Abstract In this paper, we address the study of elliptic boundary value problems in presence of a boundary condition of integral type (IBC) where the potential is an unknown constant and the flux (the integral of the flux density) over a portion of the boundary is given by a value or a coupling condition. We first motivate our work with realistic examples from nano-electronics, high field magnets and ophthalmology. We then define a general framework stemming from the Hybridizable Discontinuous Galerkin method that accounts naturally for the IBC and we provide a complete analysis at continuous and discrete levels. The implementation in the Feel++ framework is then detailed and the convergence and scalability properties are verified. Finally, numerical experiments performed on the real-life motivating applications are used to illustrate our methodology.

Keywords Integral boundary conditions · Hybridizable Discontinuous Galerkin · 3D-0D coupling · nMOS transistor · High field magnets · Ophthalmology

Mathematics Subject Classification (2020) 65N30 · 46N40 · 82D37 · 82D40 · 92B05

A HDG method for elliptic problems with integral boundary condition: Theory and Applications

Silvia Bertoluzza · Giovanna Guidoboni · Romain Hild · Daniele Prada · Christophe Prud'homme · Riccardo Sacco · Lorenzo Sala · Marcela Szopos

the date of receipt and acceptance should be inserted later

1 Introduction

In this paper, we study non standard boundary conditions for the following model elliptic boundary value problem of second order, written in mixed form:

$$\mathbf{j} + \mathcal{K}\nabla p = \mathbf{0} \qquad \text{in } \Omega, \tag{1a}$$

$$\nabla \cdot \mathbf{j} = f \qquad \text{in } \Omega. \tag{1b}$$

S. Bertoluzza
IMATI, CNR, Pavia, Italy E-mail: silvia.bertoluzza@imati.cnr.it

G. Guidoboni
Department of Electrical Engineering and Computer Science, Department of Mathematics, University of Missouri, Columbia, MO, 65211, USA E-mail: guidobonig@missouri.edu

R. Hild
Cemosis, IRMA UMR 7501, CNRS, Université de Strasbourg, 7 rue René Descartes, 67084 Strasbourg, France E-mail: hild@unistra.fr

D. Prada
IMATI, CNR, Pavia, Italy E-mail: daniele.prada@imati.cnr.it

C. Prud'homme
Cemosis, IRMA UMR 7501, CNRS, Université de Strasbourg, 7 rue René Descartes, 67084 Strasbourg, France E-mail: prudhomme@unistra.fr

R. Sacco
Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy E-mail: riccardo.sacco@polimi.it

L. Sala
Centre de recherche INRIA de Saclay - Ile de France, 91120 Palaiseau, France E-mail: lorenzo.sala@inria.fr

M. Szopos
Université Paris Cité, CNRS, MAP5, F-75006 Paris, France E-mail: marcela.szopos@u-paris.fr

Here, we denote by Ω the open bounded set of $\mathbb{R}^d$, $d = 2, 3$, representing the computational domain and $\mathcal{K}$ can be a scalar field or a symmetric tensor.

Many physical problems fit into this framework. The dependent variables p and $\mathbf{j}$ could represent, for example, the electric potential and electric current density in a circuit, or the temperature and heat flux in a solid. Another relevant application is the flow in a porous medium in stationary conditions. In this case, p and $\mathbf{j}$ denote the fluid pressure and the Darcy velocity, respectively. Equation (1a) represents then a constitutive law (also referred to as *Darcy's law*), while equation (1b) expresses the conservation of fluid mass under the presence of the source/sink term f. The above mentioned applications are characterized by a wide variety of boundary conditions, typically including Dirichlet, Neumann and Robin types. If such standard boundary conditions (Dirichlet, for example) are enforced on p, well-posedness of problem (1) in the space $H^1(\Omega)$ for p is a consequence of the elimination of $\mathbf{j}$ as a function of p through (1a) and the application of the Lax-Milgram Lemma to the resulting second-order elliptic boundary value problem for p. This provides also existence and uniqueness of $\mathbf{j}$ in the space $H(\mathrm{div}; \Omega)$ because of (1b). However, the simulation of complex systems in nanoscale electronics or ocular biomechanics gives rise to physical properties of some portions of the boundary that do not fall into such categories, thus making it necessary to introduce novel kinds of boundary conditions. In many cases, a boundary condition of integral type, henceforth called integral boundary condition (IBC), may be profitably adopted to mathematically represent the behavior of the unknowns on a subset of $\partial\Omega$, denoted by Γ_{ibc}, where it is not possible to directly enforce the value of p or its associated normal flux density $\mathbf{j} \cdot \mathbf{n}$, where $\mathbf{n}$ is the outward unit normal vector on the boundary of Ω. Specifically, the following boundary conditions can be considered: (i) the restriction $p|_{\Gamma_{ibc}}$ is equal to a constant value (to be determined); and (ii) the integral over Γ_{ibc} of the normal flux density $\mathbf{j} \cdot \mathbf{n}$ is equal to a given value. This latter situation may occur when it is not possible (or easy) to experimentally access the pointwise value of p on a portion of the boundary where the global flux of the vector field $\mathbf{j}$ across Γ_{ibc} is a given design target. More details about such realistic applications can be found in Section 2.

To address problems of the form (1), we carry out a complete analysis of the formulation at the continuous level in Section 3 and we propose a Hybridizable Discontinuous Galerkin method (HDG), which we adapt for the numerical treatment of the novel integral boundary conditions in Section 4. The HDG method has several attractive features: *i)* it provides optimal approximation of both primal and flux variables; *ii)* it requires less globally coupled degrees of freedom than DG methods of comparable accuracy; *iii)* it allows local element-by-element postprocessing to obtain new approximations with enhanced accuracy and conservation properties.

The HDG implementation has been carried out using Feel++ in Section 5. Feel++ is a C++ library for solving partial differential equations in $\mathbb{R}^d, d = 1, 2, 3$ and on manifolds using a wide range of Galerkin methods, from standard finite element methods, both continuous and discontinuous, to reduced basis

methods. Endowed with a domain specific language, whose grammar and vocabulary are tailored to the description of variational formulations and their discretization, Feel++ provides a scalable, versatile and expressive computational framework that applied mathematicians, engineers and scientists can use to access and test a wide range of methods and problems, both classical and newly developed.

This computational framework is then tested by various numerical experiments in presence of integral boundary conditions including convergence and performance tests in Section 5.3, and then realistic applications on large problems (up to 75 millions dofs) in Section 6. Finally, conclusions and outlook are discussed in Section 7.

2 Applications involving an integral boundary condition

Before introducing the model problem that we will use to carry out the analysis, and presenting the discretization method we will employ to tackle it, let us give some examples of realistic problems where integral boundary conditions naturally arise.

2.1 Floating-gate nMOS transistor in inversion conditions

The first example is the study of the distribution of the electric potential V in a nanoscale floating-gate nMOS (Metal-Oxide-Semiconductor) transistor working in inversion conditions. The prefix "n" indicates that the electric current is due to negatively charged electrons. A scheme of a realistic floating-gate nMOS transistor used as a nonvolatile memory device is shown in Figure 1. Under the assumption of electroneutrality, the distribution of V is obtained as the solution of a linear elliptic problem that can be written in mixed form similarly to (1) (see e.g [36, Section 6.1], [27, Sections 4.6-4.7] and [29, Section 9.1]) as:

$$\mathbf{D} + \varepsilon \nabla V = \mathbf{0} \qquad \text{in } \Omega, \tag{2a}$$

$$\nabla \cdot \mathbf{D} = 0 \qquad \text{in } \Omega, \tag{2b}$$

where $\mathbf{D}$ denotes the electric displacement, V is the electric potential and ε is the dielectric permittivity.
The computational domain Ω is the three-dimensional bounded set obtained by extruding the scheme of Figure 1 (right panel) by a width L_z along the axis z perpendicular to the plane of the figure. The domain Ω is made of the union of four subdomains, namely Ω_{ox} (silicon dioxide, here simply *oxide*), Ω_S (*source*), Ω_D (*drain*) and Ω_B (*bulk*). The union of the last three subdomains represents the silicon semiconductor region, namely $\Omega_{si} = \Omega_S \cup \Omega_D \cup \Omega_B$. The subdomains, Ω_{si} and Ω_{ox} are separated by an interface surface $\Gamma_{int} = \partial\Omega_{si} \cap \partial\Omega_{ox}$. Figure 1 also reports the notation for the various domain boundaries and their geometrical dimensions.

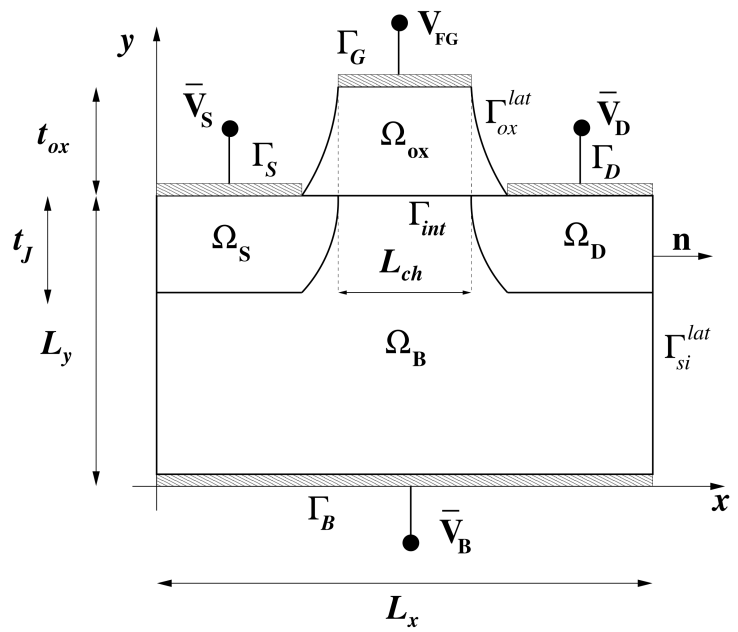


Fig. 1: Scheme of a nonvolatile memory. The device is composed of a pair of n-doped source and drain regions, a p-doped substrate region and a silicon dioxide (SiO_2) region on top of which a floating gate Γ_G is located. The regions denoted by Γ_S, Γ_D and Γ_B are the source, drain and bulk contacts, respectively.

Equations (2) are equipped with the following boundary conditions:

$$V = \overline{V}_j + \overline{V}_{bi,j} \qquad \text{on } \Gamma_j,\ j = S, D, B, \tag{3a}$$

$$\mathbf{D}\cdot\mathbf{n} = 0 \qquad \text{on } \Gamma_{\mathrm{si}}^{\mathrm{lat}} \cup \Gamma_{\mathrm{ox}}^{\mathrm{lat}}, \tag{3b}$$

$$\int_{\Gamma_G} \mathbf{D}\cdot\mathbf{n}\, d\Sigma = q\overline{N}_B \delta L_{\mathrm{ch}} L_z, \qquad V \text{ is constant on } \Gamma_G, \tag{3c}$$

and interface conditions:

$$[\![V]\!]_{\Gamma_{\mathrm{int}}} = 0, \qquad [\![\mathbf{D}]\!]_{\Gamma_{\mathrm{int}}} = q\overline{N}_B\delta \qquad \text{on } \Gamma_{\mathrm{int}}. \tag{4}$$

where $[\![\cdot]\!]_{\Gamma_{\mathrm{int}}}$ denotes the jump of a scalar or vector field across Γ_{int} . The quantity q is the electron charge, $\overline{N}_B$ is the concentration of ionized dopants in the bulk region and δ is the width of the accumulation region in the y direction of the channel, while the quantities $\overline{V}_{bi,j}$ are the built-in potentials associated with the subdomain regions Ω_j, $j = S, D, B$.

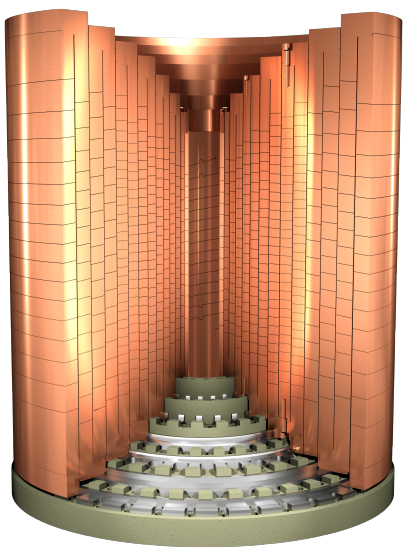


Fig. 2: A sliced view of a so called bitter magnet [14].

2.2 A nonlinear thermoelectric application

The second example is the study of the current density distribution $\mathbf{j}$, electric potential V and temperature T in a resistive magnet. We consider a real-life model devised within the HiFiMagnet project [14] which aims at developing an efficient multi-physics model for high field magnets. Let the domain Ω be a 14 helices magnet (a sliced view of the magnet is shown in Figure 2). The magnet is composed by several helices, each with a different kind of copper, linked by rings at the top and bottom of them. Between each helix, the magnet is cooled by passing water in the channels.

Under appropriate assumptions [13], the current density within Ω is described by the following elliptic problem:

$$\mathbf{j} + \sigma(T)\nabla V = \mathbf{0} \qquad \text{in } \Omega, \tag{5a}$$

$$\nabla \cdot \mathbf{j} = 0 \qquad \text{in } \Omega, \tag{5b}$$

where V is the electric potential and $\sigma = \sigma_0/(1 + \alpha(T - T_0))$ is the electric conductivity which depends on the temperature, with σ_0, α and T_0 given constant material parameters. The system is coupled with the following nonlinear elliptic problem describing heat transfer:

$$\mathbf{q} + k(T)\nabla T = \mathbf{0} \qquad \text{in } \Omega, \tag{6a}$$

$$\nabla \cdot \mathbf{q} = -\frac{\|\mathbf{j}\|^2}{\sigma(T)} \qquad \text{in } \Omega, \tag{6b}$$

where $\mathbf{q}$ is the heat flux and $k(T) = \sigma(T)LT$ is the thermal conductivity, L being the Lorentz number, and where $\|\cdot\|$ denotes the euclidean norm.

Here, the model corresponds to an experiment where an electric potential difference is applied between Γ_{in} and Γ_{out} and the aim is to reach a target current intensity I on Γ_{out}. The standard approach (both in experimental and numerical frameworks) would be to treat this as an inverse problem and test different values of the electric potential, until the target value for the current intensity on Γ_{out} is attained. Alternatively, we can equip system (5) with the boundary conditions:

$$V = 0 \qquad \text{on } \Gamma_{\text{in}}, \tag{7}$$

$$\mathbf{j} \cdot \mathbf{n} = 0 \qquad \text{on } \partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}}), \tag{8}$$

$$\int_{\Gamma_{\text{out}}} \mathbf{j} \cdot \mathbf{n} = I, \quad V \text{ is constant} \qquad \text{on } \Gamma_{\text{out}}, \tag{9}$$

and directly obtain the electric potential needed to achieve the target value I for the current intensity. In addition, problem (6) is equipped with the following boundary conditions:

$$\mathbf{q} \cdot \mathbf{n} = h_i(T - T_w) \qquad \text{on } \Gamma_{\text{channel}_i}, \tag{10}$$

$$\mathbf{q} \cdot \mathbf{n} = 0 \qquad \text{on } \partial\Omega \setminus (\cup_i \Gamma_{\text{channel}_i}), \tag{11}$$

modeling the fact that the magnet is thermally insulated on $\partial\Omega \setminus (\cup_i \Gamma_{\text{channel}_i})$ and cooled by water on $\Gamma_{\text{channel}_i}$, where the amount of exchanged heat depends on a heat transfer coefficient for each channel h_i.

2.3 Tissue perfusion with applications in ophthalmology

The last example is the perfusion of the lamina cribrosa, a collagen structure located in the optic nerve head that plays a crucial role in several diseases such as glaucoma [21,26]. In the spirit of [**?**], the lamina cribrosa is modeled as a porous material where blood vessels are viewed as isotropically distributed pores in a solid matrix comprising collagen, elastin, extracellular matrix and neural tissue, leading to the following time-dependent problem:

$$\mathbf{j} + k_p \, \nabla p = \mathbf{0} \qquad \text{in } \Omega \times (0, T), \tag{12a}$$

$$\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{j} = f \qquad \text{in } \Omega \times (0, T), \tag{12b}$$

where p is the Darcy pressure, $\mathbf{j}$ is the discharge velocity (or blood perfusion velocity), and k_p is the permeability. The domain $\Omega \subset \mathbb{R}^3$ represents the spatial domain occupied by the lamina cribrosa, which is schematized as the hollow cylinder depicted in Figure 3.

In order to account for systemic factors that influence the local perfusion, we couple the three-dimensional (3d) model (12) for the lamina perfusion with

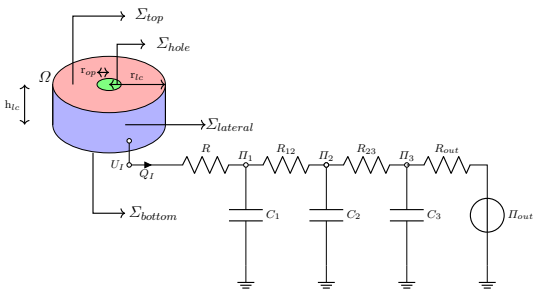


Fig. 3: Schematic representation of the multiscale coupled system describing the perfusion of the lamina cribrosa in the optic nerve head.

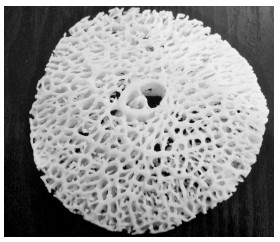


Fig. 4: Lamina cribrosa microstructure. Courtesy of `https://coggle.it`.

a simplified zero-dimensional (0d) model for the blood circulation in the posterior ciliary arteries nourishing the lamina, as depicted in Figure 3. Leveraging the electric analogy to fluid flow, the current through the circuit corresponds to volumetric blood flow rate and the electric potential corresponds to blood pressure [16]. Denoting by $\boldsymbol{\Pi} = [\Pi_1,\ \Pi_2,\ \Pi_3]^T$ the vector of unknown pressures at the circuit nodes, the dynamics of the 0d circuit is described by

$$\frac{d\boldsymbol{\Pi}}{dt} = \underline{\underline{A}}\boldsymbol{\Pi} + \mathbf{s} + \mathbf{b}, \tag{12c}$$

where $\underline{\underline{A}}$ is a matrix accounting for the effect of the vascular resistances and compliances, and

$$\mathbf{b} = [Q_I,\ 0,\ 0]^T, \qquad \mathbf{s} = \left[0,\ 0,\ \frac{\Pi_{out}}{R_{out}}\right]^T,$$

with $Q_I = \dfrac{U_I - \Pi_1}{R}$, where U_I is the unknown pressure on $\Sigma_{lateral}$ (assumed to be spatially uniform). We remark that Q_I and U_I are only functions of time, namely $Q_I = Q_I(t)$ and $U_I = U_I(t)$.

The coupling conditions at $\Sigma_{lateral}$ guarantee the continuity of mass and pressure and read:

$$\int_{\Sigma_{lateral}} \mathbf{j}\cdot\mathbf{n} = Q_I, \qquad p \text{ is a constant on } \Sigma_{lateral}, \tag{12d}$$

$$U_I = p \text{ on } \Sigma_{lateral}. \tag{12e}$$

System (12a) – (12d) is completed by the boundary conditions:

$$p = p_{hole} \qquad \text{on } \Sigma_{hole} \tag{12f}$$

$$\mathbf{j}\cdot\mathbf{n} = 0 \qquad \text{on } \Sigma_{top}\cup\Sigma_{bottom} \tag{12g}$$

with p_{hole} given, and the initial conditions:

$$p(\mathbf{x}, t = 0) = p_0(\mathbf{x}) \qquad \text{in } \Omega \tag{12h}$$
$$\boldsymbol{\Pi}(t = 0) = \boldsymbol{\Pi}_0\,. \tag{12i}$$

The blood flow through the 3d-0d coupled system is driven by the pressure difference between the imposed values Π_{out} and p_{hole}, representing the blood pressure at the level of the posterior ciliary arteries and the central retinal vein, respectively.

3 The model problem

The three realistic problems described in the previous section share some common features: they involve the mixed formulation of a second order elliptic equation in divergence form, complemented by several boundary conditions, among which an IBC. These characteristics are synthesized in the model problem presented in this section.

Let $\Omega \subset \mathbb{R}^d$ denote a polygonal (for $d = 2$) or polyhedral (for $d = 3$) domain, for which we do not assume convexity; the boundary of Ω is denoted by Γ, whereas $\mathbf{n}$ is the outward unit normal vector on Γ. $\mathcal{K} \in (L^\infty(\Omega))^{d\times d}$ is a symmetric $d \times d$ matrix, uniformly positive definite over Ω and $f \in L^2(\Omega)$. Let the boundary Γ of Ω be partitioned into three disjoint subsets: Γ_D, Γ_N, and Γ_{ibc}. We assume that $\bar{\Gamma}_D \cap \bar{\Gamma}_{ibc} = \emptyset$, implying that the Dirichlet and integral boundary portions are not adjacent.

Our goal is to solve the following mixed problem: find $\mathbf{j}$, p such that:

$$\mathbf{j} + \mathcal{K}\nabla p = \mathbf{0} \qquad \text{in } \Omega, \tag{13}$$
$$\nabla \cdot \mathbf{j} = f \qquad \text{in } \Omega, \tag{14}$$

subject to the non standard boundary conditions:

$$p = 0 \text{ on } \Gamma_D, \qquad \mathbf{j}\cdot\mathbf{n} = g_N \text{ on } \Gamma_N, \tag{15}$$
$$p \text{ is an (unknown) constant on } \Gamma_{ibc}, \qquad \int_{\Gamma_{ibc}} \mathbf{j}\cdot\mathbf{n} = I_{target}, \tag{16}$$

where $|\Gamma_{ibc}|$ denotes the measure of Γ_{ibc} and where I_{target} is a given constant. We use a compact notation for integrals: for functions $p, q \in L^2(D)$, we write $(p, q)_D = \int_D pq$ if $D \subset \mathbb{R}^d$, and $\langle p, q\rangle_D = \int_D pq$ if $D \subset \mathbb{R}^{d-1}$.

We next introduce a function $\bar{\varphi} \in H^{1/2}(\Gamma)$ verifying

$$\bar{\varphi}|_{\Gamma_D} = 0, \quad \bar{\varphi}|_{\Gamma_{ibc}} = 1.$$

Since $\bar{\Gamma}_D \cap \bar{\Gamma}_{ibc} = \emptyset$, such a function exists and can be chosen as regular as needed. We let $\Phi = \text{span} < \bar{\varphi} > \oplus H^{1/2}_{00}(\Gamma_N)$, where span $< \bar{\varphi} >$ denotes the linear space generated by the function $\bar{\varphi}$, and

$$H^{1/2}_{00}(\Gamma_N) = \{\varphi \in H^{1/2}(\Gamma) :\ \varphi = 0 \text{ on } \Gamma_D \cup \Gamma_{ibc}\}.$$

We then consider the following weak formulation of (13)-(16): find $\mathbf{j} \in H(\mathrm{div};\Omega)$, $p \in L^2(\Omega)$, $\widehat{p} \in \Phi$ such that for all $\mathbf{v} \in H(\mathrm{div};\Omega)$, $w \in L^2(\Omega)$, $\mu \in \Phi$:

$$(\mathcal{K}^{-1}\mathbf{j}, \mathbf{v})_\Omega - (p, \nabla\cdot\mathbf{v})_\Omega + \langle \widehat{p}, \mathbf{v}\cdot\mathbf{n}\rangle_\Gamma = 0, \tag{17a}$$

$$(\nabla\cdot\mathbf{j}, w)_\Omega = (f, w)_\Omega, \tag{17b}$$

$$\langle \mathbf{j}\cdot\mathbf{n}, \mu\rangle_\Gamma = \langle g_N, \mu\rangle_{\Gamma_N} + I_{target}|\Gamma_{ibc}|^{-1}\langle \mu, 1\rangle_{\Gamma_{ibc}}. \tag{17c}$$

In order to analyze Problem (17) we start by observing that, since the normal trace operator from $H(\mathrm{div};\Omega)$ to $H^{-1/2}(\Gamma)$ admits a bounded right inverse, the following inf-sup condition holds

$$\inf_{\substack{\psi\in\Phi\\ \psi\neq 0}} \sup_{\substack{\mathbf{j}\in H(\mathrm{div};\Omega)\\ \mathbf{j}\neq\mathbf{0}}} \frac{\langle \mathbf{j}\cdot\mathbf{n}, \psi\rangle_\Gamma}{\|\mathbf{j}\|_{H(\mathrm{div};\Omega)}\|\psi\|_{1/2,\Gamma}} \gtrsim 1, \tag{18}$$

where, from here on, we will use the notation $A \lesssim B$ (resp. $A \gtrsim B$) to indicate that the quantity $A \leq cB$ (resp. $A \geq cB$), c being a strictly positive constant, which, in the discrete framework, we will assume to be independent of the mesh size parameter.

Next, we prove the following Lemma.

Lemma 1 *Setting*

$$\mathcal{Z} = \{\mathbf{j} \in H(\mathrm{div};\Omega) : \ \langle \mathbf{j}\cdot\mathbf{n}, \psi\rangle_\Gamma = 0 \ \forall \psi \in \Phi\},$$

the following inf-sup condition holds

$$\inf_{\substack{p\in L^2(\Omega)\\ p\neq 0}} \sup_{\substack{\mathbf{j}\in\mathcal{Z}\\ \mathbf{j}\neq\mathbf{0}}} \frac{(p, \nabla\cdot\mathbf{j})_\Omega}{\|p\|_{0,\Omega}\|\mathbf{j}\|_{H(\mathrm{div};\Omega)}} \gtrsim 1. \tag{19}$$

Proof In order to prove the Lemma, for any given $p \in L^2(\Omega)$ we need to find $\mathbf{j} \in \mathcal{Z}$ such that

$$(p, \nabla\cdot\mathbf{j})_\Omega \gtrsim \|p\|_{0,\Omega}\|\mathbf{j}\|_{H(\mathrm{div};\Omega)}.$$

Let then $p \in L^2(\Omega)$ be given and let $\psi \in H^1(\Omega)$ be the zero mean solution of

$$-\nabla\cdot(\mathcal{K}\nabla\psi) = p \ \text{ in } \Omega, \quad \mathcal{K}\nabla\psi\cdot\mathbf{n} = \widehat{g} \ \text{ on } \Gamma,$$

where $\widehat{g} \in L^2(\Gamma)$ is defined as

$$\widehat{g} = \begin{cases} -|\Gamma|^{-1}\int_\Omega p & \text{on } \Gamma_D, \\ 0 & \text{on } \Gamma\setminus\Gamma_D, \end{cases}$$

so that the source term p and Neumann boundary data $\widehat{g}$ are compatible, which implies that ψ is well defined and $\mathbf{j} = -\mathcal{K}\nabla\psi$ belongs to $\mathcal{Z}$. By construction we have

$$(p, \nabla\cdot\mathbf{j})_\Omega = \int_\Omega |p|^2.$$

The inf-sup bound (19) follows by observing that

$$\|\mathbf{j}\|_{H(\mathrm{div};\Omega)} \lesssim \|\psi\|_{1,\Omega} \lesssim \|p\|_{(H^1(\Omega))'} + \|\widehat{g}\|_{-1/2,\Gamma} \lesssim \|p\|_{0,\Omega},$$

where $(H^1(\Omega))'$ is the dual of $H^1(\Omega)$.

In view of (18) and of (19), we can then prove the following Theorem

Theorem 1 *For $f \in L^2(\Omega)$, $g_N \in L^2(\Gamma_N)$, Problem (17) admits a unique solution $(\mathbf{j}, p, \widehat{p})$, with $\mathbf{j} = -\mathcal{K}p$ and $\widehat{p} = p|_\Gamma$, p satisfying $p \in H^1(\Omega)$ and*

$$-\nabla \cdot (\mathcal{K}\nabla p) = f \text{ in } \Omega, \quad p = 0 \text{ on } \Gamma_D, \quad -\mathcal{K}\nabla p \cdot \mathbf{n} = g_N \text{ on } \Gamma_N, \tag{20}$$

as well as the non standard boundary condition:

$$p \text{ is constant on } \Gamma_{ibc}. \tag{21}$$

Moreover, if g_N satisfies suitable compatibility conditions, the non standard boundary condition

$$\int_{\Gamma_{ibc}} -\mathcal{K}\nabla p \cdot \mathbf{n} = I_{target} \tag{22}$$

is also satisfied.

Proof Using the theory developed in [30], the two inf-sup conditions (18) and (19), imply existence, uniqueness and stability of the solution $(\mathbf{j}, p, \widehat{p})$. Testing equation (17b) with an arbitrary $w \in C_0^\infty(\Omega)$ and integrating by part we can see that $\nabla p = \mathcal{K}^{-1}\mathbf{j}$ in Ω. As, under our assumptions, $\mathcal{K}^{-1} \in L^\infty(\Omega)$, this implies $\nabla p \in L^2(\Omega)$ and, hence, $p \in H^1(\Omega)$. By standard arguments, we can then verify that (20) holds and that $p = \widehat{p}$ on Γ. Condition (21) is then encoded directly in the definition of the space Φ. To conclude, we only need to prove that (22) holds. In order to do so, we need to give a rigorous meaning of such an equality, by showing that p has enough regularity for the integral on the left hand side to be well defined. Indeed, writing

$$\int_{\Gamma_{ibc}} \mathbf{j} \cdot \mathbf{n} = \int_\Gamma (\mathbf{j} \cdot \mathbf{n})\bar{\bar{\varphi}} \qquad \text{with} \qquad \bar{\bar{\varphi}} = \begin{cases} 1 & \text{on } \Gamma_{ibc} \\ 0 & \text{on } \Gamma \setminus \Gamma_{ibc} \end{cases}, \tag{23}$$

as $\bar{\bar{\varphi}} \notin H^{1/2}(\Gamma)$, for a generic $\mathbf{j} \in H(\mathrm{div};\Omega)$, we would not be able to give a sense to such an integral. We start by observing that p also solves equation (20) with standard mixed Dirichlet-Neumann boundary condition obtained by complementing the boundary conditions in (20) with the additional boundary condition $p = \widehat{p}$ on Γ_{ibc}. It is known (see *e.g.* [18,19]) that, under minimal regularity assumptions on the domain and on the data, the solution of a standard mixed Dirichlet-Neumann problem verifies $p \in W^{1,q}(\Omega)$ for some $q > 2$, where $W^{1,q}(\Omega)$ denotes the Sobolev space of $L^q(\Omega)$ function with gradient in $L^q(\Omega)$. This, in turn, implies that $\mathbf{j} \cdot \mathbf{n} = -\mathcal{K}\nabla p \cdot \mathbf{n} \in W^{-1/q,q}(\Gamma) = (W^{1/q,q'})'$ with $q' < 2$ such that $1/q + 1/q' = 1$ ([18, Section 1.5]). Since $\bar{\bar{\varphi}} \in W^{1/2,q'}(\Gamma) \subset$

$W^{1/q,q'}(\Gamma)$, the integral on the right hand side of (22) can be defined by duality. Let now $\bar{\varphi}_0 = \bar{\varphi} - \bar{\bar{\varphi}}$. We have

$$\int_{\Gamma_{ibc}} \mathbf{j}\cdot\mathbf{n} = \int_{\Gamma} (\mathbf{j}\cdot\mathbf{n})\bar{\bar{\varphi}} = \int_{\Gamma} (\mathbf{j}\cdot\mathbf{n})\bar{\varphi} - \int_{\Gamma} (\mathbf{j}\cdot\mathbf{n})\bar{\varphi}_0$$
$$= \int_{\Gamma_N} g_N\bar{\varphi} + I_{target} - \int_{\Gamma} (\mathbf{j}\cdot\mathbf{n})\bar{\varphi}_0 = I_{target},$$

where the last equation descends from the observation that $\mathrm{supp}(\bar{\varphi}_0) = \Gamma_N$ and that $\bar{\varphi}_0 = \bar{\varphi}$ on its support.

4 The HDG method

Among the many possible approaches that can be used to discretize (13)-(16) we choose to resort to the Hybridizable Discontinuous Galerkin method (HDG) proposed in [6]. This method has some features that make it particularly well suited for the applications at hand: it provides a direct approximation of the normal component of the flux density vector field across the facets of the discretization, a quantity that is the one the end users are often interested in. Moreover it benefits from a superconvergence property that allows for a better accuracy when compared to other methods based on polynomials of the same order, and the particular form considered below turns out to outperform other methods with similar characteristics (see [8,9]).

We then consider a family of conforming, regular triangulations $\{\mathcal{T}_h\}_{h>0}$ of Ω into closed d-simplices K, *i.e.* straight triangles if $d = 2$ and straight tetrahedra if $d = 3$ (see [32, Chapter 3]). For each $K \in \mathcal{T}_h$, we denote by h_K the diameter of K and by $|K|$ its d-dimensional measure. We let $h = \max_{K\in\mathcal{T}_h} h_K$. We let ∂K denote the boundary of K and $\mathbf{n}_{\partial K}$ the associated outward unit normal vector on ∂K.

To simplify the exposition we adopt the convention to use a three dimensional terminology, and use the word *face* to refer both to the faces of a tetrahedron for $d = 3$ and to the edges of a triangle for $d = 2$. We let $\mathcal{F}_h$ denote the collection of all the faces of $\mathcal{T}_h$, whose union forms the *skeleton* of the decomposition $\mathcal{T}_h$. For all faces $F \in \mathcal{F}_h$ we let $|F|$ denote the $(d-1)$-dimensional measure of F. The set $\mathcal{F}_h$ naturally splits into the subset $\mathcal{F}_h^{\Gamma}$ of faces belonging to Γ and into the subset of faces belonging to the interior of Ω, denoted by $\mathcal{F}_h^0$. Finally, assuming that the decomposition $\mathcal{T}_h$ is such that for all faces F in $\mathcal{F}_h$ with $F \subset \Gamma$ either $F \subset \Gamma_D$, $F \subset \Gamma_N$ or $F \subset \Gamma_{ibc}$, the sets of boundary faces can be further split into the subsets $\mathcal{F}_h^{\Gamma_D}$, $\mathcal{F}_h^{\Gamma_N}$ and $\mathcal{F}_h^{\Gamma_{ibc}}$.

Let $\mathbf{q} \in (L^2(\Omega))^d$ with $\mathbf{q}^K = \mathbf{q}|_K \in H(\mathrm{div}; K)$ for all $K \in \mathcal{T}_h$, and $p \in L^2(\Omega)$ with $p^K = p|_K \in H^1(K)$ for all $K \in \mathcal{T}_h$. Let $F = \partial K_1 \cap \partial K_2$, $K_1, K_2 \in \mathcal{T}_h$, be a face belonging to $\mathcal{F}_h^0$. Then we can define the *jump* $[\![\mathbf{q}]\!]_F \in H^{-1/2}(F)$ of the normal trace of $\mathbf{q}$ as well as the *jump* $[\![p]\!]_F \in H^{1/2}(F)$ of the

scalar field p across F as

$$[\![\mathbf{q}]\!]_F = \mathbf{q}^{K_1} \cdot \mathbf{n}_{\partial K_1}|_F + \mathbf{q}^{K_2} \cdot \mathbf{n}_{\partial K_2}|_F. \tag{24}$$

$$[\![p]\!]_F = p^{K_1}\mathbf{n}_{\partial K_1}|_F + p^{K_2}\mathbf{n}_{\partial K_2}|_F. \tag{25}$$

It can be proved (see [32, Prop. 3.2.2]) that

$$[\![\mathbf{q}]\!]_F = 0 \quad \forall F \in \mathcal{F}_h^0 \qquad \Leftrightarrow \qquad \mathbf{q} \in H(\mathrm{div}; \Omega). \tag{26}$$

4.1 The HDG formulation

Let us introduce the finite element spaces:

$$\mathbb{V}_h = \prod_{K\in\mathcal{T}_h} \mathbb{V}(K), \quad W_h = \prod_{K\in\mathcal{T}_h} W(K), \quad M_h = M_h^* \oplus \widetilde{M}_h \tag{27}$$

with:

$$\widetilde{M}_h = \{\mu \in L^2(\mathcal{F}_h)\,|\,\mu|_F \in \mathcal{P}_k(F)\ \forall F \in \mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N}, \quad \mu|_{\Gamma_D\cup\Gamma_{ibc}} = 0\}, \tag{28}$$

$$M_h^* = \{\mu \in L^2(\mathcal{F}_h)\,|\,\mu|_{\Gamma_{ibc}} = \alpha,\ \alpha \in \mathbb{R}, \quad \mu|_{\mathcal{F}_h\setminus\Gamma_{ibc}} = 0\}, \tag{29}$$

where, as proposed in [6], the local spaces are

$$\mathbb{V}(K) = \mathbf{P}_k(K), \qquad W(K) = \mathcal{P}_k(K), \qquad M(F) = \mathcal{P}_k(F), \tag{30}$$

$\mathcal{P}_k(K)$ (resp. $\mathcal{P}_k(F)$) denoting the space of polynomials of degree less or equal to k on K (resp. F), $k \geq 0$, with $\mathbf{P}_k(K) = (\mathcal{P}_k(K))^d$. Definitions (27) and (30) imply that functions belonging to $\mathbb{V}_h$ and W_h are, in general, discontinuous across element edges in 2D and element faces in 3D of $\mathcal{T}_h$, whereas functions in M_h are discontinuous across element vertices in 2D and element edges in 3D of $\mathcal{F}_h \setminus \mathcal{F}_h^{\Gamma_{ibc}}$, *single-valued* on each face $F \in \mathcal{F}_h$ of the skeleton of $\mathcal{T}_h$ and constant on Γ_{ibc}.

The functions of M_h play the role of "connectors" between adjacent elements that, otherwise, would be irremediably uncoupled in the discretization process (see Section 5). For a mechanical interpretation of the interelement connecting role of the space M_h, see [5] and references cited therein.

Defining the *numerical normal flux* on ∂K as

$$\widehat{\mathbf{j}}_h^{\partial K} \cdot \mathbf{n}_{\partial K} = \mathbf{j}_h^K|_{\partial K} \cdot \mathbf{n}_{\partial K} + \tau_{\partial K}(p_h^K|_{\partial K} - \widehat{p}_h|_{\partial K}), \tag{31}$$

the discrete formulation reads: find $\mathbf{j}_h \in \mathbb{V}_h, p_h \in W_h$ and $\widehat{p}_h \in M_h$ such that $\forall \mathbf{v}_h \in \mathbb{V}_h$, $\forall w_h \in W_h$ and $\forall \mu_h \in M_h$:

$$\sum_{K\in\mathcal{T}_h} \left[(\mathcal{K}^{-1}\mathbf{j}_h^K, \mathbf{v}_h^K)_K - (p_h^K, \nabla\cdot\mathbf{v}_h^K)_K + \langle\widehat{p}_h, \mathbf{v}_h^K\cdot\mathbf{n}_{\partial K}\rangle_{\partial K}\right] = 0, \tag{32a}$$

$$\sum_{K\in\mathcal{T}_h} \left[-(\mathbf{j}_h^K, \nabla w_h^K)_K + \langle\widehat{\mathbf{j}}_h^{\partial K}\cdot\mathbf{n}_{\partial K}, w_h^K\rangle_{\partial K}\right] = \sum_{K\in\mathcal{T}_h} (f, w_h^K)_K, \tag{32b}$$

$$\sum_{K\in\mathcal{T}_h} \langle\widehat{\mathbf{j}}_h^{\partial K}\cdot\mathbf{n}_{\partial K}, \mu_h\rangle_{\partial K} = \langle g_N, \mu_h\rangle_{\Gamma_N} + I_{target}|\Gamma_{ibc}|^{-1}\langle\mu_h, 1\rangle_{\Gamma_{ibc}}. \tag{32c}$$

The dependent variables $\mathbf{j}_h$ and p_h are the approximations of $\mathbf{j}$ and p in the interior of each element $K \in \mathcal{T}_h$, whereas the dependent variable $\widehat{p}_h$ is the approximation of the trace of p on each face of $\mathcal{F}_h$. The numerical normal flux (31) is characteristic of a particular class of HDG methods, the so-called Local Discontinuous Galerkin Hybridizable (LDG-H) methods proposed and investigated in a series of seminal papers [8,10,11,9].

The quantity $\tau_{\partial K}$ is a nonnegative stabilization parameter and may assume different values on each face $F \in \partial K$ depending on the mesh element which F belongs to. We notice that if $\tau_{\partial K} \simeq \mathcal{O}(h^{-1})$, the second term on the right-hand side of (31) may be regarded as an artificial diffusion, thus justifying the name *stabilization parameter* attributed to $\tau_{\partial K}$.

Equations (32a) and (32b) are the HDG discrete counterpart of equations (17a) and (17b), respectively. These discrete equations hold in the interior of each $K \in \mathcal{T}_h$ and can be solved for each K to eliminate $\mathbf{j}_h^K$ and p_h^K in favor of the variable $\widehat{p}_h^{\partial K}$. This elimination procedure is called *static condensation* and was introduced for the first time in [38] in the context of dual mixed methods for equilibrium in stress analysis and, subsequently, in [2] in the context of the analysis and implementation of the dual mixed-hybridizable finite element method.

Combining this procedure with equation (31) for each $K \in \mathcal{T}_h$, it is possible to express the normal numerical flux as a function of the sole variable $\widehat{p}_h^{\partial K}$. At this point, we can use the remaining equations (32c). It is convenient to rewrite these equations in a different form by exploiting the property of the test function μ_h of being single-valued on each face of $\mathcal{F}_h$ and discontinuous across face edges. This leads to three sets of distinct equations. One group of equations enforces in a weak sense the interelement continuity of the normal component of $\widehat{\mathbf{j}}_h$ across each internal face as

$$\langle [\![\widehat{\mathbf{j}}_h]\!], \mu_h \rangle_F = 0, \qquad \forall F \in \mathcal{F}_h^0, \qquad \mu_h \in M_h. \tag{33}$$

Another group of equations enforces, again in a weak sense, the Neumann boundary condition (equation (15), right) as

$$\langle \widehat{\mathbf{j}}_h \cdot \mathbf{n}, \mu_h \rangle_F = \langle g_N, \mu_h \rangle_F, \qquad \forall F \in \mathcal{F}_h^{\Gamma_N}, \qquad \mu_h \in M_h. \tag{34}$$

We are left with a single scalar equation weakly enforcing the integral boundary condition (equation (16), right) as

$$\langle \widehat{\mathbf{j}}_h \cdot \mathbf{n}, \mu_h \rangle_{\Gamma_{ibc}} = I_{target} |\Gamma_{ibc}|^{-1} \langle \mu_h, 1 \rangle_{\Gamma_{ibc}}, \qquad \mu_h \in M_h. \tag{35}$$

In [2], it was proven that $\widehat{p}_h|_{\mathcal{F}_h^0}$ can also be interpreted as a *Lagrange multiplier* associated with the continuity condition (33). Similarly, we could prove that $\widehat{p}_h|_{\Gamma_{ibc}}$ is the multiplier associated with the integral boundary condition (35), thereby implying the condition on the left hand side of (16).

More details about the implementation will be provided in Section 5.

Remark 1 (Dirichlet boundary condition) The boundary condition (equation (15), left) on the dependent variable $\widehat{p}_h$ is enforced in an essential manner in the definition of the trial space M_h as in the standard CG method.

Theorem 2 (Existence and uniqueness of the discrete solution) *The discrete problem* (31)–(32) *has a unique solution.*

Proof Proposition 3.1 of [6] holds unchanged. Indeed by testing the discrete equations with $(\mathbf{v}_h, w_h, \mu_h) = (\mathbf{j}_h, p_h, -\widehat{p}_h)$ and integrating by parts, also in our case we obtain the discrete energy equation

$$(\mathcal{K}^{-1}\mathbf{j}_h, \mathbf{j}_h)_\Omega + \sum_{K\in\mathcal{T}_h} \tau_{\partial K}\langle p_h - \widehat{p}_h, p_h - \widehat{p}_h\rangle_{\partial K} = (f, p_h)_\Omega \\ -\langle g_N, \widehat{p}_h\rangle_{\Gamma_N} - I_{target}|\Gamma_{ibc}|^{-1}\langle \widehat{u}_h, 1\rangle_{\Gamma_{ibc}}.$$

Now, to prove uniqueness (which, in the finite dimensional setting, implies existence) we need to prove that vanishing data yield the null solution. Let then $f = 0$, $g_N = 0$ and $I_{target} = 0$. The discrete energy equation then imply $\mathbf{j}_h = 0$ and $p_h^K = \widehat{p}_h$ on ∂K for all K. Testing equation (32a) with $\mathbf{v}_h^K = \nabla p_h^K \in \mathbb{V}(K)$ we obtain that ∇p_h^K is a constant for all K. Then $\widehat{p}_h$ is a constant itself, which is necessarily 0, due to the Dirichlet boundary condition.

In order to provide an a priori bound on the discretization error, let $(\mathbf{j}, p)$ be the solution of Problem (13)–(14) with boundary conditions (15)–(16). Observe that, by Theorem 1, $p \in H^1(\Omega)$, and this allows us to define $\widehat{p}$ as the trace of p on the skeleton. In order to prove an error estimate we start with an observation: let $\mathrm{P}_{M_h} : L^2(\mathcal{F}_h) \to M_h$ denote the L^2 projection onto M_h. Thanks to the definition of M_h, it turns out that the value of $\mathrm{P}_{M_h}(\widehat{p})|_{\Gamma_{ibc}}$ is the average of $\widehat{p}$ on Γ_{ibc}. Since $\widehat{p}$is a constant on Γ_{ibc}, we have

$$\mathrm{P}_{M_h}(\widehat{p}) = \widehat{p} = p \quad \text{ on } \Gamma_{ibc}. \tag{36}$$

Moreover, letting $\mathrm{P}_{\widehat{M}_h} : L^2(\mathcal{F}_h) \to \widehat{M}_h$ denote the L^2 orthogonal projection onto the auxiliary space

$$\widehat{M}_h = \{q \in L^2(\mathcal{F}_h) : q|_F \in \mathcal{P}_k(F)\ \forall F \in \mathcal{F}_h\} \supseteq M_h,$$

we have that $\mathrm{P}_{\widehat{M}_h} p = \mathrm{P}_{M_h} p$. In view of this observation, Theorem 3.4 of [6] holds with unchanged proof also in our case, that is, we have the following theorem.

Theorem 3 *Letting*

$$e_{\mathbf{j}} = \mathbf{\Pi}\mathbf{j} - \mathbf{j}_h, \qquad e_p = \Pi p - p_h, \qquad e_{\widehat{p}} = P_{\widehat{M}_h} p - \widehat{p}_h,$$

we have that

$$\|e_{\mathbf{j}}\|_{0,\Omega}^2 + \sum_{K\in\mathcal{T}_h} \tau_{\partial K}|e_p^K - e_{\widehat{p}}|_{0,\partial K}^2 \lesssim |\mathbf{\Pi}\mathbf{j} - \mathbf{j}|_{0,\Omega}^2.$$

Morover, if the elliptic regularity estimate $\|u\|_{2,\Omega} \lesssim \|\nabla\cdot(\mathcal{K}\nabla u)\|_{0,\Omega}$ *holds for all* $u \in H_0^1(\Omega)$ *with* $\nabla\cdot(\mathcal{K}\nabla u) \in L^2(\Omega)$*, then we have that*

$$\|e_p\|_{0,\Omega} \lesssim h^{\min\{k,1\}}\|\mathbf{\Pi}\mathbf{j} - \mathbf{j}\|_{0,\Omega}.$$

We now recall that, if $\mathbf{j}|_K \in (H^{k+1}(\Omega))^d$ with $\nabla \cdot \mathbf{j} \in H^{k+1}(\Omega)$, and $p \in H^{k+1}(\Omega)$ then we have ([6, Theorem 3.3])

$$|\mathbf{\Pi}^K \mathbf{j} - \mathbf{j}|_{0,K} \lesssim h_K^{k+1} |\mathbf{j}|_{k+1,K} + h_K^{k+1} \tau_{\partial K} |p|_{k+1,K}, \tag{37}$$

$$|\Pi^K p - p|_{0,K} \lesssim h_K^{k+1} \tau_{\partial K} |p|_{k+1,K} + h_K^{k+1} \tau_{\partial K}^{-1} |\nabla \cdot \mathbf{j}|_{k+1,K}, \tag{38}$$

which yields

$$|e_p|_{0,\Omega} \lesssim h^{k+1} \quad \text{and} \quad |e_{\mathbf{j}}|_{0,\Omega} \lesssim h^{k+1+\min\{k,1\}},$$

and then, by triangular inequality,

$$\|p - p_h\|_{0,\Omega} \lesssim h^{k+1} \quad \text{and} \quad \|\mathbf{j} - \mathbf{j}_h\|_{0,\Omega} \lesssim h^{k+1}.$$

Remark 2 We would like to point out that resorting to the proposed HDG method is by no means the only viable option to solve (13)-(16), and other methods exist, such as the hybridized version of the order k Brezzi-Douglas-Marini method, that, by using polynomials of different order for the different spaces, potentially yield the same order of convergence, possibly with less degrees of freedom. However, it has been observed that the HDG method here considered allows to achieve an overall better accuracy than several of such methods with minimal computational overhead (see [8,9]), which justifies our choice.

4.2 Static condensation

As it is customarily done for the HDG method, using a static condensation approach, the system (31)–(32c) can be recast in terms of a global linear system for the sole unknown $\widehat{p}_h$ (the numerical trace of the solution on the boundaries of the mesh elements). More precisely, equations (32a) and (32b) can be locally solved, yielding, in each element, $\mathbf{j}_h^K$ and p_h^K as a function of $\widehat{p}_h$. We can then substitute $\mathbf{j}_h^K = j_h^K(\widehat{p}_h)$ and $p_h^K = p_h^K(\widehat{p}_h)$ into (32c) to obtain an equation for $\widehat{p}_h$.

In order to account for the integral boundary condition in the static condensation procedure, we leverage the splitting $M_h = \widetilde{M}_h \oplus M_h^*$, which allows us to split $\widehat{p}_h$ and μ_h as $\widehat{p}_h = \widehat{p}_{1,h} + \lambda\bar{\bar{\varphi}}$ and $\mu_h = \mu_{1,h} + \xi\bar{\bar{\varphi}}$ with $\widehat{p}_{1,h}, \mu_{1,h} \in \widetilde{M}_h$ and $\lambda, \xi \in \mathbb{R}$, where $\bar{\bar{\varphi}}$ is defined in (23). The resulting linear system is naturally split as

$$\underbrace{\begin{pmatrix} & & \mathcal{A}_{14} \\ & \mathcal{A}_0 & -\mathcal{A}_{24} \\ & & 0 \\ \mathcal{A}_{14}^T & \mathcal{A}_{24}^T \; 0 & -\mathcal{A}_{44} \end{pmatrix}}_{\mathcal{A}} \underbrace{\begin{pmatrix} \mathbf{J} \\ P \\ \widehat{P}_1 \\ \lambda \end{pmatrix}}_{\mathcal{U}} = \underbrace{\begin{pmatrix} \mathbf{0} \\ A_f \\ 0 \\ 0 \end{pmatrix}}_{\mathcal{B}}. \tag{39}$$

We observe that $\mathcal{A}_0$ is the matrix corresponding to the monolithic discretization of the HDG discretization with homogeneous boundary conditions on $\Gamma_D \cup \Gamma_{ibc}$. The matrix $\mathcal{A}$ is obtained by complementing $\mathcal{A}_0$ with a single line and column, accounting for the contribution of the space $M_h^* \sim \mathbb{R}$ to the test and trial functions. The vectors $\mathcal{A}_{14}$ and $\mathcal{A}_{24}$, as well as the scalar $\mathcal{A}_{44}$ are obtained by assembling local contributions corresponding to the following bilinear operators:

$$\mathcal{A}_{14}^K \leftrightarrow \langle \lambda\bar{\bar{\varphi}}, \mathbf{v}_h^K \cdot \mathbf{n}_{\partial K} \rangle_{\partial K} \tag{40}$$

$$\mathcal{A}_{24}^K \leftrightarrow \langle \tau_{\partial K}\lambda\bar{\bar{\varphi}}, w_h^K \rangle_{\partial K}, \tag{41}$$

$$\mathcal{A}_{44}^K \leftrightarrow \langle \tau_{\partial K}\lambda\bar{\bar{\varphi}}, \xi\bar{\bar{\varphi}} \rangle_{\partial K}. \tag{42}$$

Remark 3 If more than one independent integral boundary conditions are imposed on different connected components of Γ_{ibc}, we will have to complement $\mathcal{A}_0$ with one row and one column for each component.

Eliminating $\mathbf{J}$ and P by static condensation results in a system of the form

$$\underbrace{\begin{pmatrix} \mathcal{S} & B^T \\ B & C \end{pmatrix}}_{\mathbb{D}} \underbrace{\begin{pmatrix} \widehat{P}_1 \\ \lambda \end{pmatrix}}_{} = \mathbf{F} + \mathbf{G}_N + \mathbf{G}_{\Gamma_{ibc}}, \tag{43}$$

where $\mathbf{G}_N$ is the vector containing the elements of $\langle g_N, \mu_{1,h} \rangle_{\Gamma_N}$ in the degrees of freedom corresponding to Neumann faces and zeros everywhere else, whereas $\mathbf{G}_{\Gamma_{ibc}}$ is a vector that has only zeros except for one single entry containing $I_{target}|\Gamma_{ibc}|^{-1}\langle \xi\bar{\bar{\varphi}}, 1 \rangle_{\Gamma_{ibc}}$ in the degree of freedom corresponding to faces on $\mathcal{F}_h^{\Gamma_{ibc}}$. The vector $\mathbf{F}$ and the matrix $\mathcal{S}$ are, respectively, the right hand side and matrix obtained by static condensation of the HDG system with homogeneous Dirichlet boundary conditions on $\Gamma_D \cup \Gamma_{ibc}$ and homogeneous Neumann boundary conditions on Γ_N (details on how such matrix and vector are constructed can be found in [7]). Observe that the Dirichlet boundary condition $p = 0$ on Γ_D is enforced onto $\widehat{p}_h = \widehat{p}_{1,h} + \lambda\bar{\bar{\varphi}}$ in an essential manner in the definition of the spaces $\widetilde{M}_h$ and M_h^*. The resulting global system has smaller size than other DG methods of comparable accuracy [10], and this results in renewed interest in HDG, also because, after solving the global system, the unknowns can be recovered locally, in parallel.

Remark 4 The linear system (39) could be solved in a monolithic way without resorting to static condensation. However, such an approach would be impractical and inefficient, as, letting N_{ibc} denote the number of independent integral boundary conditions, the size of the system is

$$\begin{aligned} \dim V_h + \dim W_h + \dim M_h = {} & \operatorname{card}\{K\} \times (n_V + n_W) \\ & + \operatorname{card}\{\mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N}\} \times \dim n_{\widetilde{M}} + N_{ibc} \times \underbrace{\dim n_{M_h^*}}_{=1}, \end{aligned}$$

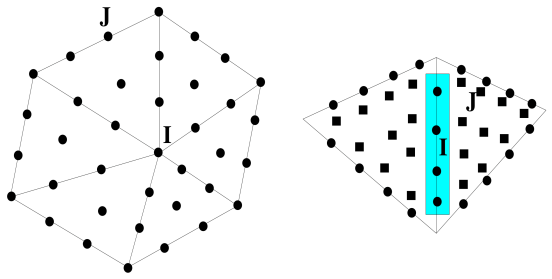


Fig. 5: Comparison between the connectivity of the continuous Galerkin (CG) and HDG methods in the case $d = 2$ without integral boundary conditions. Cubic polynomials are used for both methods. Black circles identify the nodal degrees of freedom of the CG method whereas black circles and black squares identify the degrees of freedom of the HDG method on the boundary and in the interior of each element, respectively. Left panel (CG method): all the elements sharing node I contribute to the connections between node I and the remaining nodes in $\mathcal{T}_h$. The number of connections (I, J) is equal to 37. Right panel (HDG method): the two neighboring elements are interconnected by the boundary degrees of freedom highlighted in the light blue rectangle. The number of connections (I, J) in the global system is equal to 24.

which is indeed much larger than the standard Continuous Galerkin and Hybridizable Discontinuous Galerkin counterparts.

4.3 Connectivity and partitioning

In absence of integral boundary conditions, the HDG method has degrees of freedom inside each element and on each face of $\mathcal{F}_h$. Therefore, unlike the case of the standard continuous Galerkin (CG) method, *no* degree of freedom is associated with the mesh vertices. This property makes the HDG method extremely efficient as far as coding is concerned because of its *binary* connectivity structure, i.e., two elements K_1 and K_2 belonging to $\mathcal{T}_h$ share their degrees of freedom (including those in their interiors) *only* if they share a face. This is not the case with the CG method for which nodal continuity at vertices connects the degrees of freedom of a number of elements that is not possible to predict a priori and which depends strongly on the regularity of the mesh. This issue is thoroughly discussed in [25] where a detailed comparison between the computational performance of the CG and HDG methods is carried out in the case $d = 3$. An example of the different connectivity structure associated with the CG and HDG approaches is shown in Figure 5 in the case of the space W_h with polynomial degree $k = 3$. Piecewise cubic continuous elements are used also for the CG method. We see that, for the same polynomial degree, the HDG scheme has a far lower connectivity than that of the CG method.

The presence of an integral boundary condition slightly alters the connectivity of the HDG (and CG) method. All the faces $F \in \mathcal{F}_h^{\Gamma_{ibc}}$ share a single common degree of freedom $\lambda_{2,h}$. This fact couples all the degrees of freedom on ∂K for all the elements $K \in \mathcal{T}_h$ having, at least, one face $F \in \partial K \cap \mathcal{F}_h^{\Gamma_{ibc}}$. Therefore, denoting by n_M the dimension of $P_k(F), F \in \mathcal{F}_h$, the row (column) of the global matrix $\mathbb{H}$ corresponding to $\lambda_{2,h}$ will have as many $1 \times n_M$ block rows ($n_M \times 1$ block columns) as the number of faces $F \notin \mathcal{F}_h^{\Gamma_{ibc}}$ belonging to the elements coupled by the integral boundary condition, that is, the number of faces in the following sets:

$$\mathcal{E}_{\text{ibc}} = \left\{ K \in \mathcal{T}_h, \partial K \cap \mathcal{F}_h^{\Gamma_{ibc}} \neq \emptyset \right\} \tag{44}$$

$$\mathcal{F}_{\text{ibc}} = \left\{ F \in \mathcal{F}_h \,|\, F \in \partial K \setminus \mathcal{F}_h^{\Gamma_{ibc}}, \text{ with } \partial K \cap \mathcal{F}_h^{\Gamma_{ibc}} \neq \emptyset, \ \forall K \in \mathcal{T}_h \right\}. \tag{45}$$

This property has an important consequence for parallelism: the integral boundary condition induces parallel communications between each element in $\mathcal{E}_{\text{ibc}}$. To avoid considerable performance loss, the elements in $\mathcal{E}_{\text{ibc}}$ should all belong to the same partition. This can be achieved by stating that all elements $\mathcal{E}_{\text{ibc}}$ are neighbors through the faces in $\{F; F \in \partial K \cap \mathcal{F}_h^{\Gamma_{ibc}}\}$ when setting the graph to be partitioned. Numerical experiments reported in Section 5 illustrate this property.

The process of static condensation outlined above can be easily extended to the case of $\mathcal{N}$ different integral boundary conditions.

In such a situation, the set $\mathcal{F}_h$ of faces of $\mathcal{T}_h$ is split as follows

$$\mathcal{F}_h = \mathcal{F}_h^0 \cup \mathcal{F}_h^{\Gamma_N} \cup \mathcal{F}_h^{\Gamma_D} \cup \mathcal{F}_h^{\Gamma_{ibc}^1} \cup \cdots \cup \mathcal{F}_h^{\Gamma_{ibc}^{\mathcal{N}}},$$

and the finite element space for the numerical trace $\widehat{p}_h$ becomes

$$M_h = M_h^{*,1} \oplus \cdots \oplus M_h^{*,\mathcal{N}} \widetilde{\oplus} M_h,$$

where $\widetilde{M}_h$ is defined as before, and, for $i = 1, \ldots, \mathcal{N}$,

$$M_h^{*,i} = \{\mu \in L^2(\mathcal{F}_h) \,|\, \mu|_F \in \mathbb{R} \ \forall F \in \mathcal{F}_h^{\Gamma_{ibc}^i}, \quad \mu|_F = 0 \ \forall F \in \mathcal{F}_h \setminus \mathcal{F}_h^{\Gamma_{ibc}^i}\}.$$

Remark 5 (Performance) A comparison between CG and HDG methods is carried out in [25]. We reached similar conclusions for standard boundary conditions. To compare the HDG and CG formulations in the presence of integral boundary conditions, we need to develop some strategies to handle the conditions in CG. Possible approaches include: (i) solving an inverse problem using a standard CG problem with Dirichlet condition on the potential; (ii) solving a CG problem using Lagrange multipliers to enforce the IBC; (iii) solving a mixed CG problem using Lagrange multipliers. The former method is clearly more expensive than the HDG formulation in the present paper. A full comparison is beyond the scope of this paper and will be treated in a future publication. However in Section 6.2 we show a simple comparison with a variant of (i) to solve a real life problem.

5 Computational framework

We discuss in the following the computational framework implementing the mathematical concepts described in the previous sections. We start by a general introduction to the Open Source library Feel++ and then move to the actual implementation showing the one-to-one relationship with the mathematical framework. Next, we discuss some specific aspects of HDG, namely the static condensation process and the algebraic representation. Even though parallelism is hidden in the framework, we provide here some details since HDG offers interesting opportunities implemented in Feel++ . Finally we illustrate partitioning, performances and convergence tests in presence of integral boundary conditions.

5.1 Feel++ general description

The computational framework described hereafter is based on the Finite Element Embedded Library in C++ (Feel++). Feel++ allows using a very wide range of Galerkin methods, as well as other numerical methods such as domain decomposition methods including mortar and three fields methods, fictitious domain methods or certified reduced basis. The ingredients of the software include a very expressive embedded language, seamless interpolation, mesh adaption and seamless parallelization. Feel++ has been used in various contexts including the development and numerical verification of (new) mathematical methods or the development of large multi-physics applications. The range of users span from mechanical engineers in industry, physicists in complex fluids, computer scientists in biomedical applications to applied mathematicians thanks to the shared common mathematical embedded language hiding linear algebra and computer science technical issues.

Feel++ provides a mathematical kernel for solving partial differential equations using arbitrary order Galerkin methods (FEM, SEM, CG, DG, CRB) in 1D, 2D, 3D and manifolds using simplices and hypercubes meshes [31]. It is based on: *(i)* a polynomial library allowing for a wide range polynomial expansions including H_{div} and H_{curl} elements; *(ii)* a light interface to Boost.UBlas, Eigen3 and PETSc/SLEPc as well as a scalable in-house solution strategy; *(iii)* a language for Galerkin methods starting with fundamental concepts such as function spaces, linear and bilinear forms, operators, functionals and integrals; *(iv)* a framework that allows user codes to scale seamlessly from single core computation to thousands of cores and enables hybrid computing.

Feel++ takes advantage of the newest C++ standard (C++17) such as type inference and the Boost C++ Libraries such as the Boost.Parameter, Boost.Fusion, Boost.Hana or Boost.MPL and many more. These language enhancements and libraries favor the development of very concise, robust and expressive C++ codes.

5.2 HDG Implementation in Feel++

Feel++ provides the mathematical language ingredients to implement HDG methods in C++ thanks to its Galerkin Domain Specific Embedded Language (DSEL).

We start with the function spaces and meshes. First, we need to construct $\mathcal{F}_h^{ibc}$ and $\mathcal{F}_h \setminus \mathcal{F}_h^{ibc}$.

```
// T_h(Ω)
auto mesh=loadMesh( _mesh=new Mesh<Simplex<d>>);
// filter to retrieve the complement of F_h^ibc in F_h
auto complement_integral_bdy = complement(faces(mesh),
  [&mesh]( auto const& e ) {
    if ( e.hasMarker() &&
         e.marker().matches(mesh->markerName("Ibc*") ) )
      return true;
    return false;
   });
// F_h \ F_h^ibc
auto face_mesh = createSubmesh( mesh,
                                complement_integral_bdy);
// F_h^ibc
auto ibc_mesh = createSubmesh( mesh,
                               markedfaces(mesh,"Ibc*"));
```

Then, we need to construct the spaces $V_h, W_h, \widetilde{M}_h$ and M_h^*. We also would like to handle an arbitrary number of integral boundary conditions, the number of which will only be known at execution. We have

$$X_h = V_h \times W_h \times \widetilde{M}_h \times (M_h^*)^n$$

The specification of the function spaces is obtained as follows:

```
Vh_ptr_t Vh = Pdhv<OrderP>( _mesh=mesh );
Wh_ptr_t Wh = Pdh<OrderP>( mesh );
Mh_ptr_t Mh = Pdh<OrderP>( face_mesh );
// only one degree of freedom
Ch_ptr_t Ch = Pch<0>(ibc_mesh);
// n IBC
auto ibcSpaces = product( nb_ibc, Ch);
auto Xh = product( Vh, Wh, Mh, ibcSpaces  );
```

Next we define the linear and bilinear forms on X_h and start assembling terms as follows:

```
auto a = blockform2( Xh )
auto rhs = blockform1( Xh );

. . .
// Assembling the right hand side
rhs(1_c) += integrate(_range=elements(mesh),_expr=-f*id(w));
. . .
// Assembling the main matrix
a(0_c,0_c) += integrate(_range=elements(mesh),
                        _expr=(trans(lambda*idt(u))*id(v)) );
. . .
//⟨p̂_h|_{M̃_h}, v_h^K · n_∂K⟩
a(0_c,2_c) += integrate(_range=internalfaces(mesh),
```

```
        _expr=( idt(phat)*(leftface(trans(id(v))*N())+
               rightface(trans(id(v))*N())));
```

where `0_c`, `1_c`, `2_c` ...refer to the index at compile time of the test or trial function spaces in `Xh` (X_h). Hence `a(0_c,0_c)` refers to the terms associated with the flux.

We can choose at runtime whether to solve the problem using static condensation or the monolithic approach. Although the latter approach becomes rapidly inefficient in $3D$ as the number of elements in the mesh increases, it is useful to provide its construction to debug and verify the static condensation process. Finally, preconditioners such as algebraic multigrid or alternatives from PETSc are used to solve the system. A matrix-free method might be also employed, however the preconditioning approach would then have to be redesigned.

In presence of several IBCs the framework enables a dynamic access to each of them using a relative index to the corresponding block as shown in the following code:

```
// A_30 dealing with the i-th IBC
a( 3_c, 0_c, i, 0 ) +=
   integrate( _range=markedfaces(mesh,"Ibc"),
              _expr=(trans(idt(u))*N()) * id(nu) );
```

The last step consists in solving the problem using either the monolithic or static condensation strategy and retrieve the components (flux and potential) of the solution field:

```
auto U = Xh.element();
// static condensation is done during the solve
a.solve(_solution=U, _rhs=rhs, _name="hdg");
// get views over each component
auto up = U(0_c); // element of V_h
auto pp = U(1_c); // element of W_h
auto phat = U(2_c); // element of M_h
auto ip = U(3_c,0); // element of M_h^*
```

In order to reconstruct the flux and potential once the trace has been obtained, the local solvers take advantage of the embarrassingly parallel context using multi-threading, including within the MPI process when we run in parallel. Similar considerations apply whenever the solution is postprocessed to get a potential of enhanced accuracy and/or a flux with better conservation properties, i.e. $\in H(\mathrm{div};\Omega)$.

5.3 Convergence tests

We now turn to the verification step by checking the convergence of the HDG algorithm for the mixed Poisson problem (46) in 2D and 3D domains. Here

we assume that $\mathcal{K}$ is the identity matrix:

$$\begin{aligned} \mathbf{j} + \nabla p &= 0 && \text{in } \Omega, \\ \nabla \cdot \mathbf{j} &= f && \text{in } \Omega, \\ p &= g_D && \text{on } \Gamma_D, \\ \mathbf{j} \cdot \mathbf{n} &= g_N && \text{on } \Gamma_N, \\ \int_{\Gamma_{ibc}} \mathbf{j} \cdot \mathbf{n} &= I_{target},\ p \text{ constant} && \text{on } \Gamma_{ibc}, \end{aligned} \tag{46}$$

where the 2D and 3D domains are built by constructive solid geometry: in 2D

$$\begin{aligned} \Omega \equiv \Omega^{2\mathrm{d}} &= \mathcal{D}(\mathbf{0}, 1, (0, 90)) \backslash \mathcal{D}(\mathbf{0}, 0.5, (0, 90)) \\ \Gamma_D &= \{(x, y) \in \Gamma \,|\, y = 0\},\ \Gamma_{ibc} = \{(x, y) \in \Gamma \,|\, x = 0\},\ \Gamma_N = \Gamma \setminus (\Gamma_D \cup \Gamma_{ibc}), \end{aligned}$$

where $\mathcal{D}(\mathbf{x}, R, (0, 90))$ is the disk arc between 0^o and 90^o of radius R centered in $\mathbf{x}$ and, similarly, in 3D

$$\begin{aligned} \Omega \equiv \Omega^{3\mathrm{d}} &= \{(x, y, z) \in \mathbb{R}^3 | (x, y) \in \Omega^{2\mathrm{d}}, z \in [0, 1]\} \\ \Gamma_D &= \{(x, y, z) \in \Gamma \,|\, y = 0\},\ \Gamma_{ibc} = \{(x, y, z) \in \Gamma \,|\, x = 0\},\ \Gamma_N = \Gamma \setminus (\Gamma_D \cup \Gamma_{ibc}). \end{aligned}$$

Exact solutions in 2D are:

$$\mathbf{j} = \frac{1}{2\pi} \begin{pmatrix} \dfrac{-y}{x^2 + y^2} \\ \dfrac{x}{x^2 + y^2} \end{pmatrix}, \qquad p = \frac{1}{2\pi} \mathrm{atan2}(y, x),$$

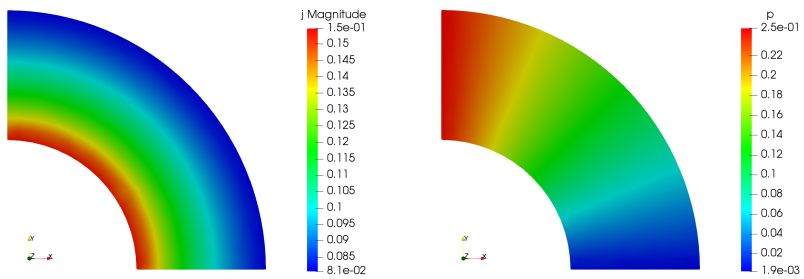
and in 3D are:

$$\begin{aligned} \mathbf{j} &= \frac{\mathrm{atan2}(y, x) \cos(xyz)}{2\pi} \begin{pmatrix} yz \\ xz \\ xy \end{pmatrix} + \frac{1 + \sin(xyz)}{2\pi(x^2 + y^2)} \begin{pmatrix} -y \\ x \\ 0 \end{pmatrix}, \\ p &= \frac{1}{2\pi} \mathrm{atan2}(y, x)(1 + \sin(xyz)). \end{aligned}$$

The domains and exact solutions in $2D$ are presented in Figure 6. In $3D$, the domain is obtained by extruding the $2D$ domain.

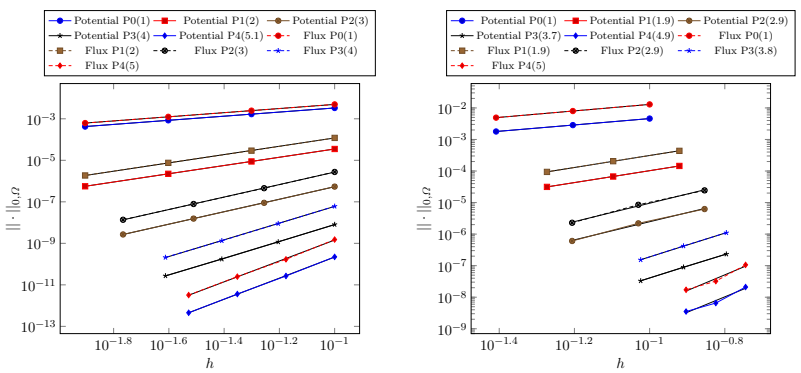
Thanks to the assumption that Γ_D and Γ_{ibc} are not adjacent, for the continuous problem the non homogeneous Dirichlet boundary conditions can be reduced, by a standard approach, to homogeneous boundary conditions, without impacting the integral boundary condition. In practice, at the the discrete level, the non homogeneous Dirichlet boundary conditions are treated as they usually are when dealing with the standard mixed Dirichlet-Neumann boundary conditions case.

The results in $2D$ and $3D$ are shown in Figures 7(a) and 7(b), respectively.



(a) 2D exact flux. (b) 2D exact potential.

Fig. 6: Computational domain and exact solution in 2D.



(a) 2D (b) 3D

Fig. 7: Convergence with IBC conditions.

The expected convergence rates (see Theorem 3) are obtained for polynomials up to degree 4.

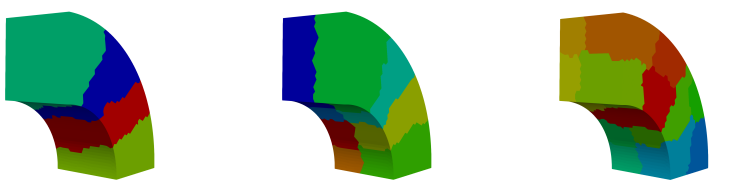
5.3.1 Partitioning and parallel performance

Following the discussion in Section 4.3, we now turn to a numerical experiment to illustrate specific treatment of the partitioning. Feel++ uses MPI as a parallel framework and partitioning is handled through the library Metis [24]. In the presence of integral boundary conditions, we need to adapt the element wise graph data structure to force the partition algorithm to aggregate the elements in $\mathcal{E}_{\text{ibc}}$ on the same partition to avoid communication. Without this change, the parallel performance can seriously deteriorate.

We conducted a performance study on a 30×10^6 dofs problem, solving (46) in the 3D domain, to investigate the effect of the aggregated mesh partitioning

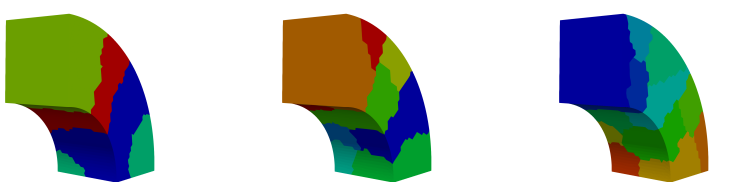
	SOLVER COMPUTATIONAL GAIN			
Partitioning	$n_p = 2$	$n_p = 4$	$n_p = 8$	$n_p = 16$
Standard	1 (1132.7)	2.74(412.666)	0.06(17147.3)	0.05(21778.5)
Adapted	1.01(1145.71)	2.66(424.482)	3.19(354.697)	36.22(31.27
Gain	1.01	0.97	48.34	696.42

Table 1: The first and second rows display the speedup relative to $n_p = 2$ (row-wise) with the absolute timing in parenthesis. The last row corresponds to the gain with respect to the same partitioning (column-wise)



(a) 4 subdomains (b) 8 subdomains (c) 16 subdomains

Fig. 8: Standard partitioning. The colors identify the process identifier for $n_p = 4, 8, 16$.



(a) 4 subdomains (b) 8 subdomains (c) 16 subdomains

Fig. 9: Aggregated partitioning. The colors identify the process identifier for $n_p = 4, 8, 16$.

proposed in the case of integral conditions on $n_p = 2, 4, 8, 16$ processors. Table 1 presents the results with and without aggregation. Figures 8 and 9 illustrate the partitioning without and with aggregation respectively. In the latter case, one can see that on the IBC face the elements belong to the same processor.

With 2 and 4 processors the results are very similar since the boundary with the integral condition hasn't been partitioned. With 8 processors, the solver in the aggregation partitioning test case is almost **50 times faster** than in the case without aggregation. With 16 processors, the solver computational time with aggregation partitioning is nearly **700 times faster** than without the aggregation (see Table 1).

6 Numerical simulation of realistic problems and a real application

In this section, we aim at demonstrating the potential of the theoretical and computational framework described above. By tackling the problems presented in Section 2, we show that the newly introduced integral boundary condition, together with its HDG discretization, can successfully address problems of different nature arising in science, engineering and medicine. More precisely, in the simulation of a floating-gate nanoscale electronic device (see Section 6.1) the combined adoption of the IBC, the interface condition and the HDG discretization method allows us to obtain an accurate prediction of the threshold voltage of the device; in the simulation of a high field magnet (see Section 6.2) which deals with a nonlinear model coupling electric potential and temperature with material properties depending on the temperature, the adoption of the IBC allows us to directly retrieve the value of the potential difference that must be applied to obtain a target current without the need to resort to a feedback loop or to the solution of an inverse problem; the time-dependent multiscale PDE-ODE problem arising in biomedical application from the coupling of a 3d tissue perfusion with the systemic circulation (see Section 6.3), is accurately, efficiently and naturally handled, at each time step, by the HDG approach for solving elliptic problems with the adoption of the IBC.

While they share a similar conceptual structure, including integral boundary conditions, which is the main object of study of the present paper, the three problems have, each, different mathematical and computational challenges in adition to those embodied in the basic linear elliptic problem discussed in Section 3. The results reported in this section show that the computational framework described in detail for linear elliptic problems is quite flexible and can be easily adapted to different situations, including also large-scale computing problems.

6.1 Floating-gate nMOS transistor in inversion conditions

We consider here the simulation of a floating-gate nMOS transistor in inversion conditions, modeled by equation (2a) and (2b) with boundary conditions given by (3a), (3b) and (3c) and interface condition (4). This can be cast as particular instance of the model problem (13)–(14) for $\mathbf{j} = \mathbf{D}$, $p = V$ and $\mathcal{K} = \varepsilon\mathcal{I}$ ($\mathcal{I}$ denoting the identity matrix).

The dielectric permittivity ε in equations (2a) and (48) below can be written as $\varepsilon = \varepsilon_0\varepsilon_r$, where ε_0 is the dielectric permittivity of vacuum and ε_r is the relative dielectric permittivity of silicon and oxide, equal to $\varepsilon_r^{\mathrm{si}}$ in Ω_{si} and $\varepsilon_r^{\mathrm{ox}}$ in Ω_{ox}, respectively. The quantity δ can be estimated using a simplified one-dimensional theory from solid state physics (see [29, Section 9.1]) as

$$\delta = \sqrt{\frac{4\varepsilon_0\varepsilon_r^{\mathrm{si}}V_{\mathrm{th}}\ln(\overline{N}_B/n_i)}{q\overline{N}_B}}, \tag{47}$$

where n_i is the silicon intrinsic concentration and $V_{\rm th}$ is the thermal voltage (see [36, Chapter 2]). The built-in potentials associated with the subdomain regions Ω_j, $j = S, D, B$ are given by:

$$\overline{V}_{bi,S} = +V_{\rm th} \ln\left(\frac{\overline{N}_S}{n_i}\right), \quad \overline{V}_{bi,D} = +V_{\rm th} \ln\left(\frac{\overline{N}_D}{n_i}\right), \quad \overline{V}_{bi,B} = -V_{\rm th} \ln\left(\frac{\overline{N}_B}{n_i}\right),$$

$\overline{N}_S$ and $\overline{N}_D$ being the concentrations of the ionized dopant impurities in the source and drain regions, respectively.

Tables 2 and 3 report the values of physical and geometrical parameters used in the simulations.

Parameter	**Symbol**	**Value**	**Units**
Electron charge	q	$1.602 \cdot 10^{-19}$	C
Permittivity of vacuum	ε_0	$8.854 \cdot 10^{-12}$	F m^{-1}
Relative permittivity of silicon	$\varepsilon_r^{\rm si}$	11.7	[–]
Relative permittivity of silicon dioxide	$\varepsilon_r^{\rm ox}$	3.9	[–]
Thermal voltage (at $T = 300$K)	$V_{\rm th}$	0.02589	V
Intrinsic concentration (at $T = 300$K)	n_i	$1.45 \cdot 10^{16}$	m^{-3}

Table 2: Values of physical parameters. Data from [29].

Parameter	**Symbol**	**Value**	**Units**
Horizontal length	L_x	$480 \cdot 10^{-9}$	m
Vertical length	L_y	$320 \cdot 10^{-9}$	m
Width	L_z	$320 \cdot 10^{-9}$	m
Oxide thickness	t_{ox}	$10 \cdot 10^{-9}$	m
Source and drain lengths	$\lvert\Gamma_S\rvert$, $\lvert\Gamma_D\rvert$	$160 \cdot 10^{-9}$	m
Channel length	L_{ch}	$40 \cdot 10^{-9}$	m
Junction depth	t_J	$106 \cdot 10^{-9}$	m
Source potential	$\overline{V}_S$	0	V
Bulk potential	$\overline{V}_B$	0	V
Drain potential	$\overline{V}_D$	0	V
Source doping concentration	$\overline{N}_S$	10^{26}	m^{-3}
Bulk doping concentration	$\overline{N}_B$	$5 \cdot 10^{22}$	m^{-3}
Drain doping concentration	$\overline{N}_D$	10^{26}	m^{-3}

Table 3: Values of input parameters for device simulation.

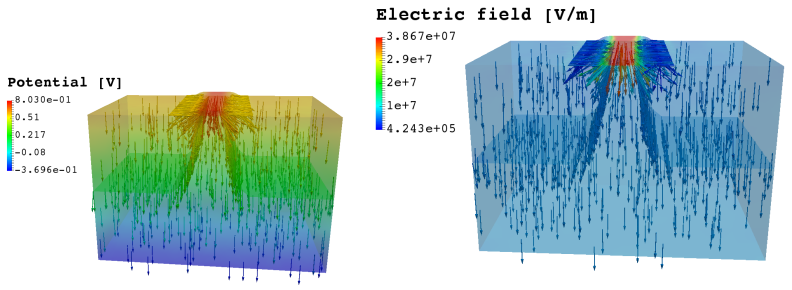
The main goal of the simulations is to determine, via the integral boundary condition (3c), the value attained on Γ_G at inversion conditions by the electric potential. This value is the threshold voltage of the device, denoted henceforth by V_T, and is a fundamental design parameter in integrated circuit nanoelectronics (see [29] and [1]). The electric field $\mathbf{E} = -\nabla V$ can be easily determined on each $K \in \mathcal{T}_h$ by inverting the algebraic equation

$$\mathbf{D} = \varepsilon \mathbf{E} \qquad \text{in } \Omega, \tag{48}$$

which, according to the theory of Maxwell equations [23], relates the electric field to the electric displacement $\mathbf{D}$, unlike standard displacement-based finite element formulations, for which the electric field needs to be post-processed through numerical differentiation of the electric potential V. Figure 10(a) illustrates the 3D distribution of the computed electric potential superposed with the vector plot of the electric field. We notice that the electric potential is an increasing function of the vertical coordinate y, negative at the bulk contact and positive at the source and drain contacts. The predicted value of V_T is positive and given by $V_T^{\text{computed}} = 0.8337965$ V. The sign of V_T^{computed} is in physical agreement with the fact that the transistor is of n type, so that electron charges are attracted from the bulk region up towards the gate contact. To quantitatively assess the accuracy of the estimate of the threshold voltage we adopt the analytical theory for an ideal MOS system developed in [29, Chapter 8] and use formula (8.3.18) of [29] to obtain $V_T^{\text{ideal}} = 0.8591$, which agrees with the value predicted by the numerical simulation within 3%. This result demonstrates that the combined adoption of the interface condition (4) and of the integral boundary condition (3c), in conjunction with the use of the HDG discretization method, are able to provide a very accurate prediction of the threshold voltage of the device.

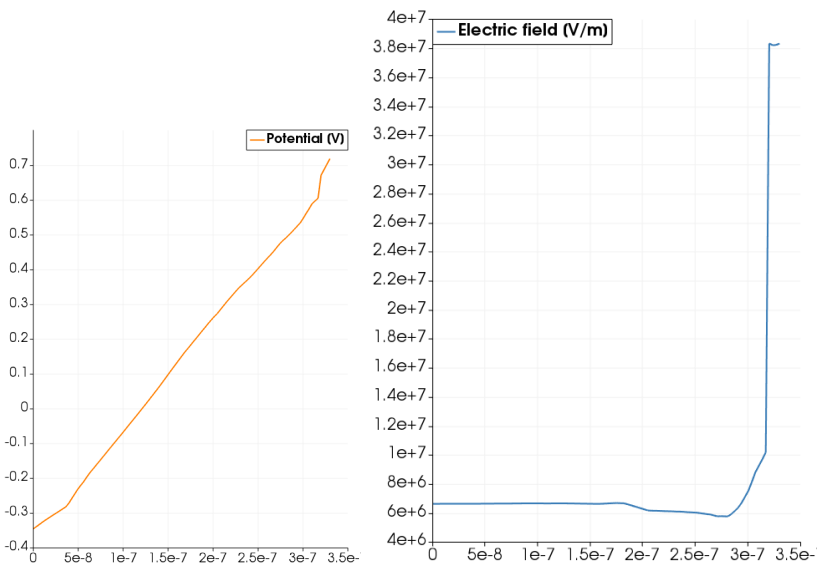
It is remarkable to notice that this latter quantity is always directed from the gate of the device towards the bulk contact of the device in such a way to attract electron charge from the bulk region towards the interface Γ_{int} to form the conductive channel. We can also notice the separation lines between the source and bulk regions and the drain and bulk regions, which correspond to a change of the sign of the ionized dopant charge density (positive in Ω_S and Ω_D, negative in Ω_B). Figure 10(b) shows a vector plot of the computed electric field with a color map of its strength throughout the device. Results clearly indicate that the electric field is mainly directed parallel to the y axis except in the region comprising the oxide layer, the interface and the two source and drain contacts, where the electric field arrows strongly deviate from the y direction and point from the floating gate towards the two contacts, in accordance with the fact that the electric potential is much higher on Γ_G than on Γ_S and Γ_D.

Figure 11(a) shows a 1D cut of the computed electric potential at $x = L_x/2$ along the y axis. We notice that the spatial distribution of V is approximately linear in both semiconductor and oxide regions, with a change in the slope across the interface coordinate $y = L_y = 320$nm. This is physically consistent with the fact that (a) the relative dielectric constant ε_r has a jump between silicon and oxide and (b) there is no space charge density inside each region of the device (the right-hand side of the Poisson equation (2b)). To further support these considerations about the electric potential V, we show in Figure 11(b) the 1D cut of the computed electric field strength at $x = L_x/2$ along the y axis. We notice that the field is discontinuous across the interface, in agreement with (a), and that it is approximately constant in both semiconductor and oxide regions, in agreement with (b).



(a) (b)

Fig. 10: (a) 3D view of the potential with electric field direction. (b) 3D view of the electric field with direction.



(a) (b)

Fig. 11: (a) 1D cut of the electric potential at $x = L_x/2$. (b) 1D cut of the electric field at $x = L_x/2$. In both panels the abscissa is the y direction. In panel (a) the y-axis is the potential whereas in panel (b) the y-axis is the magnitude of the electric field.

6.2 A nonlinear thermoelectric application

We now consider the simulation of the current density, potential and temperature in one of the resistive magnet from the national laboratory of high magnetic fields (LNCMI, France) , a modeled by the coupled equations (5) and (6). Both problems can be cast as a particular instance of problem (13)-(14), where $\mathbf{j}$ (resp. p and $\mathcal{K}$) in (13)-(14) corresponds to $\mathbf{j}$ (resp. V and $\sigma(T)\mathcal{I}$) in (5) and $\mathbf{q}$ (resp. T and the nonlinear function $k(T)\mathcal{I}$) in (6). In physical experiments, a feedback loop is performed on the potential difference to reach I, this approach corresponds to imposing Dirichlet conditions on the electric potential. Numerically, if Dirichlet conditions are imposed on the electric potential on Γ_{ibc}, many solves are required to reach the targeted experimental setup. This computationally expensive strategy is used with the CG method. Conversely, in our case, thanks to the IBC condition, only one solve is sufficient. By solving (5)- (5b) with the integral boundary conditions (7)- (9), we in fact impose the target current intensity value I directly, and we get the required electric potential V on Γ_{out} as an output.

The heat transfer coefficients can be determined from the thermal conductivity $k(T)$, the hydraulic diameter D_h and the Nusselt number Nu by $h = k(T)Nu/D_h$. The Nusselt number is characteristic of the flow, and in the case of high field magnet cooling, with perpendicular flows in annular region between cylinders, different correlation can be used, for example those of Colburn [12], Montgomery [28], Dittus [15] or Silberberg [37]. In this case, we use a constant approximation for the heat transfer coefficients h_i. On other surfaces, the thermal exchanges are not considered. The values of the target intensity imposed is 22148A.

Remark 6 We consider a simple update of the matrix blocks impacted by the non-linearity, whereas the static condensation needs to be executed at each nonlinear step.

The mesh has 3.3 million elements, the discontinuous vectorial space for the flux has 39 million dofs, the discontinuous scalar space for the potential has 13 million dofs and the discontinuous scalar space for the trace has 22 million dofs. On 48 processors (Intel Xeon E5-2680 v3 2.50GHz), using a Picard iteration to solve this nonlinear problem, see Algorithm 1, took around 1600s of computational time. Here, we want to emphasize the need of a proper partition of the mesh, see Section 5.3.1, in order to be able to reach the solution in a bounded time.

Figure 12 displays the computed electric potential (left panel) and the temperature (right panel), along red dots representing the heat and voltage sensors used during the experiment. Figure 13 shows the error for the different sensors between the simulation and the experiment. We can see that for the Dittus correlation, the error is less than 4%, and even under 2% for most of the sensors.

Finally, we compare the conservation of the current density obtained with the present framework with the results of the CG method. We solve the same

Algorithm 1: Picard algorithm for thermoelectric problem

Choose ε, N, V_0, T_0;
$i \leftarrow 1$, $\delta \leftarrow \varepsilon + 1$;
while $i \leq N$ **and** $\delta > \varepsilon$ **do**
  $\sigma \leftarrow \sigma(T_{i-1})$, $k \leftarrow k(T_{i-1})$;
  $(V_i, \mathbf{j}_i) \leftarrow \text{SolveV}(\sigma)$, $(T_i, \mathbf{q}_i) \leftarrow \text{SolveT}(k, \mathbf{j}_i)$;
  $\delta \leftarrow \max(\frac{||V_i - V_{i-1}||_{L^2}}{||V_{i-1}||_{L^2}}, \frac{||T_i - T_{i-1}||_{L^2}}{||T_{i-1}||_{L^2}})$;
  $i \leftarrow i + 1$;
end

	Mesh elements	Potential dofs	Flux dofs	Trace dofs	Picard
Single CG	22×10^6	4.3×10^6	-	-	500s
HDG	3.3×10^6	13×10^6	39×10^6	22×10^6	1600s

Table 4: Comparison between CG and HDG, for CG we need to solve several times to achieve the correct intensity.

Method	Number of dofs	Input $\mathbf{j}$	Output $\mathbf{j}$	Difference
CG Newton	4.3×10^6	30982.94	-31013.84	30.9
CG Picard	1×10^6	30991.47	-31021.05	29.58
HDG Picard	74×10^6	30898.33	-30898.35	0.02

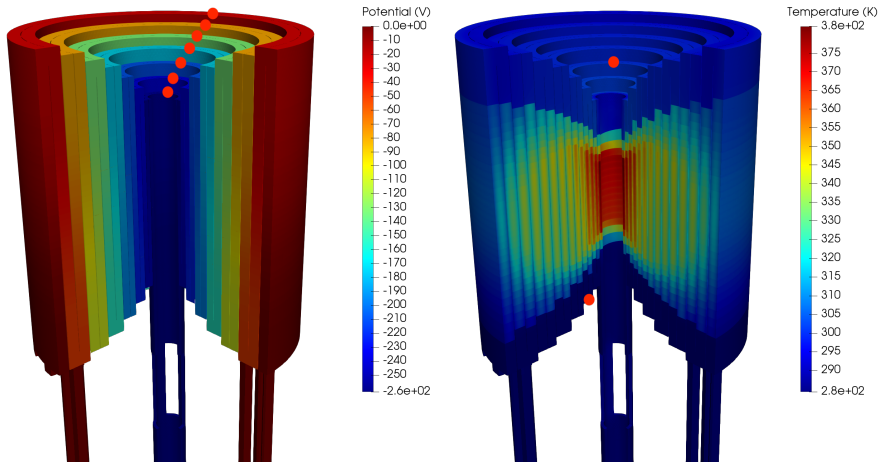
Table 5: Difference on the current between inlet and outlet

problem using the CG method and the Newton method for the nonlinearity on a finer mesh (22M elements) and the CG method and the Picard method for the nonlinearity on the same mesh (3.3M elements). The comparison between the computational time of CG and HDG is summarized in Table 4. We recall that for CG several iterations are required to achieve the target intensity, so the computational cost becomes quickly similar, or even significantly larger, when compared to a single HDG problem solving. Table 5 shows the difference between the input current and the output current compared for CG and HDG method.

In conclusion, results are very similar to those in [13] obtained with the CG formulation. However the HDG formulation is current conservative since the IBC conforms with the experimental setup where we impose, on Γ_{out}, the current I rather than the current density $\mathbf{j} \cdot \mathbf{n}$.

6.3 Tissue perfusion in ophthalmology

For the last test we illustrate the tissue perfusion problem introduced in Section 2.3. The assumption of isotropically distributed pores implies that $\mathcal{K} = k_p \mathcal{I}$ with k_p positive constant. In addition, for the specific circuit schema-



(a) Electric potential (b) Temperature

Fig. 12: Actual magnet operating at $I = 22148$ A

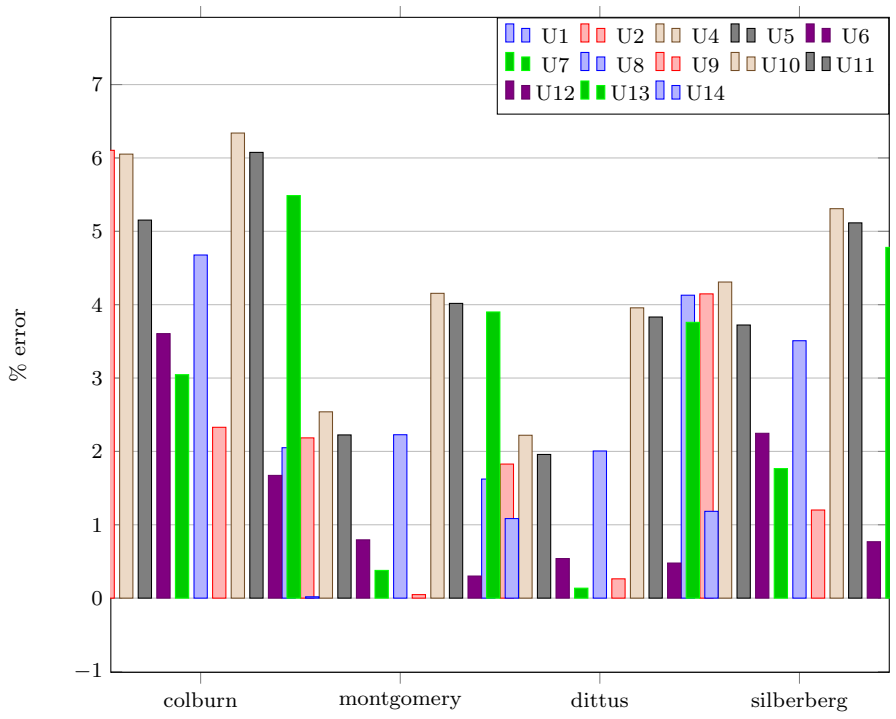


Fig. 13: Errors on the electric potential for HDG

tized in Figure 3, the matrix $\underline{\underline{A}}$ takes the form

$$\underline{\underline{A}} = \begin{bmatrix} -\dfrac{1}{C_1\, R_{12}} & \dfrac{1}{C_1\, R_{12}} & 0 \\ \dfrac{1}{C_2\, R_{12}} & -\dfrac{1}{C_2}\left(\dfrac{1}{R_{12}} + \dfrac{1}{R_{23}}\right) & \dfrac{1}{C_2\, R_{23}} \\ 0 & \dfrac{1}{C_3 R_{23}} & -\dfrac{1}{C_3}\left(\dfrac{1}{R_{23}} + \dfrac{1}{R_{out}}\right) \end{bmatrix},$$

where the vascular resistances R, R_{12}, R_{23} and R_{out}, and the compliances C_1, C_2, C_3 are assumed to be given positive constants, see Table 6. In order to solve the 3d-0d coupled system, we start by performing a semi-discretization in time of problem (12) using operator splitting, in the same spirit as [3], thus allowing to maintain the flexibility to apply the HDG method even in this time-dependent multiscale coupled problem. We refer to [17] for a review of operator splitting methods. Let us introduce a uniform time discretization so that $t^n = n\Delta t$, for $n \geq 0$, and let us utilize the notation $\varphi^n := \varphi(t^n)$ for a generic quantity φ. Then, given p^n and $\boldsymbol{\Pi}^n$, for $n \geq 0$ we solve the following two steps sequentially to advance from t^n to t^{n+1}:

Step 1: find $\mathbf{j}$, p and $\boldsymbol{\Pi}$ such that:

$$\begin{aligned} \mathbf{j} + k_p \nabla p &= \mathbf{0} && \text{in } \Omega \times \left(t^n, t^{n+1}\right), \\ \frac{\partial p}{\partial t} + \nabla \cdot \mathbf{j} &= f && \text{in } \Omega \times \left(t^n, t^{n+1}\right), \\ \frac{d\boldsymbol{\Pi}}{dt} &= \mathbf{b} && \text{in } \left(t^n, t^{n+1}\right), \end{aligned}$$

equipped with the boundary conditions (12f) –(12g), the interface conditions (12d)– (12e) and the initial conditions $p(t^n) = p^n$, $\boldsymbol{\Pi}(t^n) = \boldsymbol{\Pi}^n$ and then set $p^{n+\frac{1}{2}} = p(t^{n+1})$, $\boldsymbol{\Pi}^{n+\frac{1}{2}} = \boldsymbol{\Pi}(t^{n+1})$ and $\mathbf{j}^{n+\frac{1}{2}} = \mathbf{j}(t^{n+1})$.

Step 2: find p and $\boldsymbol{\Pi}$ such that

$$\begin{aligned} \frac{\partial p}{\partial t} &= 0 && \text{in } \Omega \times \left(t^n, t^{n+1}\right), \\ \frac{d\boldsymbol{\Pi}}{dt} &= \underline{\underline{A}}\boldsymbol{\Pi} + \mathbf{s} && \text{in } \left(t^n, t^{n+1}\right), \end{aligned}$$

equipped with the initial conditions $p(t^n) = p^{n+\frac{1}{2}}$ and $\boldsymbol{\Pi}(t^n) = \boldsymbol{\Pi}^{n+\frac{1}{2}}$, and then set $p^{n+1} = p(t^{n+1}) \left(= p^{n+\frac{1}{2}}\right)$, $\boldsymbol{\Pi}^{n+1} = \boldsymbol{\Pi}(t^{n+1})$ and $\mathbf{j}^{n+1} = \mathbf{j}(t^{n+1}) \left(= \mathbf{j}^{n+\frac{1}{2}}\right)$.

We remark that the solution of Step 1, upon discretization in time, reduces to the solution of an elliptic problem with integral interface conditions similar to Section 3. The operator splitting scheme described above is characterized by a first-order convergence in time, as discussed in [3,17] and verified in Figure 14. In addition, this method could also be improved to achieve second-order convergence in time via symmetrization, as described in [4,17]. For our application we have used the data summarized in Table 6.

Quantity	Value	Unit
r_{lc}	0.095	cm
r_{op}	0.008	cm
h_{lc}	0.03	cm
$k_p = k_{p,1}$	0.015192	kg cm^{-1} s^{-1}
$RR = R_{12} = R_{23} = R_{out}$	10^3	N s cm^{-5}
$C = C_1 = C_2 = C_3$	10^{-3}	cm^5 N^{-1}
p_{crv}	0.25	N cm^{-2}
Π_{out}	$\left(\frac{9}{10} - \frac{17\pi^2}{10000}\right)\sin\left(\frac{\pi t}{10}\right) + + \left(\frac{13\pi}{250} - \frac{3\pi^3}{10000}\right)\cos\left(\frac{\pi t}{10}\right) + \frac{47}{10}$	N cm^{-2}

Table 6: Data of the test case

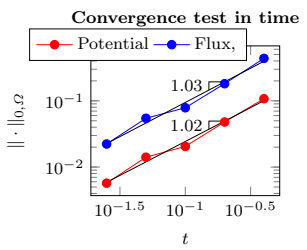
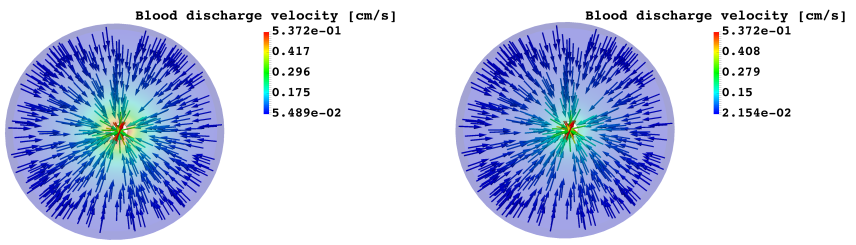


Fig. 14: Convergence test for the coupled model $3d - 0d$

Numerical simulations are conducted to compare the distributions of pressure and blood perfusion velocity in the lamina cribrosa (see Figure 15) and pressure and flow rate in the upstream circulation (see Figure 16) corresponding to healthy and disease conditions. Specifically, we study the effect of decreased permeability in the lamina cribrosa, which is associated with capillary dropout in glaucoma, see [20]. To simulate healthy and disease conditions, we assume $k_p = k_{p,1} = 0.015192$ and $k_p = k_{p,2} = k_{p,1}/10 = 0.0015192$, respectively. Overall, the simulations suggest that the pressure in the lamina decreases radially from the periphery to the center, consistently leading to a distribution of blood perfusion velocity directed towards the center. This result is consistent with the vascular structure in the lamina, which is supplied by the posterior ciliary arteries at its periphery and drains into the central retinal vein at its center. Interestingly, a decrease in permeability is predicted to lead to an increase in pressure, maximum value from 1.093 to 3.476 N cm^{-2}, and a slightly increase in the perfusion velocity, especially close to the opening, (see Figure 15) within the lamina, accompanied by a substantial increase in pressure in the upstream circulation modeled by the circuit (see Figure 16). A local pressure increase may lead to further capillary damage, thereby potentially accelerating the disease process. These results suggest that a decrease in permeability has a similar effect as closing a faucet, which leads to an increase in pressure upstream of the faucet itself. Since this faucet is distributed across the lamina, the consequent effect is also distributed across the tissue. This pilot investigation demonstrates the importance of coupling local tissue perfusion with the rest of the circulation to determine hemodynamic conditions in an organ, thereby motivating the quest for efficient and accurate computational methods for the solution of multiscale coupled problems such as those considered in this article.

7 Conclusions and outlook

In this paper, we propose a complete analysis of a HDG method for elliptic problems with integral boundary conditions from theory to applications. Theoretical results are verified by an experimental convergence analysis and



(a) $k_p = k_{p,1}$ (b) $k_p = k_{p,2}$

Fig. 15: Simulated discharge velocity inside the lamina cribrosa, driven by the pressure drop source in the circuit at $t = 30\ s$.

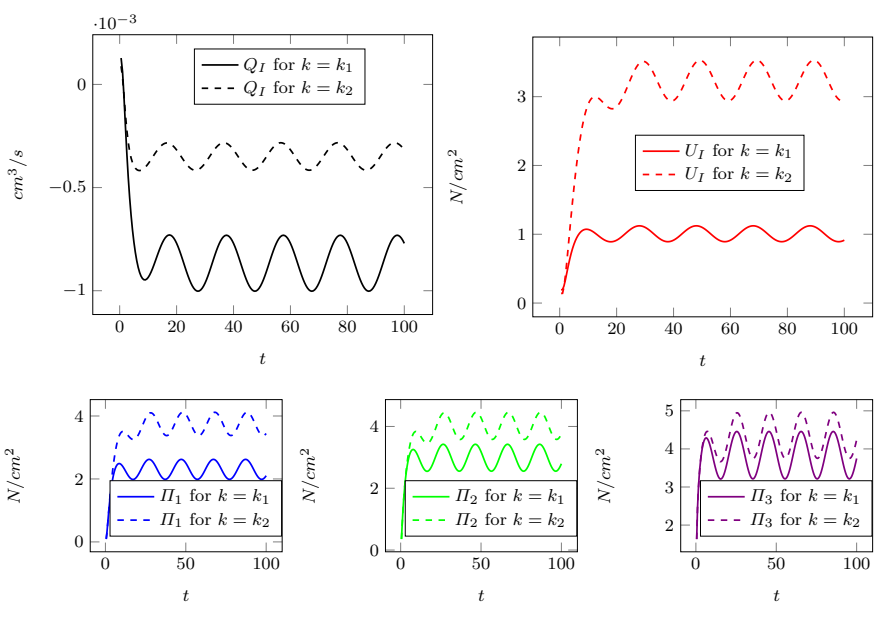


Fig. 16: Simulation results time evolution of the main variables in the circuit (Π_1, Π_2 and Π_3) and on the interface (U_I and Q_I).

relevant examples in industry and medicine. Implementation aspects are discussed and solutions are proposed to mitigate some issues such as scalability.

Besides enjoying standard HDG properties, by incorporating integral boundary conditions, our method accounts naturally for flux boundary conditions as well as natural spatial coupling between 3D+t and 0D+t models.

The suggested method is currently being used in various applications, in particular in the Eye2Brain project [34,35] where it is employed in combination with an operator splitting method for the time discretization [33, Chapter 7] of the multiscale model, as well as in the commissioning process of high field magnets [22].

As mentioned in Section 5, a further comparison with other methods should be conducted to better understand the pros and cons of our approach. This work can also be extended to other types of equations such as elasticity [33, Chapter 24] or the Stokes problem for viscous incompressible fluid [32, Chapter 9]. Finally, we will consider an alternative approach to aggregating faces associated with an integral boundary condition on a single partition in a parallel setting. This will be investigated in forthcoming publications.

Statements & Declarations

Christophe Prud'homme, Romain Hild and Lorenzo Sala wish to thank Vincent Chabannes from Cemosis for many fruitful discussions.

The authors wish to acknowledge the Labex IRMIA from University of Strasbourg for supporting the Eye2Brain project. The PhD thesis of Romain Hild has been also supported by the Labex IRMIA.

Giovanna Guidoboni, Christophe Prud'homme, Lorenzo Sala and Marcela Szopos acknowledge the funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 731063.

Giovanna Guidoboni has been partially supported by NSF-DMS 1853222/2021192. She also would like to disclose that she has received remuneration for serving as a consultant for Foresite healthcare LLC.

Data and meshes for the thermoelectric application are not open: the exact material specifications and geometries are secrets due to the high international competition between USA, Europe, China, Korea and Japan. However close enough models and the software configuration are openly available on Feel++ GitHub repository, see `https://github.com/feelpp/feelpp/tree/develop/toolboxes/thermoelectric/cases/ElectroMagnets`.
The data of the other two applications are openly available in Feel++ repository (`https://github.com/feelpp/feelpp`).

References

1. 2015 International Technology Roadmap for Semiconductors (2016). Http://www.itrs2.net/itrs-reports.html
2. Arnold, D., Brezzi, F.: Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates. Math. Modeling and Numer. Anal. **19**(1), 7–32 (1985)
3. Carichino, L., Guidoboni, G., Szopos, M.: Energy-based operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows: The Stokes case. Journal of Computational Physics **364**, 235–256 (2018)
4. Carichino, L., Guidoboni, G., Szopos, M.: Second-order time accuracy for coupled lumped and distributed fluid flow problems via operator splitting: A numerical investigation. In: Numerical Mathematics and Advanced Applications ENUMATH 2019, pp. 959–967. Springer (2021)
5. Causin, P., Sacco, R.: A discontinuous Petrov–Galerkin method with Lagrangian multipliers for second order elliptic problems. SIAM Journal on Numerical Analysis **43**(1), 280–302 (2005)

6. Cockburn, B.: The Hybridizable Discontinuous Galerkin methods. In: Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures, pp. 2749–2775. World Scientific (2010)
7. Cockburn, B.: Static condensation, hybridization, and the devising of the HDG methods. In: Building bridges: connections and challenges in modern approaches to numerical partial differential equations, pp. 129–177. Springer (2016)
8. Cockburn, B., Dong, B., Guzmán, J.: A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. Math. Comp. **77**(264), 1887–1916 (2008)
9. Cockburn, B., Dong, B., Guzmán, J., Restelli, M., Sacco, R.: A Hybridizable Discontinuous Galerkin method for steady-state convection-diffusion-reaction problems. SIAM Journal on Scientific Computing **31**(5), 3827–3846 (2009)
10. Cockburn, B., Gopalakrishnan, J., Lazarov, R.: Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM Journal on Numerical Analysis **47**(2), 1319–1365 (2009)
11. Cockburn, B., Guzmán, J., Wang, H.: Superconvergent discontinuous Galerkin methods for second-order elliptic problems. Math. Comp. **78**(265), 1–24 (2009)
12. Colburn, A.: A method of correlating forced convection heat transfer data and a comparison with fluid friction. Trans. AIChE **29**, 174–210 (1933)
13. Daversin, C.: Reduced basis method applied to large non-linear multi-physics problems: Application to high field magnets design. Ph.D. thesis, Université de Strasbourg (2016)
14. Daversin, C., Prud'homme, C., Trophime, C.: Full three-dimensional multiphysics model of high-field polyhelices magnets. IEEE transactions on applied superconductivity **26**(5) (2016)
15. Dittus, F., Boelter, L.: Heat transfer in automobile radiators of the tubular type. International Communications in Heat and Mass Transfer **12**(1), 3 – 22 (1985)
16. Formaggia, L., Quarteroni, A., Veneziani, A.: Cardiovascular Mathematics: Modeling and simulation of the circulatory system, vol. 1. Springer Science & Business Media (2010)
17. Glowinski, R.: Finite element methods for incompressible viscous flow. Handbook of numerical analysis **9**, 3–1176 (2003)
18. Grisvard, P.: Elliptic problems in non smooth domains. No. 24 in Monographs and studies in Mathematics. Pitman (1985)
19. Gröger, K.: A $w^{1,p}$-estimate for solutions to mixed boundary value problems for second order elliptic differential equations. Math. Ann. **283**, 679–687 (1989)
20. Harris, A., Guidoboni, G., Siesky, B., Mathew, S., Verticchio-Vercellin, A., Rowe, L., Arciero, J.: Ocular blood flow as a clinical observation: Value, limitations and data analysis. Progress in Retinal and Eye Research **78**(100841) (2020)
21. Harris, A., Kagemann, L., Ehrlich, R., Rospigliosi, C., Moore, D., Siesky, B.: Measuring and interpreting ocular blood flow and metabolism in glaucoma. Canadian Journal of Ophthalmology **43**(3), 328–336 (2008)
22. Hild, R.: Optimization and control of high fields magnets. Theses, Université de Strasbourg (2020). URL `https://tel.archives-ouvertes.fr/tel-03025312`
23. Jackson, J.D.: Classical Electrodynamics Third Edition, third edn. Wiley (1998)
24. Karypis, G., Kumar, V.: MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. `http://www.cs.umn.edu/~metis` (2009)
25. Kirby, R.M., Sherwin, S.J., Cockburn, B.: To CG or to HDG: A comparative study. Journal of Scientific Computing **51**(1), 183–212 (2012)
26. Leske, M.C.: Open-angle glaucoma—an epidemiologic overview. Ophthalmic epidemiology **14**(4), 166–172 (2007)
27. Markowich, P., Ringhofer, C., Schmeiser, C.: Semiconductor Equations. Springer-Verlag (1990)
28. Montgomery, D.: Solenoid magnet design: the magnetic and mechanical aspects of resistive and superconducting systems. Solenoid magnet design. Wiley-Interscience (1969). URL `https://books.google.fr/books?id=D1138LYUoXAC`
29. Muller, R., Kamins, T., Chan, M.: Device Electronics for Integrated Circuits. Wiley (2003). URL `https://books.google.it/books?id=x64knwEACAAJ`
30. Nicolaides, R.: Existence, uniqueness and approximation for generalized saddle point problems. SIAM Jour. on Numer. Anal. **19**(2), 349–357 (1982)

31. Prud'homme, Christophe, Chabannes, Vincent, Doyeux, Vincent, Ismail, Mourad, Samake, Abdoulaye, Pena, Goncalo: Feel++ : A computational framework for Galerkin methods and advanced numerical methods. ESAIM: Proc. **38**, 429–455 (2012)
32. Quarteroni, A., Valli, A.: Numerical Approximation of Partial Differential Equations. Lecture Notes in Mathematics. Springer-Verlag (1994). URL https://books.google.it/books?id=MB23keacoWMC
33. Sala, L.: Mathematical modelling and simulation of ocular blood flows and their interactions. Ph.D. thesis, Université de Strasbourg (2019). URL https://tel.archives-ouvertes.fr/tel-02284233/file/Sala_Lorenzo_2019_ED269.pdf
34. Sala, L., Prud'Homme, C., Guidoboni, G., Szopos, M.: Ocular mathematical virtual simulator: A hemodynamical and biomechanical study towards clinical applications. Journal of Coupled Systems and Multiscale Dynamics **6**(3), 241–247 (2018)
35. Sala, L., Prud'Homme, C., Prada, D., Salerni, F., Trophime, C., Chabannes, V., Szopos, M., Repetto, R., Bertoluzza, S., Sacco, R., et al.: Patient-specific virtual simulator of tissue perfusion in the lamina cribrosa. p. 727. Invest. Ophthalmol. Vis. Sci. (2017)
36. Selberherr, S.: Analysis and Simulation of Semiconductor Devices. Springer-Verlag (1984)
37. Silberberg, Y., Lahini, Y., Bromberg, Y., Small, E., Morandotti, R.: Universal correlations in a nonlinear periodic 1D system. Physical Review Letters **102**(23) (2009)
38. de Veubeke, B.M.F.: Displacement and equilibrium models in the finite element method. In: O. Zienkiewicz, G. Holister (eds.) Stress Analysis, pp. 145–197. John Wiley & Sons, New York (1965)