequently ill-conditioned) scalar elliptic Poisson equation.

*Corresponding author
URL: martin.vymazal@imperial.ac.uk (Martin Vymazal), d.moxey@exeter.ac.uk (David Moxey), s.sherwin@imperial.ac.uk (Spencer J. Sherwin), c.cantwell@imperial.ac.uk (Chris D. Cantwell), kirby@cs.utah.edu (Robert M. Kirby)

The difficulty in solving the governing equation for pressure is further increased when one considers complex flow features, for example the formation and evolution of a wingtip vortex simulated by a high-order method (Figure 1). Under-integration of nonlinear terms in Navier-Stokes equations introduces an aliasing error which may compromise the stability of the simulation [4]. This is usually not problematic when the flow features are adequately resolved. Once the buildup of aliasing errors becomes an issue, however, the stability of the solver is often compromised in the vicinity of (wall) boundaries, where the introduction of boundary constraints into the discrete problem further degrades the conditioning of the stiffness matrix. This problem motivated our search for a new formulation of boundary terms, which is proposed in this paper.

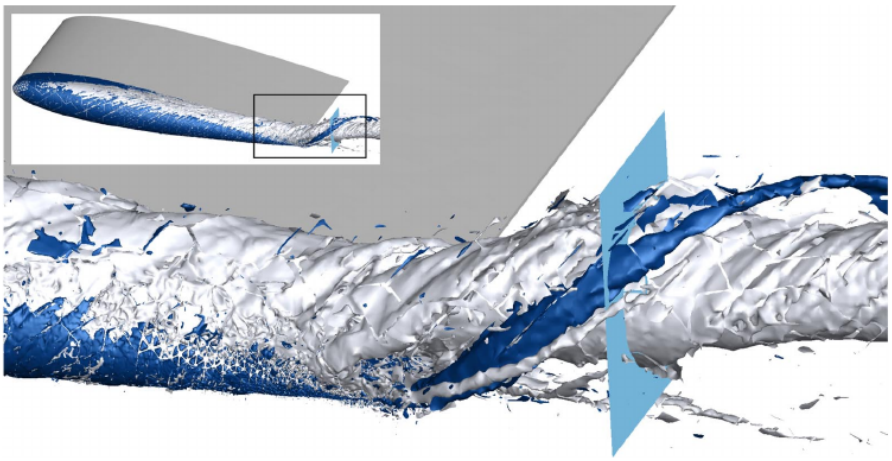


Figure 1: Implicit LES of a wingtip vortex using a high-order hp/spectral element solver. Reproduced from [2].

As noted by Hesthaven [5], in the context of hyperbolic conservation laws, if the requirement to satisfy boundary conditions exactly (within machine accuracy) is relinquished and the only requirement is that the boundary operator does not degrade the overall spatial accuracy of the discretization, a range of new techniques becomes possible. More specifically, our aim is to be able to write the discrete form of the original differential equation as a sum of an interior term and one or more boundary corrections which enforce the boundary conditions so that they are satisfied in the convergence limit, but not necessarily from the first iteration of the linear solver.

Boundary conditions in the form of penalty terms can be incorporated into the variational form using the approach first described by Nitsche [6] and later on developed in a number of other papers, for example in [7] or [8]. We compare its accuracy and performance with our method in Section 5.2.

Other work on weak imposition of Dirichlet boundary conditions includes the contributions of Bazilevs et al. [9, 10]. The authors used their formulation to solve an advection-diffusion problem and the incompressible Navier-Stokes equations with low-order stabilized finite element methods. The Dirichlet constraints are incorporated directly into the variational form as Euler-Lagrange

equations. In their paper [11], the authors note that the conditions render the simulation more robust on coarse meshes where near-wall resolution is low. Building on their previous work, Evans et al. [12] utilised Nitsche's method because it allowed them to account for sharp boundary layers in a stable and consistent manner without having to directly resolve them. They in fact claim that Nitsche's method can be interpreted as a variationally consistent wall model (based upon [9, 10, 11]).

Conceptually similar mechanisms can also be found in the works of Liakos [13], Liakos and Caglar [14], Layton [15] and Urquiza et al. [16], where the discrete form of the governing law together with weak boundary conditions is again enforced either by Lagrange multipliers or by penalizing certain components of the velocity field on wall boundaries. The existing body of work differs from our contribution in terms of scope, however. We are interested in elliptic problems and high-order Galerkin finite element methods, while the references cited above consider conservation laws of advection-diffusion type and use low-order stabilized methods.

High-order Navier-Stokes solvers on unstructured grids can be devised by considering one of many methods of Galerkin type, each of which has specific advantages and drawbacks. The high-order continuous Galerkin (CG) method [17], [18] is the oldest. Compared to its discontinuous counterparts, the unknowns at element interfaces are not duplicated, as shown in Figure 2). The CG solution can be accelerated by means of static condensation [19], [20], [21], [22], which produces a globally coupled system involving only those degrees of freedom on the mesh skeleton. The element interior unknowns are subsequently obtained from the mesh skeleton data by solving independent local problems that do not require any communication.

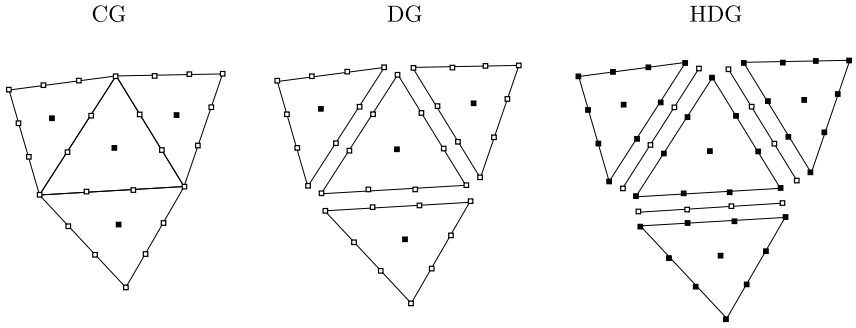


Figure 2: Distribution of unknowns for continuous and discontinuous Galerkin methods.

Discontinuous Galerkin (DG) methods [23], on the other hand, duplicate discrete variables on element boundaries, thus decoupling mesh elements and requiring at most pairwise communication between them. This is at the expense of a larger linear system and more time spent in the linear solver. Discontinuous discretization is therefore expected to scale better on parallel computers, but the improved scaling is not necessarily reflected in significantly smaller CPU times when compared to a CG solver. With regards to our work, many DG methods provide natural frameworks for the

design of weak boundary conditions implemented by means of flux functions.

Hybrid discontinuous Galerkin (HDG) methods [24] address the performance deficiencies of CG by introducing an additional (hybrid) variable on the mesh skeleton. The hybrid degrees of freedom determine the rank of the global system matrix and HDG therefore produces a statically condensed system that is similar in size to the CG case. In contrast with CG, the static condensation in HDG takes place by construction rather than being an optional iterative technique.

The formulation used for Dirichlet boundary conditions in this paper was formally derived by treating the whole domain triangulation as one element in HDG and by replacing the commonly used polynomial basis in the element interior by a piecewise-polynomial and globally continuous CG interpolant. While the mixed HDG system was our starting point, we eventually revert to the primal form of the local solver which leads to a variational formulation for a scalar elliptic PDE obtained with continuous Galerkin method and augmented by additional terms responsible for the enforcing of boundary constraints. These terms preserve the symmetry and positivity of the discrete CG operator and thus constitute and attractive alternative to existing methods. In summary, our goal is to capitalize on the properties seen when weakly imposing boundary conditions (such as in Nitsche's method [11, 12]) while leveraging the computational efficiencies seen in the Hybridized Discontinuous Galerkin Method [25, 26].

1.1. Outline

The paper is organized as follows. We first recall the mixed form of scalar elliptic problems considered in this paper and its discretization by Hybridizable Discontinuous Method in Section 2. In particular, we discuss the HDG local and global solvers which provide a natural framework for prescribing Dirichlet boundary conditions in a weak sense. Section 3 discusses the matrix form of the HDG local solver in more detail and then proceeds to the description of weak Dirichlet formulation in Section 4. Accuracy and convergence rates of the CG solver with weak boundary conditions are compared with the classical strong boundary formulation in Section 5. This Section also presents several test cases showcasing the behaviour of weak boundary conditions in a fluid dynamics context. Before concluding the paper, the differences between our formulation and existing penalty approaches for enforcing Dirichlet boundary conditions are discussed in Section 5.2.

2. Overview of the formulation of HDG method

We begin with a brief recap of the standard HDG formulation for a finite element mesh, following a similar approach to that taken in [25] and [26]. For consistency, we have selected to use the notation in [25]. Formulation details omitted here for brevity can be found therein.

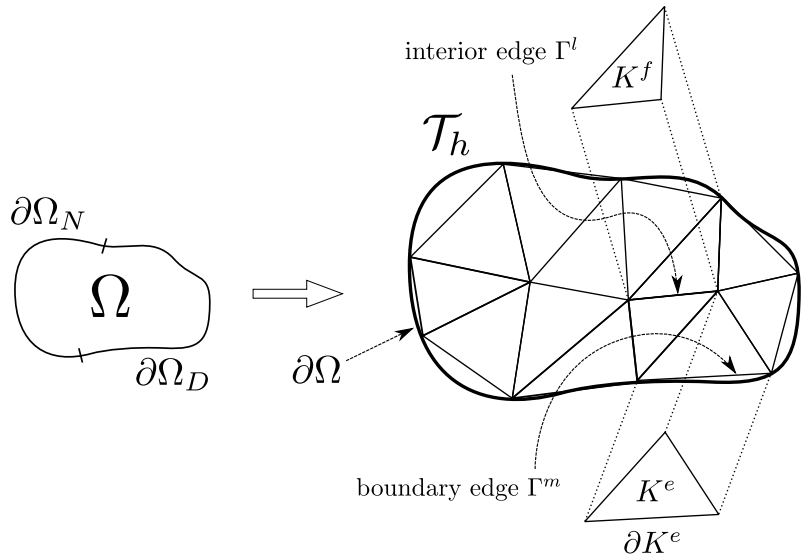


Figure 3: Computational domain and its tesselation demonstrating notation used in the text. Although not explicitly highlighted in the diagram, we assume that element edges/faces can be isoparametric.

2.1. Continuous problem

We seek the solution of a Poisson equation as a representative elliptic problem

$$\begin{aligned} -\nabla^2 u(\boldsymbol{x}) &= f(\boldsymbol{x}) && \boldsymbol{x} \in \Omega, \\ u(\boldsymbol{x}) &= g_D(\boldsymbol{x}) && \boldsymbol{x} \in \partial\Omega_D, \\ \boldsymbol{n} \cdot \nabla u(\boldsymbol{x}) &= g_N(\boldsymbol{x}) && \boldsymbol{x} \in \partial\Omega_N, \end{aligned} \tag{1}$$

on a domain Ω with Dirichlet ($\partial\Omega_D$) and Neumann ($\partial\Omega_N$) boundary conditions, where $\partial\Omega_D \bigcup \partial\Omega_N = \partial\Omega$ and $\partial\Omega_D \bigcap \partial\Omega_N = \emptyset$. To formulate the HDG method, we consider a mixed form of (1) by introducing an auxiliary variable $\boldsymbol{q} = \nabla u$:

$$-\nabla \cdot \boldsymbol{q} = f(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Omega, \tag{2}$$

$$\boldsymbol{q} = \nabla u(\boldsymbol{x}) \qquad \boldsymbol{x} \in \Omega, \tag{3}$$

$$u(\boldsymbol{x}) = g_D(\boldsymbol{x}) \qquad \boldsymbol{x} \in \partial\Omega_D, \tag{4}$$

$$\boldsymbol{q} \cdot \boldsymbol{n} = g_N(\boldsymbol{x}) \qquad \boldsymbol{x} \in \partial\Omega_N. \tag{5}$$

The gradient variable $\boldsymbol{q}$ is approximated together with the primal variable u, which contrasts with the CG method and other discontinuous methods for (1).

2.2. HDG interpolation spaces and discretization

We limit ourselves to two-dimensional problems for sake of simplicity, but the formal description remains unchanged in three dimensions. We assume that in the discrete setting, the computational domain Ω is approximated by its tesselation $\mathcal{T}_h$ consisting of non-overlapping and

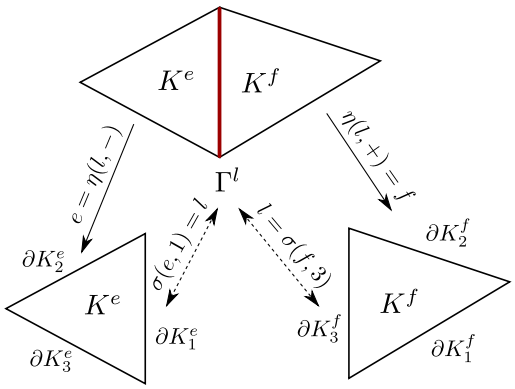


Figure 4: Index mappings relating edge and element ids.

conformal elements K^e such that for each pair of distinct indices $e_i \neq e_j$, $K^{e_i} \cap K^{e_j} = \emptyset$. The symbol Γ^l denotes an interior edge of the tesselation $\mathcal{T}_h$, i.e. an edge $\Gamma^l = \bar{K}^i \cap \bar{K}^j$ where K^i and K^j are two distinct elements of the tesselation. We say that Γ^l is a boundary edge of the tesselation $\mathcal{T}_h$ if there exists an element K^e such that $\Gamma^l = K^e \cap \partial\Omega$ and the length of Γ^l is not zero, as shown in Figure 3. The set of all internal edges is denoted by $\mathcal{E}_h^0$, while $\mathcal{E}_h^\partial$ is a set of all boundary edges. Their union $\mathcal{E}_h$ comprises of all mesh edges, $\mathcal{E}_h = \mathcal{E}_h^0 \cup \mathcal{E}_h^\partial$.

In order to describe some terms in the HDG formulation, it is also useful to introduce mappings that relate elements to their local edges, as shown in Figure 4. Let ∂K_j^e be the j-th edge of element K^e, and suppose that this is also the l-th edge Γ^l in the global edge numbering. Then we define the local-to-global edge mapping σ by setting $\sigma(e,j) = l$ so that we can write $\partial K_j^e = \Gamma^{\sigma(e,j)}$. An interior edge Γ^l is the intersection of the boundaries of two elements K^e and K^f, hence we set $\eta(l,+) = e$ and $\eta(l,-) = f$ in order to be able to write $\Gamma^l = \partial K^{\eta(l,+)} \cap \partial K^{\eta(l,-)}$.

2.3. Approximation spaces

The finite element spaces supported by the (two-dimensional) tesselation $\mathcal{T}_h$ are defined as follows:

$$
\begin{aligned}
V_h &\coloneqq \{v \in L^2(\Omega) \quad : \quad v|_{K^e} \in \mathcal{P}(K^e) \quad \forall K^e \in \mathcal{T}_h\},\\
\boldsymbol{\Sigma}_h &\coloneqq \{\boldsymbol{w} \in [L^2(\Omega)]^2 : \quad \boldsymbol{w}|_{K^e} \in \boldsymbol{\Sigma}(K^e) \quad \forall K^e \in \mathcal{T}_h\},\\
\mathcal{M}_h &\coloneqq \{\mu \in L^2(\Gamma) \quad : \quad \mu|_{\Gamma^l} \in \mathcal{P}(\Gamma^l) \quad \forall \Gamma^l \in \Gamma\},
\end{aligned}
$$

where $\mathcal{P}(\Gamma^l) = \mathcal{S}_P(\Gamma^l)$ is the polynomial space over the standard segment

$$
\mathcal{S}_P(\Gamma^l) = \{s^p \ : \ 0 \leq p \leq P, \ [x_1(s), x_2(s)] \in \Gamma^l, \ -1 \leq s \leq 1\},
$$

and $\mathcal{P}(K^e)$ is the space of polynomials of degree P defined on a standard region, which can either be the standard triangle

$$
\mathcal{P}(K^e) = \mathcal{T}_P(K^e) = \{\xi_1^p \xi_w^q \ : \ 0 \leq p+q \leq P, \ [x_1(\xi_1,\xi_2), x_2(\xi_1,\xi_2)] \in K^e, \ -1 \leq \xi_1 + \xi_2 \leq 0\},
$$

or standard quadrilateral

$$\mathcal{P}(K^e) = \mathcal{Q}_P(K^e) = \{\xi_1^p \xi_2^q \ : \ 0 \le p, q \le P, \ [x_1(\xi_1, \xi_2), (x_2(\xi_1, \xi_2)] \in K^e, \ -1 \le \xi_1, \xi_2 \le 1\}.$$

Similarly $\mathbf{\Sigma}(K^e) = [\mathcal{T}_P(K^e)]^2$ or $\mathbf{\Sigma}(K^e) = [\mathcal{Q}_P(K^e)]^2$. There is no requirement on global continuity of the expansion. This is also true for the trace space $\mathcal{M}_h$: a discrete variable $\lambda \in \mathcal{M}_h$ is multi-valued at every mesh vertex shared by multiple interior edges.

2.4. Global formulation for HDG problem

Given an element $K \in \mathcal{T}_h$ and two functions $u, v \in L^2(\mathcal{T}_h)$, we define their L^2 scalar product by

$$(u, v)_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} (u, v)_K, \quad \text{where} \quad (u, v)_K = \int_K uv \, d\boldsymbol{x}.$$

Similarly, the L^2 product of functions u and v that are square-integrable on element traces are defined by:

$$\langle u, v \rangle_{\partial \mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \langle u, v \rangle_{\partial K} \quad \text{where} \quad \langle u, v \rangle_{\partial K} = \int_{\partial K} uv \, d\boldsymbol{s}$$

The DG method seeks an approximation pair $(u^{DG}, \boldsymbol{q}^{DG})$ to u and $\boldsymbol{q}$, respectively, in the space $V_h \times \mathbf{\Sigma}_h$. The solution is required to satisfy the weak form of (2) and (3)

$$\left(\boldsymbol{q}^{DG}, \nabla v\right)_{\mathcal{T}_h} = (f, v)_{\mathcal{T}_h} + \left\langle \boldsymbol{n}^e \cdot \tilde{\boldsymbol{q}}^{DG}, v \right\rangle_{\partial \mathcal{T}_h} \tag{6}$$

$$\left(\boldsymbol{q}^{DG}, \boldsymbol{w}\right)_{\mathcal{T}_h} = -\left(u^{DG}, \nabla \cdot \boldsymbol{w}\right)_{\mathcal{T}_h} + \left\langle \tilde{u}^{DG}, \boldsymbol{w} \cdot \boldsymbol{n}^e \right\rangle_{\partial \mathcal{T}_h} \tag{7}$$

for all $(v, \boldsymbol{w}) \in V_h(\Omega) \times \mathbf{\Sigma}_h(\Omega)$, where the numerical traces $\tilde{u}^{DG}$ and $\tilde{\boldsymbol{q}}^{DG}$ have to be suitably defined in terms of the approximate solution $(u^{DG}, \boldsymbol{q}^{DG})$. For details, we refer the reader to [24]. This choice of trace variables allows us to construct the discrete HDG system involving only trace degrees of freedom $\tilde{u}^{DG}$. Once $\tilde{u}$ is known, the element-interior degrees of freedom represented by both the primal variable u and gradient $\boldsymbol{q}$ can be reconstructed from element-boundary values.

We note that the element-interior variable u restricted to element traces is not equal to the hybrid variable $\tilde{u}$, but only approximates it: due to the definition of approximation spaces V_h and $\mathcal{M}_h$, u must be continuous along element boundaries, while $\tilde{u}^{DG}$ is allowed to have jumps in element vertices.

2.5. Local solvers in the HDG method

Assume that the function

$$\lambda \coloneqq \tilde{u}^{DG} \in \mathcal{M}_h, \tag{8}$$

is given. Then the solution restricted to element K^e is a function $u^e, \boldsymbol{q}^e$ in $P(K^e) \times \boldsymbol{\Sigma}(K^e)$ that satisfies the following equations:

$$(\boldsymbol{q}^e, \nabla v)_{K^e} = (f, v)_{K^e} + \langle \boldsymbol{n}^e \cdot \tilde{\boldsymbol{q}}^e, v \rangle_{\partial K^e} \tag{9}$$

$$(\boldsymbol{q}^e, \boldsymbol{w})_{K^e} = -(u^e, \nabla \cdot \boldsymbol{w})_{K^e} + \langle \lambda, \boldsymbol{w} \cdot \boldsymbol{n}^e \rangle_{\partial K^e}, \tag{10}$$

for all $(v, \boldsymbol{w}) \in P(K^e) \times \boldsymbol{\Sigma}(K^e)$. For a unique solution of the above equations to exist, the numerical trace of the flux must depend only on λ and on $(u^e, \boldsymbol{q}^e)$:

$$\tilde{\boldsymbol{q}}^e(\boldsymbol{x}) = \boldsymbol{q}^e(\boldsymbol{x}) - \tau\big(u^e(\boldsymbol{x}) - \lambda(\boldsymbol{x})\big)\boldsymbol{n}^e \quad \text{on } \partial K^e \tag{11}$$

for some positive function τ. The analysis presented in [24] reveals that as long as $\tau > 0$, its value can be arbitrary without degrading the robustness of the solver. For the limiting value of $\tau \to \infty$, one obtains a statically condensed continuous Galerkin formulation. In this sense, τ plays the role of a method selector as opposed to traditional penalty parameter used in Nitsche's method, for example.

2.6. Global problem for trace variable

We denote by $(U_\lambda, \boldsymbol{Q}_\lambda)$ and by $(U_f, \boldsymbol{Q}_f)$ the solution to the local problem (9), (10) when $\lambda = 0$ and $f = 0$, respectively. Due to the linearity of the original problem (1) and its mixed form, the solution satisfies

$$(u^{HDG}, \boldsymbol{q}^{HDG}) = (U_\lambda, \boldsymbol{Q}_\lambda) + (U_f, \boldsymbol{Q}_f). \tag{12}$$

In order to uniquely determine λ, we require that the boundary conditions be weakly satisfied and the normal component of the numerical trace of the flux $\tilde{\boldsymbol{q}}$ given by (11) is single valued, rendering the numerical trace conservative.

We say that λ is the element of $\mathcal{M}_h$ such that

$$\lambda = \mathbb{P}_h(g_D) \quad \text{on } \partial\Omega_D \tag{13}$$

$$\langle \mu, \tilde{\boldsymbol{q}} \cdot \boldsymbol{n} \rangle_{\partial \mathcal{T}} = \langle \mu, g_N \rangle_{\partial \Omega_N}, \tag{14}$$

for all $\mu \in \mathcal{M}_h^0$ such that $\mu = 0$ on $\partial\Omega_D$. Here $\mathbb{P}_h$ denotes the L^2-projection into the space of restrictions to $\partial\Omega_D$ of functions of $\mathcal{M}_h$.

In the following, we consider $u^e(\boldsymbol{x}), \boldsymbol{q}^e(\boldsymbol{x}) = [q_1, q_2]^T$ and $\lambda^l(\boldsymbol{x})$ to be finite expansions in terms of basis functions $\phi_j^e(\boldsymbol{x})$ for the expansions over elements and the basis $\psi_j^l(\boldsymbol{x})$ over the traces of the form:

$$u^e(\boldsymbol{x}) = \sum_{j=1}^{N_u^e} \phi_j^e(\boldsymbol{x}) \underline{\hat{u}}^e[j] \qquad \boldsymbol{q}_k^e(\boldsymbol{x}) = \sum_{j=1}^{N_q^e} \phi_j^e(\boldsymbol{x}) \underline{\hat{q}}_k^e[j] \qquad \lambda^l(\boldsymbol{x}) = \sum_{j=1}^{N_\lambda^l} \psi_j^l(\boldsymbol{x}) \underline{\hat{\lambda}}^l[j]$$

3. Discrete form of HDG local solver

We now define several local matrices stemming from standard Galerkin formulation, where scalar test functions v^e are represented by $\phi_i^e(\boldsymbol{x})$, with $i = 1, \ldots, N_u^e$.

$$\mathbb{D}_k^e[i,j] = \left(\phi_i^e, \frac{\partial \phi_j^e}{\partial x_k}\right)_{K_e} \qquad \mathbb{M}^e[i,j] = \left(\phi_i^e, \phi_j^e\right)_{K_e}$$

$$\mathbb{E}_l^e[i,j] = \left\langle \phi_i^e, \phi_j^e \right\rangle_{\partial K_l^e} \qquad \widetilde{\mathbb{E}}_{kl}^e[i,j] = \left\langle \phi_i^e, \phi_j^e n_k^e \right\rangle_{\partial K_l^e}$$

$$\mathbb{F}_l^e[i,j] = \left\langle \phi_i^e, \psi_j^{\sigma(e,l)} \right\rangle_{\partial K_l^e} \qquad \widetilde{\mathbb{F}}_{kl}^e[i,j] = \left\langle \phi_i^e, \psi_j^{\sigma(e,l)} n_k^e \right\rangle_{\partial K_l^e}$$

If the trace expansion matches the expansions used along the edge of the elemental expansion and the local coordinates are aligned, that is $\psi_i^{\sigma(e,l)}(s) = \phi_{k(i)}(s)$ then $\mathbb{E}_l^e$ contains the same entries as $\mathbb{F}_l^e$ and similarly $\widetilde{\mathbb{E}}_{kl}^e$ contains the same entries as $\widetilde{\mathbb{F}}_{kl}^e$.

Inserting the finite expansions of the trial functions into equations (9) and (10), and using the definition of the flux (11) yields the matrix form of *local solvers*

$$\sum_{k=1,2} \left\{ (\mathbb{D}_k^e)^T - \sum_{l=1}^{N_b^e} \left[\widetilde{\mathbb{E}}_{kl}^e\right] \right\} \underline{\hat{q}}_k^e + \sum_{l=1}^{N_b^e} \tau^{e,l} \left[\mathbb{E}_l^e \underline{\hat{u}}^e - \mathbb{F}_l^e \underline{\hat{\lambda}}^{\sigma(e,l)} \right] = \underline{f}^e \tag{15}$$

$$\mathbb{M}^e \underline{\hat{q}}_k^e = -(\mathbb{D}_k^e)^T \underline{\hat{u}}^e + \sum_{l=1}^{N_b^e} \widetilde{\mathbb{F}}_{kl}^e \underline{\hat{\lambda}}^{\sigma(e,l)} \qquad k = 1,2 \tag{16}$$

The *global equation for* λ can be obtained by discretizing the transmission condition (14). We introduce local element-based and edge-based matrices

$$\overline{\mathbb{F}}^{l,e}[i,j] = \left\langle \psi_i^l, \phi_j^e \right\rangle_{\Gamma^l} \qquad \underset{\sim}{\overline{\mathbb{F}}}_k^{l,e}[i,j] = \left\langle \psi_i^l, \phi_j^e n_k^e \right\rangle_{\Gamma^l} \qquad \bar{\mathbb{G}}^l[i,j] = \left\langle \psi_i^l, \psi_j^l \right\rangle_{\Gamma^l}$$

and define

$$\underline{g}_N^l[i] = \left\langle g_n, \psi_i^l \right\rangle_{\Gamma^l \bigcap \partial\Omega_N}.$$

The transmission condition in matrix form is then

$$\begin{bmatrix} \underset{\sim}{\overline{\mathbb{F}}}_1^{l,e} & \underset{\sim}{\overline{\mathbb{F}}}_2^{l,e} \end{bmatrix} \begin{bmatrix} \underline{\hat{q}}_1^e \\ \\ \underline{\hat{q}}_2^e \end{bmatrix} + \begin{bmatrix} \underset{\sim}{\overline{\mathbb{F}}}_1^{l,f} & \underset{\sim}{\overline{\mathbb{F}}}_2^{l,f} \end{bmatrix} \begin{bmatrix} \underline{\hat{q}}_1^f \\ \\ \underline{\hat{q}}_2^f \end{bmatrix} + (\tau^{e,i} + \tau^{f,j}) \bar{\mathbb{G}}^l \underline{\hat{\lambda}}^l - \tau^{e,i} \bar{\mathbb{F}}^{l,e} \underline{u}^e - \tau^{f,j} \bar{\mathbb{F}}^{l,f} \underline{u}^f = \underline{g}_N^l,$$

where we are assuming that $l = \sigma(e,i) = \sigma(f,j)$.

We see that the transmission condition can be constructed from elemental contributions. In Section 4 of [25], it was shown how to use the elemental local solvers given by the equations above to obtain a matrix equation for λ only. Due to space considerations, we refer the reader to [25] for further details.

4. Continuous finite elements with weak Dirichlet boundary conditions

With the standard HDG formulation now outlined, we investigate how this approach can be applied to derive a weak Dirichlet boundary condition implementation for a continuous Galerkin problem. Since the HDG local solver naturally imposes a weak boundary condition on a single element, we choose to apply the HDG local solver to a single 'macro element' that covers the whole domain tesselation $\mathcal{T}_h$. The term 'macro element' in this setting denotes a conformal triangulation (as described in section 2) which supports a piecewise polynomial expansion, as is generally common in Galerkin methods.

We start again from the weak mixed problem (9), (10), but integrate the second term in the flux equation (10) by parts once again. This modified flux form allows for a symmetric boundary contribution to the linear system as will be explained shortly. In order to distinguish between the standard HDG local solver within a single element and HDG applied to the whole domain tesselation $\mathcal{T}_h$, the superscript 'e' has been replaced by $\mathcal{T}_h$ where appropriate. The 'macro element' form yields a system

$$\left(\boldsymbol{q}^{\mathcal{T}_h}, \nabla v\right)_{\mathcal{T}_h} = (f, v)_{\mathcal{T}_h} + \left\langle \boldsymbol{n}^{\mathcal{T}_h} \cdot \tilde{\boldsymbol{q}}^{\mathcal{T}_h}, v \right\rangle_{\partial \mathcal{T}_n} \tag{17}$$

$$\left(\boldsymbol{q}^{\mathcal{T}_h}, \boldsymbol{w}\right)_{\mathcal{T}_h} = \left(\nabla u^{\mathcal{T}_h}, \boldsymbol{w}\right)_{\mathcal{T}_h} - \left\langle u^{\mathcal{T}_h}, \boldsymbol{w} \cdot \boldsymbol{n}^{\mathcal{T}_h} \right\rangle_{\partial \mathcal{T}_h} + \left\langle \lambda, \boldsymbol{w} \cdot \boldsymbol{n}^{\mathcal{T}_h} \right\rangle_{\partial \mathcal{T}_h} \tag{18}$$

The numerical approximation $u^{\mathcal{T}_h}$ belongs to the space $V_h^{\mathcal{T}_h}$ and $\boldsymbol{q}^{\mathcal{T}_h}$ lies in $\boldsymbol{\Sigma}_h^{\mathcal{T}_h}$, which are defined as

$$\begin{aligned} V_h^{\mathcal{T}} &\coloneqq \{v \in \mathcal{C}^0(\Omega) \quad : \quad v|_{K^e} \in P(K^e) \quad \forall K^e \in \mathcal{T}_h\}, \\ \boldsymbol{\Sigma}_h^{\mathcal{T}} &\coloneqq \{\boldsymbol{w} \in [L^2(\Omega)]^2 : \quad \boldsymbol{w}|_{K^e} \in \boldsymbol{\Sigma}(K^e) \quad \forall K^e \in \mathcal{T}_h\}. \end{aligned}$$

Using the definition of the trace flux

$$\tilde{\boldsymbol{q}}^{\mathcal{T}_h}(\boldsymbol{x}) = \boldsymbol{q}^{\mathcal{T}_h}(\boldsymbol{x}) - \tau(u^{\mathcal{T}_h}(\boldsymbol{x}) - \lambda(\boldsymbol{x}))\boldsymbol{n}^{\mathcal{T}_h} \quad \text{on } \partial\mathcal{T}_h,$$

and the fact that the integral over $\mathcal{T}_h$ can be written as a sum of integrals over all $K^e \in \mathcal{T}_h$, equations (17) and (18) become

$$\begin{aligned} &\sum_{K^e \in \mathcal{T}_h} \left(\nabla v, \boldsymbol{q}^e\right)_{K^e} - \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \left\langle v, \boldsymbol{n}^e \cdot \boldsymbol{q}^e \right\rangle_{\partial K^e} + \tau \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \left\langle v, u^e \right\rangle_{\partial K^e} \\ -\tau &\sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \left\langle v, \lambda \right\rangle_{\partial K^e} = \sum_{K^e \in \mathcal{T}_h} (v, f)_{K^e} \end{aligned} \tag{19}$$

$$\begin{aligned} &\sum_{K^e \in \mathcal{T}_h} \left(\boldsymbol{w}, \boldsymbol{q}^e\right)_{K^e} + \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \left\langle u^e, \boldsymbol{w} \cdot \boldsymbol{n}^e \right\rangle_{\partial K^e} \\ &- \sum_{K^e \in \mathcal{T}_h} \left(\boldsymbol{w}, \nabla u^e\right)_{K^e} - \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \left\langle \boldsymbol{w} \cdot \boldsymbol{n}^e, \lambda \right\rangle_{\partial K^e} = 0 \end{aligned} \tag{20}$$

A continuous Galerkin solver with Dirichlet data prescribed by the variable λ can be obtained by eliminating the flux variable from the system and reverting back to primal form for the unknown u. The mass matrix which appears in the second equation of the local solver after evaluating the dot product $\left(\boldsymbol{q}^{\mathcal{T}_h}, \boldsymbol{w}\right)_{\mathcal{T}_h}$ is now block-diagonal as a consequence of the discontinuous nature of the discrete flux $\boldsymbol{q}^{\mathcal{T}_h}$, hence the elimination of $\boldsymbol{q}^{\mathcal{T}_h}$ from the system can be performed element-wise. The matrix equivalent of (19), (20) written for a single element $K^e \in \mathcal{T}_h$ adjacent to Dirichlet boundary reads

$$\sum_{k=1,2} \left\{ (\mathbb{D}_k^e)^T - \sum_{l=1}^{N_b^e} \widetilde{\mathbb{E}}_{kl}^e \right\} \underline{\hat{q}}_k^e + \sum_{l=1}^{N_b^e} \tau^{e,l} \left[\mathbb{E}_l^e \underline{\hat{u}}^e - \mathbb{F}_l^e \underline{\hat{\lambda}}^{\sigma(e,l)} \right] = \underline{f}^e \tag{21}$$

$$\mathbb{M}^e \underline{\hat{q}}_k^e = \left\{ (\mathbb{D}_k^e) - \sum_{l=1}^{N_b^e} \widetilde{\mathbb{E}}_{kl}^e \right\} \underline{\hat{u}}^e + \sum_{l=1}^{N_b^e} \widetilde{\mathbb{F}}_{kl}^e \underline{\hat{\lambda}}^{\sigma(e,l)} \qquad k = 1, 2 \tag{22}$$

The discrete flux $\underline{\hat{q}}_k^e$ expressed from (22) and substituted in equation (21) yields element-wise contribution to the left- and right-hand side of the linear system which can be expressed as

$$\begin{aligned}
&\sum_{k=1,2} \Bigg\{ \underbrace{(\mathbb{D}_k^e)^T \left(\mathbb{M}^e\right)^{-1} \mathbb{D}_k^e}_{\boxed{1}} - \underbrace{\left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{E}}_{kl}^e \right) \left(\mathbb{M}^e\right)^{-1} \mathbb{D}_k^e}_{\boxed{2a}} \Bigg\} \underline{\hat{u}}^e \\
&- \sum_{k=1,2} \Bigg\{ \underbrace{(\mathbb{D}_k^e)^T \left(\mathbb{M}^e\right)^{-1} \left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{E}}_{kl}^e \right)}_{\boxed{2b}} + \underbrace{\left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{E}}_{kl}^e \right) \left(\mathbb{M}^e\right)^{-1} \left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{E}}_{kl}^e \right)}_{\boxed{3}} \Bigg\} \underline{\hat{u}}^e + \underbrace{\sum_{l=1}^{N_b^e} \tau^{(e,l)} \mathbb{E}_l^e \underline{\hat{u}}^e}_{\boxed{4}} \\
&= \underline{\boldsymbol{f}}^e + \sum_{k=1,2} \left\{ \left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{E}}_{kl}^e - (\mathbb{D}_k^e)^T \right) \left(\mathbb{M}^e\right)^{-1} \left(\sum_{l=1}^{N_b^e} \tilde{\mathbb{F}}_{kl}^e \underline{\hat{\lambda}}^{\sigma(e,l)} \right) \right\} + \sum_{l=1}^{N_b^e} \tau^{(e,l)} \mathbb{F}_l^e \underline{\hat{\lambda}}^{\sigma(e,l)}
\end{aligned}$$

Term $\boxed{1}$ on the left-hand side is a discrete Laplacian that arises from the standard continuous Galerkin discretization, which would typically be accompanied by the forcing term $\underline{\boldsymbol{f}}^e$ on the right hand side. This new expression therefore denotes a modification of the existing matrix system and right hand side, which makes implementation relatively straightforward. The matrix expressions $\boxed{2a}$, $\boxed{2b}$, $\boxed{3}$ and $\boxed{4}$ appear in the formulation only for elements K^e containing at least one edge on Dirichlet boundary of Ω. In addition, expressions $\boxed{3}$ and $\boxed{4}$ are symmetric as a consequence of symmetry of $\tilde{\mathbb{E}}_{kl}^e, \mathbb{E}_l^e$ and $\left(\mathbb{M}^e\right)^{-1}$. The products $\boxed{2a}$ and $\boxed{2b}$ are transposes of each other, hence their sum is again symmetric. The modifications to the symmetric discrete Laplacian therefore preserve symmetry of the discrete weak form, meaning that efficient iterative solvers such as the conjugate gradient method can be used to obtain solutions.

When the domain trace λ and forcing term $\boldsymbol{f}$ are both zero, the bilinear forms (19), (20) yield a homogeneous linear system with a regular matrix. This can be shown by testing the two forms

with $v = u$ and $\boldsymbol{w} = \boldsymbol{q}$:

$$\sum_{K^e \in \mathcal{T}_h} (\nabla u^e, \boldsymbol{q}^e)_{K^e} - \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \langle u^e, \boldsymbol{n}^e \cdot \boldsymbol{q}^e \rangle_{\partial K^e} + \tau \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \langle u^e, u^e \rangle_{\partial K^e} = 0$$

$$\sum_{K^e \in \mathcal{T}_h} (\boldsymbol{q}^e, \boldsymbol{q}^e)_{K^e} + \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \langle u^e, \boldsymbol{q}^e \cdot \boldsymbol{n}^e \rangle_{\partial K^e} - \sum_{K^e \in \mathcal{T}_h} (\boldsymbol{q}^e, \nabla u^e)_{K^e} = 0$$

Their sum

$$\sum_{K^e \in \mathcal{T}_h} (\boldsymbol{q}^e, \boldsymbol{q}^e)_{K^e} + \tau \sum_{\substack{K^e \\ \partial K^e \cap \partial \mathcal{T}_{h,D} \neq \emptyset}} \langle u^e, u^e \rangle_{\partial K^e} = 0$$

has a unique solution $\boldsymbol{q} = \boldsymbol{0}$ and $u = 0$ provided $\tau > 0$. This means that the CG system with weakly imposed Dirichlet boundary conditions is uniquely solvable.

5. Results

In this section, we apply the weak Dirichlet boundary conditions to various elliptic problems. We demonstrate that this technique preserves the expected convergence properties of high-order methods, and then apply it to a standard fluid dynamics test case in order to showcase its use in a more realistic application.

5.1. Convergence of continuous Galerkin solver with weak boundary conditions

We first present a straightforward evaluation of the convergence properties of weakly imposed Dirichlet boundary conditions on a scalar Helmholtz problem

$$\nabla^2 u - \lambda u = f \tag{23}$$

in a square domain $(-1, 1)^2$ with $\lambda = 1$ and $f(x, y)$ chosen so that the exact solution is of the form

$$u(x, y) = \sin(10\pi x) \cos(10\pi y) + x + y \tag{24}$$

Two meshes were considered: a structured Cartesian grid and an unstructured mesh consisting of triangles. Figure 6 compares the L^2 error for polynomial orders varying between 1 and 20 when the Dirichlet boundary conditions are imposed strongly and weakly. The behaviour of both strong and weak methods produces nearly identical errors up to $p = 12$ on the structured grid and $p = 11$ on triangles. With further increase of polynomial degree of the basis, however, the weak errors fail to further decrease. The observed differences are not surprising, because the HDG-based algorithm only penalizes the solution in order to satisfy boundary conditions, while the strong implementation completely eliminates known degrees of freedom and moves them to the right-hand side of the linear system, thus fulfilling the boundary conditions exactly by construction. Furthermore, the stiffness matrix with weak constraints is larger, hence less favourably conditioned

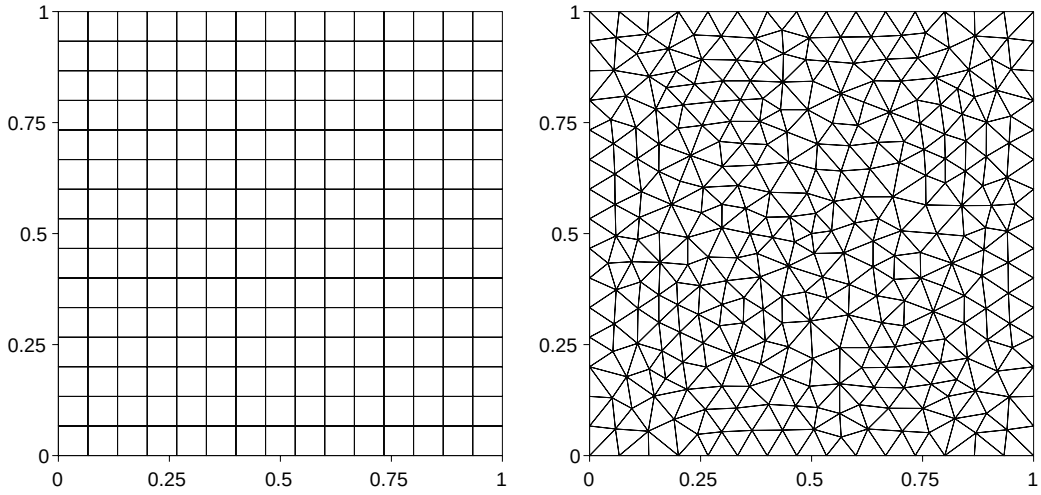
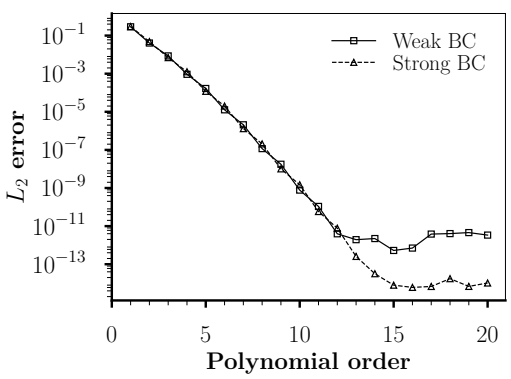
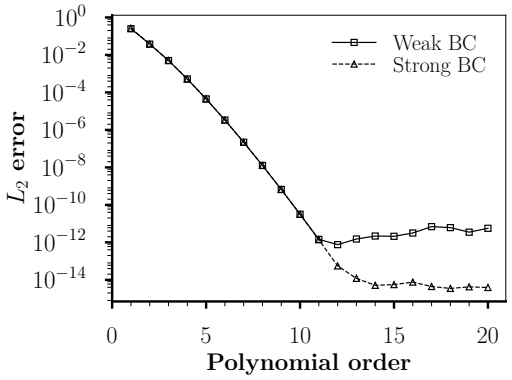


Figure 5: Meshes for Helmholtz convergence test.



(a) Convergence to exact solution, strong vs. weak boundary conditions on unstructured triangular mesh.



(b) Convergence to exact solution, strong vs. weak boundary conditions on quadrilateral mesh.

Figure 6: Convergence to the exact solution in L_2 norm.

and round-off errors in the linear solver become important as the error values approach the limits of finite-precision arithmetic on a given machine.

5.2. Comparison with classical penalty techniques

We now provide a comparison of our method against an existing technique for imposing weak boundary conditions. The main motivation for this example is to highlight a known drawback of the classic formulation: that the penalty parameter τ_N is problem-dependent. A classical penalty approach for boundary conditions in finite element methods is due to Nitsche [6]. Consider a Poisson equation

$$-\nabla^2 u(\boldsymbol{x}) = f(\boldsymbol{x}) \quad \boldsymbol{x} \text{ in } \Omega$$
$$u(\boldsymbol{x}) = g_D(\boldsymbol{x}) \quad \boldsymbol{x} \in \partial\Omega,$$

Multiplying both sides of the equation by a test function v and adding a term $\langle u - g_D, \nabla v \cdot \boldsymbol{n} \rangle_{\partial\Omega}$ which should vanish for u satisfying the boundary condition yields

$$(\nabla u, \nabla v)_\Omega - \langle \nabla u \cdot \boldsymbol{n}, v \rangle_{\partial\Omega} - \langle u - g_D, \nabla v \cdot \boldsymbol{n} \rangle_{\partial\Omega} = (f, v)_\Omega$$

A coercive bilinear form can be obtained by adding a penalty term $\tau_N \langle u - g_D, v \rangle_{\partial\Omega}$ which should again be equal to zero for the exact solution. Nitsche's method is therefore defined as: find $u_h \in V_h$ such that

$$\mathcal{B}(u_h, v) = \mathcal{F}(v) \quad \forall v \in V_h, \tag{25}$$

where

$$\begin{aligned} \mathcal{B}(u, v) &= (\nabla u, \nabla v)_\Omega - \langle v, \nabla u \cdot \boldsymbol{n} \rangle_{\partial\Omega} - \langle u, \nabla v \cdot \boldsymbol{n} \rangle_{\partial\Omega} + \tau_N \langle u, v \rangle_{\partial\Omega}, \\ \mathcal{F}(v) &= (f, v)_\Omega - \langle g, \nabla v \cdot \boldsymbol{n} \rangle_{\partial\Omega} + \tau_N \langle g, v \rangle_{\partial\Omega} \end{aligned}$$

and

$$V_h \coloneqq \{v \in H^1(\Omega) : \, v|_{K^e} \in P(K^e) \, \forall K^e \in \mathcal{T}_h\}.$$

The main drawback of the above formulation is that the penalty parameter τ_N is problem-dependent; estimates are discussed in more detail in papers [7] or [8].

To demonstrate the differences in behaviour of (25) and our method in a concrete setting, we solve a two-dimensional Laplace problem with exact solution given by

$$u = \sin(10\pi x)\cos(10\pi y) + x + y, \tag{26}$$

i.e. the same exact solution as in Section 5.1. The numerical approximation was represented by Lagrange finite elements with polynomial degree varying between 1 and 7 and we solved the underlying linear system by a preconditioned conjugate gradient (PCG) method with algebraic multigrid as preconditioner. Each computation was required to reach a relative tolerance threshold of 10^{-9}.

The results summarized in Table 1 show that the weak boundary algorithm has little sensitivity to values of τ with respect to the obtained L_2 errors. Large values of the penalty parameter help reduce the number of PCG iterations by approximately 10%. Nitsche's method, on the other hand, yields larger variations in L_2 errors when the penalty parameter is changed, and this can be observed even for low orders. Too low values of τ_N initially lead to larger errors and with increasing p eventually prevent the method from converging.

The situation is different when an anisotropic mesh such as the one depicted in Figure 7 is considered. Despite the fact that errors computed with weak boundary conditions are now larger, the formulation remains stable and yields consistent results across a range of values of τ.

Nitsche's method, however, lacks in robustness in this case. Table 2b contains two different values of stabilization parameter τ for each polynomial order: the first is chosen as the smallest

	Value of τ in weak BCs						Value of τ_N in Nitsche's method			
	10^{-6}		1		10^{6}		10^{6}		10^{8}	
p	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$
1	17	$3.15 \cdot 10^{-1}$	17	$3.15 \cdot 10^{-1}$	15	$3.29 \cdot 10^{-1}$	15	$4.18 \cdot 10^{-1}$	14	$4.15 \cdot 10^{-1}$
2	23	$4.51 \cdot 10^{-2}$	23	$4.51 \cdot 10^{-2}$	21	$4.66 \cdot 10^{-2}$	22	$5.42 \cdot 10^{-2}$	20	$5.25 \cdot 10^{-2}$
3	35	$2.90 \cdot 10^{-3}$	35	$2.89 \cdot 10^{-3}$	31	$2.47 \cdot 10^{-3}$	32	$8.68 \cdot 10^{-3}$	29	$5.06 \cdot 10^{-3}$
4	49	$6.34 \cdot 10^{-4}$	49	$6.34 \cdot 10^{-4}$	46	$6.39 \cdot 10^{-4}$	47	$1.17 \cdot 10^{-2}$	42	$6.79 \cdot 10^{-4}$
5	66	$1.45 \cdot 10^{-4}$	66	$1.45 \cdot 10^{-4}$	61	$1.54 \cdot 10^{-4}$	–	–	55	$1.64 \cdot 10^{-4}$
6	95	$1.06 \cdot 10^{-5}$	95	$1.06 \cdot 10^{-5}$	87	$1.09 \cdot 10^{-5}$	–	–	79	$5.04 \cdot 10^{-5}$
7	141	$2.08 \cdot 10^{-6}$	141	$2.08 \cdot 10^{-6}$	128	$2.19 \cdot 10^{-6}$	–	–	116	$5.17 \cdot 10^{-5}$

Table 1: Iterative convergence and L_2 errors for different values of penalty parameters in weak boundary conditions and Nitsche's method. Missing entries in the table indicate cases where Nitsche's algorithm did not converge. N_{PMG} represents the number of iterations in preconditioned conjugate gradient solver using an algebraic multigrid solver as preconditioner and $\|e\|_{L_2}$ is the obtained L_2 error norm.

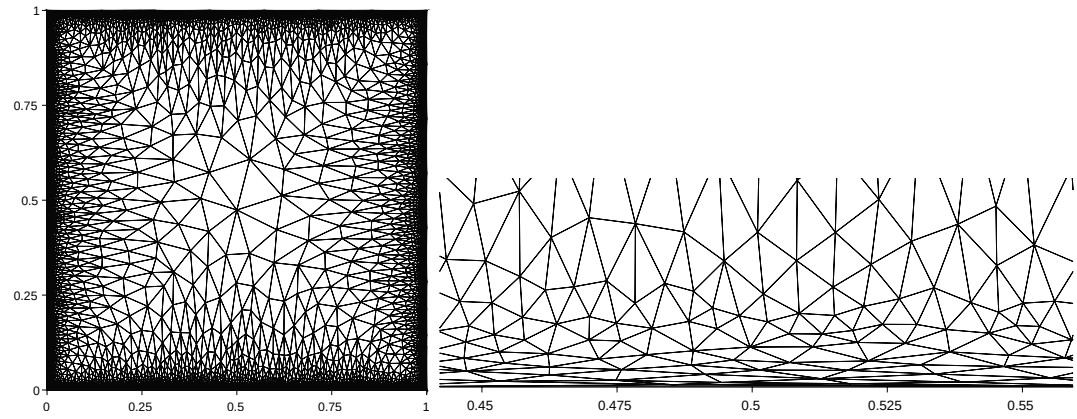


Figure 7: Anisotropic mesh of a unit square.

power of 10 for which the method converged, the second is then equal to the first multiplied by 10^5. This is to demonstrate that not even a significant increase of the stabilization parameter helps reduce the error.

	Value of τ in weak BCs					
	10^{-6}		1		10^6	
p	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$	N_{AMG}	$\|e\|_{L_2}$
1	19	$3.73 \cdot 10^{-1}$	19	$3.73 \cdot 10^{-1}$	19	$3.73 \cdot 10^{-1}$
2	37	$8.44 \cdot 10^{-2}$	37	$8.44 \cdot 10^{-2}$	37	$8.44 \cdot 10^{-2}$
3	57	$1.34 \cdot 10^{-2}$	57	$1.34 \cdot 10^{-2}$	57	$1.34 \cdot 10^{-2}$
4	81	$3.44 \cdot 10^{-3}$	81	$3.44 \cdot 10^{-3}$	81	$3.44 \cdot 10^{-3}$
5	112	$6.84 \cdot 10^{-4}$	112	$6.84 \cdot 10^{-4}$	112	$6.84 \cdot 10^{-4}$
6	166	$1.24 \cdot 10^{-4}$	166	$1.24 \cdot 10^{-4}$	166	$1.24 \cdot 10^{-4}$
7	252	$2.28 \cdot 10^{-5}$	252	$2.28 \cdot 10^{-5}$	252	$2.28 \cdot 10^{-5}$

(a) Iterative convergence and L_2 errors for different values of penalty parameters in weak boundary conditions on anisotropic mesh.

p	τ_N	N_{AMG}	$\|e\|_{L_2}$
1	10^{14}	18	1102.71
1	10^{19}	14	927.93
2	10^{15}	30	22.76
2	10^{20}	22	22.53
3	10^{15}	47	55.57
3	10^{20}	32	54.17
4	10^{15}	64	3.05
4	10^{20}	46	2.89
5	10^{16}	90	1.53
5	10^{21}	58	1.52

(b) Iterative convergence and L_2 errors for Nitsche's method on anisotropic mesh.

Table 2: Performance of Nitsche's method and weak boundary conditions on anisotropic mesh.

5.3. Navier-Stokes Results: NACA 6412

The incompressible flow past a NACA6412 airfoil was used to evaluate the performance of weak boundary conditions when computing derived quantities such as aerodynamic forces. The airfoil is placed in the flow with angle of attack $\alpha = 15^\circ$ and the Reynolds number based on chord length is $Re = 500$. No-slip condition on airfoil surface was imposed weakly, while inlet values were prescribed using the classical strong algorithm. The simulation ran for 20,000 time steps with $\Delta t = 5 \times 10^{-3}$ using a velocity correction scheme implemented in the open-source library Nektar++ [27]. The obtained flow field at $t = 10$ is plotted in Figure 8 for both strong and weak boundary conditions. Note that the flow fields look visually identical; however, the legends indicate the minor differences incurred to the differences in the imposition of the boundary conditions.

We compared lift and drag computed on the same mesh using a modal expansion with degrees 4 and 8 (Figure 9). In both cases, the forces computed with weak and strong approach are in excellent agreement.

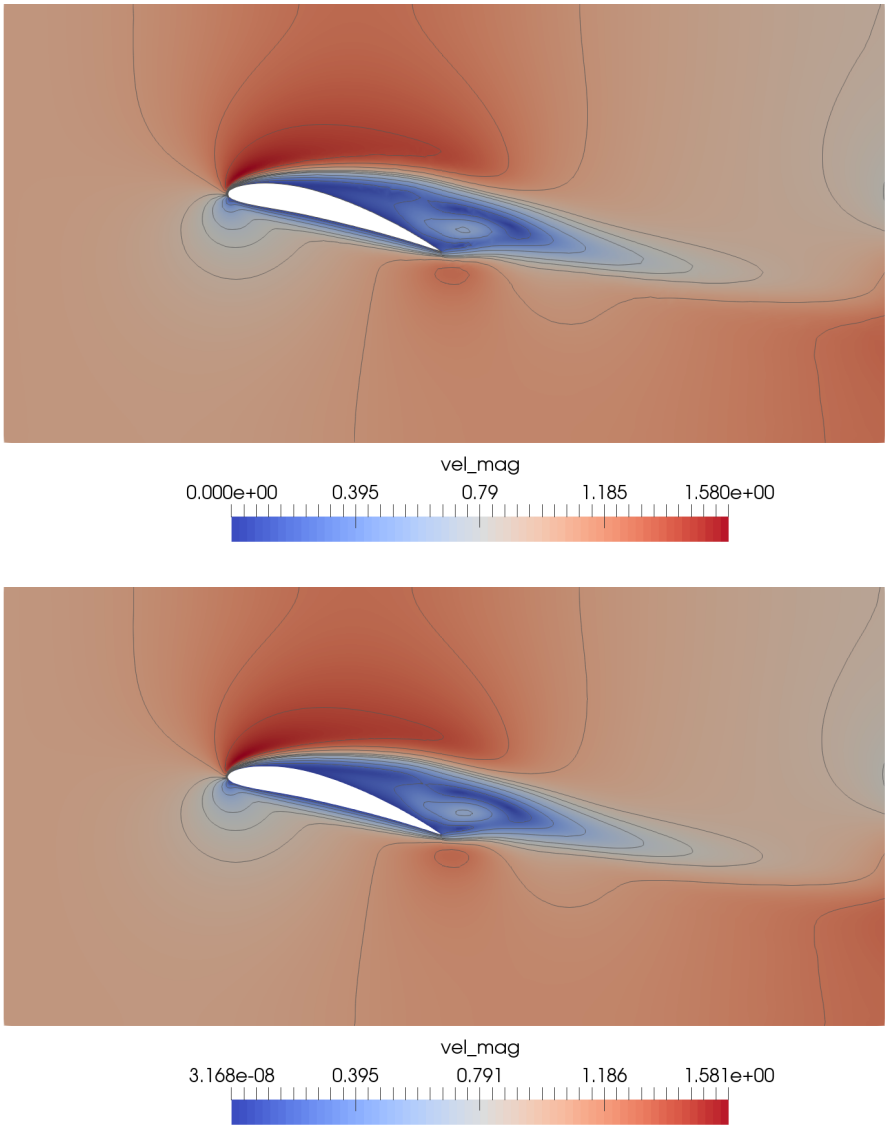


Figure 8: Viscous incompressible flow past NACA airfoil: velocity magnitude obtained with strong Dirichlet boundary conditions imposed on the airfoil surface (top) and with weak boundary conditions (bottom).

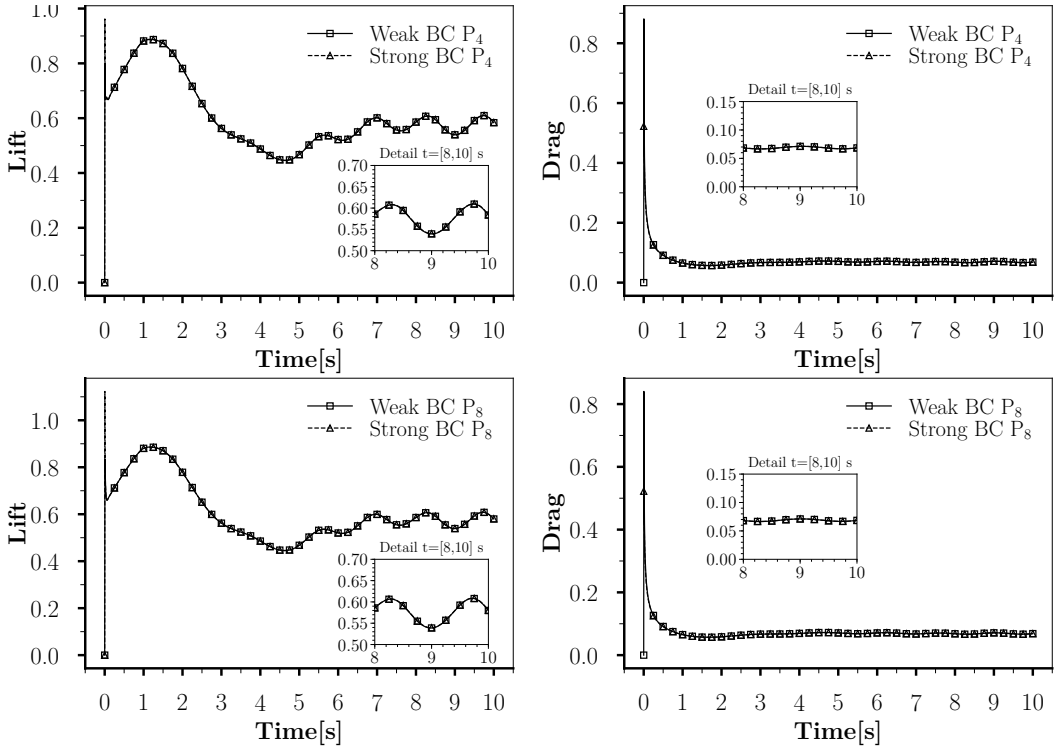


Figure 9: Viscous incompressible flow past NACA airfoil: drag and lift on P_4 elements (top) and P_8 elements (bottom).

6. Conclusion

This paper proposes a new method for the imposition of Dirichlet boundary conditions for elliptic PDEs of Helmholtz type which enforces the constraints weakly, i.e. by amending the underlying weak form with penalty terms instead of lifting known boundary values from the linear system.

The presented technique is conceptually based on hybrid Discontinuous Galerkin method, but replaces the polynomial space typically used in element interiors (a finite element basis defined in single element) by a piecewise continuous multi-element Galerkin expansion. The weak form augmented by newly derived penalty terms has all desirable properties for this particular class of problems: symmetry, positivity and very low sensitivity to underlying mesh and the polynomial degree of finite element basis. The original penalty parameter introduced in HDG acts rather as a method selector with limited effect on iterative convergence and accuracy. The weak boundary algorithm inherits this characteristic and the positivity of τ leads to a well-posed problem and its efficient discretization. This contrasts with Nitsche's method, where the failure to properly scale the penalty term yields an ill-defined problem.

Our future work will focus on application of weak boundary conditions in more challenging set ups featuring more complex geometries and turbulent flows.

Acknowledgments

The authors acknowledge support by a European Commission Horizon 2020 project grant entitled "ExaFLOW: Enabling Exascale Fluid Dynamics Simulations" (grant reference 671571). RMK additionally acknowledges support from the US National Science Foundation under DMS-1521748.

References

[1] Z. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, M. Visbal, High-order cfd methods: current status and perspective, International Journal for Numerical Methods in Fluids 72 (8) (2013) 811–845. doi:10.1002/fld.3767.
URL http://dx.doi.org/10.1002/fld.3767

[2] J.-E. W. Lombard, D. Moxey, S. J. Sherwin, J. F. Hoessler, S. Dhandapani, M. J. Taylor, Implicit large-eddy simulation of a wingtip vortex, AIAA Journal (2015).

[3] G. E. Karniadakis, M. Israeli, S. A. Orszag, High-order splitting methods for the incompressible navier-stokes equations, Journal of computational physics 97 (2) (1991) 414–443.

[4] R. M. Kirby, G. E. Karniadakis, De-aliasing on non-uniform grids: algorithms and applications, Journal of Computational Physics 191 (1) (2003) 249–264.

[5] J. S. Hesthaven, Spectral penalty methods, Applied Numerical Mathematics 33 (1-4) (2000) 23–41.

[6] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, in: Abhandlungen aus dem mathematischen Seminar der Universität Hamburg, Vol. 36, Springer, 1971, pp. 9–15.

[7] J. Freund, R. Stenberg, On weakly imposed boundary conditions for second order problems, in: Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice, 1995, pp. 327–336.

[8] M. Juntunen, R. Stenberg, Nitsche's method for general boundary conditions, Mathematics of computation 78 (267) (2009) 1353–1374.

[9] Y. Bazilevs, T. Hughes, Weak imposition of dirichlet boundary conditions in fluid mechanics, Computers and Fluids 36 (2007) 12–36.

[10] Y. Bazilevs, T. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, Computers & Fluids 36 (1) (2007) 12 – 26, challenges and Advances in Flow Simulation and

Modeling. doi:https://doi.org/10.1016/j.compfluid.2005.07.012.
URL http://www.sciencedirect.com/science/article/pii/S0045793005001258

[11] Y. Bazilevs, C. Michler, V. Calo, T. Hughes, Weak dirichlet boundary conditions for wall-bounded turbulent flows, Computer Methods in Applied Mechanics and Engineering 196 (49) (2007) 4853 – 4862. doi:https://doi.org/10.1016/j.cma.2007.06.026.
URL http://www.sciencedirect.com/science/article/pii/S0045782507002885

[12] J. A. Evans, T. J. Hughes, Isogeometric divergence-conforming B-Splines for the steady Navier-Stokes equations, Mathematical Models and Methods in Applied Sciences 23 (8) (2013) 1421–1478.

[13] A. Liakos, Weak imposition of boundary conditions in the stokes problem, Tech. rep., University of Pittsburgh (1999).

[14] A. Caglar, A. Liakos, Weak imposition of boundary conditions for the navier–stokes equations by a penalty method, International journal for numerical methods in fluids 61 (4) (2009) 411–431.

[15] W. Layton, Weak imposition of "no-slip" conditions in finite element methods, Computers & Mathematics with Applications 38 (5-6) (1999) 129–142.

[16] J. M. Urquiza, A. Garon, M.-I. Farinas, Weak imposition of the slip boundary condition on curved boundaries for stokes flow, Journal of Computational Physics 256 (2014) 748–767.

[17] S. Brenner, R. Scott, The mathematical theory of finite element methods, 3rd Edition, Vol. 15, Springer Science & Business Media, 2007.

[18] P. G. Ciarlet, The finite element method for elliptic problems, SIAM, 2002.

[19] M. Ainsworth, S. Sherwin, Domain decomposition preconditioners for p and hp finite element approximation of Stokes equations, Computer Methods in Applied Mechanics and Engineering 175 (3) (1999) 243 – 266. doi:https://doi.org/10.1016/S0045-7825(98)00356-9.
URL http://www.sciencedirect.com/science/article/pii/S0045782598003569

[20] S. J. Sherwin, R. M. Kirby, J. Peiró, R. L. Taylor, O. C. Zienkiewicz, On 2d elliptic discontinuous Galerkin methods, International Journal for Numerical Methods in Engineering 65 (5) (2006) 752–784. doi:10.1002/nme.1466.
URL http://dx.doi.org/10.1002/nme.1466

[21] L. Haupt, J. Stiller, W. Nagel, A fast spectral element solver combining static condensation and multigrid techniques, Journal of Computational Physics 255 (2013) 384 – 395. doi:

https://doi.org/10.1016/j.jcp.2013.07.035.
URL http://www.sciencedirect.com/science/article/pii/S0021999113005251

[22] D. Pardo, J. Álvarez Aramberri, M. Paszynski, L. Dalcin, V. Calo, Impact of element-level static condensation on iterative solver performance, Computers & Mathematics with Applications 70 (10) (2015) 2331 – 2341. doi:https://doi.org/10.1016/j.camwa.2015.09.005.
URL http://www.sciencedirect.com/science/article/pii/S0898122115004198

[23] D. N. Arnold, F. Brezzi, B. Cockburn, L. D. Marini, Unified analysis of discontinuous galerkin methods for elliptic problems, SIAM journal on numerical analysis 39 (5) (2002) 1749–1779.

[24] B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems, SIAM Journal on Numerical Analysis 47 (2) (2009) 1319–1365. doi:10.1137/070706616.
URL http://dx.doi.org/10.1137/070706616

[25] R. M. Kirby, S. J. Sherwin, B. Cockburn, To CG or to HDG: A comparative study, Journal of Scientific Computing 51 (1) (2011) 183–212. doi:10.1007/s10915-011-9501-7.
URL http://dx.doi.org/10.1007/s10915-011-9501-7

[26] S. Yakovlev, D. Moxey, R. M. Kirby, S. J. Sherwin, To CG or to HDG: A comparative study in 3D, Journal of Scientific Computing 67 (1) (2016) 192–220. doi:10.1007/s10915-015-0076-6.
URL http://dx.doi.org/10.1007/s10915-015-0076-6

[27] C. Cantwell, D. Moxey, A. Comerford, A. Bolis, G. Rocco, G. Mengaldo, D. D. Grazia, S. Yakovlev, J.-E. Lombard, D. Ekelschot, B. Jordi, H. Xu, Y. Mohamied, C. Eskilsson, B. Nelson, P. Vos, C. Biotto, R. Kirby, S. Sherwin, Nektar++: An open-source spectral/element framework, Computer Physics Communications 192 (2015) 205 – 219. doi:https://doi.org/10.1016/j.cpc.2015.02.008.
URL https://www.sciencedirect.com/science/article/pii/S0010465515000533