Abstract. Finite element methods for Maxwell’s equations are highly sensitive to the conformity of approximation spaces, and non-conforming methods may cause loss of convergence. This fact leads to an essential obstacle for almost all the interface-unfitted mesh methods in the literature regarding the application to Maxwell interface problems, as they are based on non-conforming spaces. In this work, a novel immersed virtual element method for solving a 3D Maxwell interface problem is developed, and the motivation is to combine the conformity of virtual element spaces and robust approximation capabilities of immersed finite element spaces. The proposed method is able to achieve optimal convergence for a 3D Maxwell interface problem. To develop a systematic framework, the $H^1$, $H(\text{curl})$ and $H(\text{div})$ interface problems and their corresponding problem-orientated immersed virtual element spaces are considered all together. In addition, the de Rham complex will be established based on which the HX preconditioner can be used to develop a fast solver for the $H(\text{curl})$ interface problem.

Key words. Maxwell’s equations, Interface problems, $H^1$, $H(\text{curl}; \Omega)$ and $H(\text{div})$-elliptic equations, virtual element methods, immersed finite element methods, maximal angle conditions, de Rham complex, fast solvers

1. Introduction. In this article, we shall develop a systematic framework to construct three-dimensional (3D) $H^1$, $H(\text{curl})$, and $H(\text{div})$ virtual element spaces involving discontinuous coefficients, referred to as the immersed virtual element (IVE) spaces, that can be used to solve the corresponding interface problems described in Section 1.1 on unfitted meshes. The proposed method is particularly important for Maxwell interface problems as the current unfitted mesh methods in the literature have essential difficulty for handling $H(\text{curl})$ systems, see the detailed discussion in Section 1.2.

1.1. Model problems. Let $\Omega \subseteq \mathbb{R}^3$ denote a bounded modeling domain, and a subdomain $\Omega^- \subseteq \Omega$ contains the medium which has physical property distinguished from the background medium. The surface $\Gamma = \partial \Omega^-$ is called interface and assumed to be sufficiently smooth with $\Omega^+ = \Omega \setminus \Omega^-$, with the normal vector $\mathbf{n}$ pointing from $\Omega^+$ to $\Omega^-$. We introduce two discontinuous piecewise constant parameters representing the medium properties:

\begin{align}
\alpha = \begin{cases}
\alpha^- & \text{in } \Omega^-,\\
\alpha^+ & \text{in } \Omega^+,
\end{cases}
\quad \beta = \begin{cases}
\beta^- & \text{in } \Omega^-,\\
\beta^+ & \text{in } \Omega^+.
\end{cases}
\end{align}

The classic $H^1$-elliptic interface problem reads as

\begin{equation}
- \nabla \cdot (\beta \nabla u) = f \quad \text{in } \Omega^- \cup \Omega^+,
\end{equation}

with $f \in L^2(\Omega)$, subject to a certain boundary condition on $\partial \Omega$ and the jump conditions

\begin{align}
[u]_\Gamma & := u^+ - u^- = 0, \\
[\beta \nabla u \cdot \mathbf{n}]_\Gamma & := \beta^+ \nabla u^+ \cdot \mathbf{n} - \beta^- \nabla u^- \cdot \mathbf{n} = 0,
\end{align}

where the parameter $\beta$ may represent, for example, the conductivity in electrical applications [51, 68] and the dielectric constant in Poisson-Boltzmann equations [31, 72].

For Maxwell interface problems, we consider the following curl curl-elliptic model

\begin{equation}
\text{curl}(\alpha \text{curl } \mathbf{u}) + \beta \mathbf{u} = \mathbf{f} \quad \text{in } \Omega = \Omega^- \cup \Omega^+,
\end{equation}

which is derived from discretizing a time-dependent Maxwell system in which the magnetic field is eliminated. Here for simplicity we assume $\mathbf{f} \in H(\text{div}; \Omega)$. If piecewise constant parameters $\epsilon$, $\sigma$ and $\mu$ represent the electric permeability, conductivity and magnetic permeability of the medium respectively, then $\alpha = \mu^{-1}$ and $\beta = \sigma^{-1}$. The proposed method is able to achieve optimal convergence for a 3D Maxwell interface problem.
\( \beta = \epsilon \Delta t^{-2} + \sigma \Delta t^{-1} \) in (1.1). Due to the presence of the interface, the following jump conditions are imposed for the electrical field \( \mathbf{u}^\pm \) at the interface:

\[
\begin{align*}
(1.5a) & \quad [\mathbf{u} \times \mathbf{n}]_\Gamma := \mathbf{u}^+ \times \mathbf{n} - \mathbf{u}^- \times \mathbf{n} = 0, \\
(1.5b) & \quad [\alpha \, \text{curl} \, \mathbf{u} \times \mathbf{n}]_\Gamma := \alpha^+ \text{curl} \, \mathbf{u}^+ \times \mathbf{n} - \alpha^- \text{curl} \, \mathbf{u}^- \times \mathbf{n} = 0, \\
(1.5c) & \quad [\beta \mathbf{u} \cdot \mathbf{n}]_\Gamma := \beta^+ \mathbf{u}^+ \cdot \mathbf{n} - \beta^- \mathbf{u}^- \cdot \mathbf{n} = 0.
\end{align*}
\]

We highlight that \( \beta \) has a similar physical meaning to the \( H^1 \) interface problem, if the latter one also represents electric conductivity. In fact, such physical relations are naturally encoded in a de Rham complex, see the discussion in Section 3. Electromagnetic interface problems are of great importance due to a large variety of science and engineering applications. Typical examples include electromagnetic motors and actuators involving metal-air or metal-metal interface [20, 40] and electromagnetic inverse scattering [38, 61] that use electromagnetic waves to detect objection. Solving the \( \mathbf{H}(\text{curl}) \) interface problem with an optimal convergence is a challenging goal that conventional unfitted mesh methods fail to meet, see Section 1.2 below. This is the main motivation for the present research.

At the last, the \( \mathbf{H}(\text{div}) \) interface problem is given by

\[
(1.6) \quad -\nabla \text{div}(\mathbf{u}) + \alpha \mathbf{u} = \mathbf{f} \quad \text{in} \ \Omega^- \cup \Omega^+,
\]

with \( \mathbf{f} \in \mathbf{H}(\text{curl}; \Omega) \), subject to a certain boundary condition on \( \partial \Omega \) and the jump conditions

\[
\begin{align*}
(1.7a) & \quad [\mathbf{u} \cdot \mathbf{n}]_\Gamma := \mathbf{u}^+ \cdot \mathbf{n} - \mathbf{u}^- \cdot \mathbf{n} = 0, \\
(1.7b) & \quad [\alpha \mathbf{u} \times \mathbf{n}]_\Gamma := \alpha^+ \mathbf{u}^+ \times \mathbf{n} - \alpha^- \mathbf{u}^- \times \mathbf{n} = 0, \\
(1.7c) & \quad [\text{div}(\mathbf{u})]_\Gamma := \text{div}(\mathbf{u}^+) - \text{div}(\mathbf{u}^-) = 0.
\end{align*}
\]

The system comes from a mixed finite element method with a gradient formulation [5]. The related interface problem is discussed in [49]. Again, the parameter \( \alpha \) here is inherited from the \( \mathbf{H}(\text{curl}) \) case.

### 1.2. Challenges of Maxwell interface problems on unfitted mesh.

For conforming finite element methods (FEMs) to perform optimally on a mesh, the mesh has to fit or approximate the interface geometry. However, an efficient high-quality 3D mesh generation itself remains a challenging problem, which is particularly expensive for complicated geometries. A promising cure is to generate a cheap background unfitted mesh and then further triangulate those elements cut by the interface [36]. The modification is highly efficient since it only needs to be done locally around the interface. As this approach cannot yield shape regular elements near the interface, the maximum angle condition justifies it by relaxing the geometry constraints. The interpolation estimates based on the maximum angle condition have been widely studied for Lagrange elements [8], Raviart-Thomas elements [1, 60], and 3D Nédélec elements [23]. Indeed, this approach is very successful in the two-dimensional (2D) case [35] as an admissible local triangulation satisfying the maximum angle condition always exists for a shape-regular background mesh [27, Lemma 3.1] and [32, Proposition 2.4]. Nevertheless, in the 3D case, these locally re-meshed triangulations may not form a globally admissible mesh, as it may not necessarily satisfy the Delaunay property and/or maximum angle conditions due to slivers [64].

To overcome the difficulty of 3D triangulation, the authors in [32] propose to use polyhedral elements cut directly from the background Cartesian meshes to fit the interface rather than a further triangulated tetrahedral mesh. To handle the polyhedral shape, a virtual element method (VEM) [11] is used for discretization. In fact, \( \mathbf{H}(\text{curl}) \) virtual element spaces have been constructed and applied to Maxwell’s equations in [15, 12, 13, 16]. However, the treatment for \( \mathbf{H}(\text{curl}) \) problems is quite different. The meta-framework for VEM developed in [22, 25] cannot be directly used to obtain even optimal error estimates, and some more delicate techniques are needed, e.g., [15, 28, 12]. For interface problems, an extra layer of difficulty is to make error bounds robust with respect to potential anisotropic element shapes. In [25] for the 2D \( H^1 \) case, a closing rigorous analysis is given for anisotropic elements generated from Cartesian meshes cut by the interface.

Another direction to circumvent the mesh generation issue is to modify finite element (FE) spaces such that the new spaces can capture the jump behaviors in an optimal sense on unfitted meshes. There have been extensive works in this direction including immersed finite element (IFE) methods [2, 58, 44], CutFEM or Nitsche’s penalty methods [19, 24, 62, 56, 59], multiscale FEMs [37] and so on, which are widely applied to various interface problems. We also refer readers to FDTD methods [73] based on finite difference formulation.
for Maxwell’s equations. For almost all the unfitted mesh methods in the literature, the modification is usually applied element-wise or piecewise relative to the interface. Thus, this practice results in discontinuities across non-interface element boundaries or at the interface. Such non-conformity can be handled by penalties on element boundaries to impose continuity such as in the IFE methods [43, 45, 58], or on the interface itself to impose jump conditions such as the Nitsche’s methods [19, 24, 62, 56, 59]. With suitable penalties, robust optimal convergence rates can be indeed obtained for \( H^1 \)-type interface problems but not the considered Maxwell interface problem.

The most drastic difference stems from the underlying Sobolev space \( H^s(\text{curl}; \Omega) \). In particular, for many non-conforming methods, one needs to estimate the penalty term, and standard techniques by [61, Lemma 5.52] lead to estimates as follows

\[
h^{-1}\|u - \pi_F u\|_{L^2(F)} \lesssim h^{s-1}\|u\|_{H^s(\text{curl}; K)},
\]

where \( \pi_F \) is a certain projection operator on a face \( F \) of an element \( K \). The order in (1.8) implies that even a moderate regularity \( s = 1 \) yields no approximation accuracy. It is worthwhile to mention that even standard discontinuous Galerkin (DG) methods for Maxwell’s equations suffer from this issue, and the analysis relies on a \( \mathbf{H}(\text{curl}) \)-conforming subspace of the broken DG space [53, 52] with sufficient approximation properties. For unfitted mesh methods in the literature, this problem becomes more severe, since such a conforming subspace may not exist. Numerically, the loss of convergence has been observed and reported in a series of work [29, 30, 47] for the \( \mathbf{H}(\text{curl}) \) interface problem. So we believe that the difficulty is essential rather than caused by limitation of analysis techniques.

More essentially, the issue comes from the scaling factor \( h^{-1} \) which is needed for stabilization but too strong for the space \( \mathbf{H}^1(\text{curl}; \Omega) \). In fact, for a Lipschitz domain \( D \), it is well-known that the trace of \( H^1(D) \) is in \( H^{1/2}(\partial D) \). While for \( \mathbf{H}(\text{curl}; D) \), the tangential trace is merely in \( \mathbf{H}^{1/2}(\text{div}; \partial D) \), which should lead to different scaling factors for the stabilization terms on faces. Here we refer readers to [21] for the analysis on the relation between the scaling factor of a non-conforming method and function’s regularity. We note that various VEM [17, 15, 27, 16] can use the “right” scaling \( h \) in (1.8) that achieves optimal convergence and still results in the stability constant independent of \( h \).

In conclusion, developing unfitted mesh methods for the \( \mathbf{H}(\text{curl}) \) interface problem is much more challenging than its \( H^1 \) counterpart. For non-matching mesh methods, some work can obtain optimal convergence under the usual \( \mathbf{H}^1(\text{curl}) \)-regularity by making a certain assumption of meshes being coupled at the interface, see [54, 34]. For many unfitted mesh methods, the meshes or spaces are generally completely broken, then at least the \( H^2 \)-regularity has to be assumed to achieve optimal convergence, for example [18, 59]. In [47] for the 2D case, a Petrov-Galerkin method is developed that can achieve optimal convergence but result in a non-symmetric scheme. The robust optimal convergence for VEM is established in [27] but relies on a “virtual” triangulation satisfying the maximum angle condition which may not be available in 3D. Therefore, to our best knowledge, currently there seems no satisfactory methodology for the problem considered.

1.3. A Novel Method. To develop unfitted mesh methods for the \( \mathbf{H}(\text{curl}) \) interface problem, it is preferable to use a conforming space, and in the meantime, this space must admit sufficient approximation capabilities robust with respect to the anisotropy of subelements. This consideration motivates us to combine the conformity of virtual element spaces and the approximation capabilities of IFE spaces. In our previous work [28], we have successfully realized this idea for the 2D case, which is referred to as immersed virtual element (IVE) methods.

The fundamental idea is to impose local PDEs on interface elements to enforce both conformity and jump condition of which the solutions are used as the spaces for discretization. Remarkably, the IVE spaces can be understood as a special family of \( H^1, \mathbf{H}(\text{curl}) \) and \( \mathbf{H}(\text{div}) \) virtual element spaces [11, 12, 13, 15] with discontinuous coefficients. For the \( H^1 \) case, it is also exactly the space of special FEMs by Babuška et al. in [9, 10] for a simple 1D case, and it becomes the multiscale FE space in [37] for higher dimensional cases where the local PDEs are solved on sub-grids. Nevertheless, the proposed IVE method follows the framework of VEM: the local PDEs need not be solved exactly, certain projections are computed instead with sufficient approximation capability to capture the jump conditions. It can successfully yield optimal convergence rates for the \( \mathbf{H}(\text{curl}) \) interface problem, which has been rigorously proved in the 2D case [28]. In this work, we focus on the development of the IVE spaces, the scheme and the implementation in the 3D case. We leave the theoretical part to another coming work.
Developing 3D IVE spaces is significantly more complicated, especially for the $\mathbf{H}(\text{curl})$ space. An immediate question is how to design appropriate div-curl systems as local problems with discontinuous coefficients that have a rigorous well-posedness. Here, special attention must also be paid to the designing the local problems such that their solutions have computable projections to IFE spaces. The key is to modify the source terms for the local problems, and construct certain weighted projections with regarding the weights as Hodge star operators. The second issue is to design suitable trace spaces on element boundary whose functions serve as the boundary conditions for those local problems. The trace spaces need to provide sufficient and robust approximation properties. In the 2D case, the boundary space consists of piecewise constants or linear functions on each edge, the simplicity of which attributes to the trivial geometry of the element boundaries, see [14, 25, 28]. However, in the 3D case, it becomes much more obscure. For the classical virtual spaces [11, 12, 13, 15], the trace spaces are generally formulated by solutions of some extra 2D local problems defined on polygonal faces. In this work, we propose a rather different yet simpler approach: to use the standard FE spaces defined on a 2D triangulation satisfying the maximum angle condition on each element face. Such a triangulation does not only benefit the robust approximation property due to the maximum angle condition, but also facilitates the code development leading to an efficient implementation with suitable data structures. Hence, the present research is, in fact, a combination of the three classical methodology: VEM, IFE and FEM, towards solving the challenging Maxwell interface problem.

Although our focus is on Maxwell interface problems, we shall develop a systematic framework for all the $H^1$, $\mathbf{H}(\text{curl})$ and $\mathbf{H}(\text{div})$ interface problems contributing to a solid mathematical foundation. They can be related by the following de Rham complex, and are shown to have the usual nodal, edge and face degrees of freedom (DoFs) through which both the exact sequence and the commutative property can be established.

\[
\begin{align*}
\mathbb{R} & \overset{\cdot}{\longrightarrow} H^1(\beta; T_h) \overset{\text{grad}}{\longrightarrow} \mathbf{H}^1(\text{curl}, \alpha; T_h) \overset{\text{curl}}{\longrightarrow} \mathbf{H}^1(\text{div}, \alpha; T_h) \overset{\text{div}}{\longrightarrow} H^1(T_h) \longrightarrow 0 \\
\mathbb{R} & \overset{\cdot}{\longrightarrow} V^n_h \overset{\text{grad}}{\longrightarrow} V^n_h \overset{\text{curl}}{\longrightarrow} V^n_h \overset{\text{div}}{\longrightarrow} Q^n_h \longrightarrow 0
\end{align*}
\]

Another layer of challenge for 3D interface problems is the appropriate fast solver. Multigrid methods are widely used, and we refer readers to [71] for FEM and [32] for VEM, both of which study the $H^1$-interface problem. For Maxwell’s equations, the fast solvers are even more challenging [69, 50] due to the non-trivial kernel space, which is another motivation to lay out the following de Rham complex for the proposed spaces. In this work, we generalize multigrid-based HX preconditioner [50, 33] for regular $\mathbf{H}(\text{curl})$ problem to the interface case. Moreover, for fitted mesh methods the condition numbers may still suffer from the possible anisotropic element shapes, even though the error bounds are robust. In [71], the DoFs near the interface and in the background mesh are split to form “fine-coarse” block matrices, thus an optimal two-level solver is developed. In this paper, a block diagonal smoother is proposed to handle the anisotropic element shape near the interface, similar to the practice in [71]. To our best knowledge, this is the first research towards applying the HX preconditioner to VEM, and also the first fast solver for unfitted mesh methods for solving Maxwell interface problems. Numerical results demonstrate that the solver is robust with respect to mesh size and small-cutting elements.

This article has additional 7 sections. In the next section, we introduce the meshes especially the element boundary triangulation. In Section 3, we describe the desired Sobolev spaces encoding jump conditions that are to be approximated. In Sections 4 and 5, we introduce IFE spaces and IVE spaces, respectively. In Section 7, we describe the computation scheme, fast solvers and implementation aspects. In the last section, we present a group of numerical experiments.

2. Meshes. In this article, we focus on a given interface-independent shape-regular tetrahedral mesh of $\Omega$, but the proposed method can be also adapted to any Cartesian cubic meshes. This tetrahedral mesh is referred as the background mesh, and is denoted by $T_h$. If an element in $T_h$ intersects the interface, then it is called an interface element, or a non-interface element otherwise. The collection of interface elements is denoted as $T^n_h$. For each $K \in T^n_h$, we denote $\Gamma^K = \Gamma \cap K$. Let $T^n_h$ be the set of non-interface elements.

One of the critical ingredients to formulate the $H^1$, $\mathbf{H}(\text{curl})$ and $\mathbf{H}(\text{div})$ local interface problems is to impose appropriate boundary conditions on element boundary. Different from the prevailing approach in the literature, [32] proposes a novel approach by using exclusively square and triangular faces in interface polyhe-
dra. In [32], a Delaunay triangulation routine is called for the nodes in the background mesh, interface nodes, and some added vertices near the interface, then the triangular faces are extracted for their corresponding polyhedra. In this work, similar to the practice in [32], standard FE functions on a 2D triangulation of any given element face are used as the boundary conditions. Hence, we make a fundamental assumption called interface fitted boundary triangulation:

(A) For each interface element $K$, each of its face admits a triangulation satisfying the maximum angle condition. The triangles are formed by only the vertices of $K$ and/or the cutting points of the interface, i.e., there are no newly-added interior vertices to form the edges. If a face is cut by the interface, then this triangulation must be fitted to the interface.

We illustrate the Assumption A in Figure 2.1: each face of an interface element is partitioned into multiple triangles by the newly added edges including the one connecting the cutting points (red points in the figure). It can be understood that a local 2D fitted mesh satisfying the maximum angle condition is generated around the mesh but only on faces. Although, as aforementioned in the introduction, generating a 3D fitted mesh may be difficult or even impossible in certain situations, it is much easier to generate a 2D fitted mesh. In particular, since the considered original background tetrahedral meshes only have triangular faces, the boundary triangulation with the maximum angle condition is always guaranteed by [27, Lemma 3.1]. See, for example, the left plot in Figure 2.1 if the element is cut by the interface only once. If cubic meshes are used, then [32, Proposition 2.4] guarantees an admissible boundary triangulation. But unlike [32] where the element is divided into two polyhedrons, here the cut tetrahedron is still treat as one element with more than 4 triangular faces.

We highlight that the interface fitted boundary triangulation is able to link the fitted 2D and unfitted 3D meshes, and also bridges the standard 2D FE spaces and 3D virtual element spaces. Our previous work in the 2D case [28] suggests that it is also one of the keys to overcome suboptimal convergence caused by non-conforming spaces for Maxwell’s equations, as well as help in anisotropic analysis for the virtual spaces. In addition, the proposed boundary triangulation, in fact, greatly benefits the computation. One of the difficult aspects of implementing polytopal finite element approximation is the ever-changing number of DoFs in an element. In our approach, since only triangular faces are present in every interface or non-interface element, the assembling can be uniformly handled fixed-width matrices in the face-oriented data structure, please refer to Section 7.2 for details, see also [17] for a face-based approach.

Another advantage of the proposed method is the flexibility to handle complex interface element geometry. Specifically, on elements that are cut by interface with multiple times, e.g., see the right plot in Figure 2.1, the proposed IVE spaces can be easily constructed as long as an admissible boundary triangulation can be constructed.

3. Some Sobolev Spaces. In this section, we describe a group of modified Sobolev spaces that incorporate the interface conditions. Let us first recall some standard spaces. Given a subdomain $D \subseteq \Omega$, for $s \geq 0$, we let $H^s(D)$ be the standard scalar Sobolev space and $H^s(\Omega) := (H^s(D))^3$. Now introduce

\begin{equation}
H^s(\text{curl}; D) = \{ u \in H^s(D) : \text{curl } u \in H^s(D) \},
\end{equation}
Let by results with general coefficients can be found in [65, 66, 41].

To pursue a rigorous definition of the IVE spaces, we first consider the well-posedness of some div-curl the interface is from the continuity of the mapped forms. Construction of the desired virtual spaces is to 

\begin{equation}
\beta \text{curl } u \in H(\text{curl}; \Omega) \cap \{\mathbf{u} \in H(\text{curl}; \Omega) : \beta \mathbf{u} \in H(\text{div}; K)\},
\end{equation}

\begin{equation}
H^1(\Omega) \cap \{u \in H^2(\Omega) : \beta \nabla u \in H(\text{div}; K), \forall K \in \mathcal{T}_h\},
\end{equation}

\begin{equation}
H^1(\text{curl}; \Omega) \cap \{\mathbf{u} \in H(\text{curl}; \Omega) : \beta \mathbf{u} \in H(\text{div}; K)\},
\end{equation}

\begin{equation}
H^1(\mathcal{T}_h) = L^2(\Omega) \cap \{u \in H^1(\Omega), \forall K \in \mathcal{T}_h\}.
\end{equation}

Note that on a non-interface element \(K\) the conditions are trivial since they are just consequences of \(H^2(\Omega)\), \(H^1(\text{curl}; \Omega)\), and \(H^1(\text{div}; \Omega)\). On an interface element \(K\) those conditions exactly encode both the conformity and interface information. To see the relation more clearly, we let \(\mathbf{u}\) be a Lipschitz polyhedral domain which is simply-connected, and let an interface \(\gamma\) separate \(\omega\) and interface information. To see the relation more clearly, we let \(\mathbf{u}\) be a Lipschitz polyhedral domain which is simply-connected, and let an interface \(\gamma\) separate \(\omega\) and interface information.

In Diagram (3.3), \(\alpha\) and \(\beta\) could be understood as Hodge star operators mapping \(k\)-forms to \((3-k)\)-forms for \(k = 2, 1\), respectively. Take \(\beta : H(\text{curl}; \Omega, \alpha, \beta; K) \rightarrow H(\text{div}; K)\) as an example. A function \(\mathbf{u}\) in \(H(\text{curl}; \Omega, \alpha, \beta; K)\) can be thought of as a vector proxy of a 1-form. Then \(\beta \mathbf{u} \in H(\text{div}; K)\) is a 2-form. The jump condition on the interface is from the continuity of the mapped forms. Construction of the desired virtual spaces is to mimic this diagram in the discretized level.

Note that the proposed global problems as well as local problems all involve discontinuous coefficients. In order to pursue a rigorous definition of the IVE spaces, we first consider the well-posedness of some div-curl systems with discontinuous coefficients. The systems with constant coefficients are discussed in [4], and the results with general coefficients can be found in [65, 66, 41].

In the following discussion, given any face \(F\) in the mesh, we shall denote the tangential component of \(\mathbf{u}\) by \(\mathbf{u}^T|_F\) for admissible \(\mathbf{u}\) defined in the bulk \(\omega\) such that \(F \subseteq \partial \omega\), and we will drop \(\mid F\) if there is no danger of causing confusion. In addition, we will also frequently use the 2D rotation operator denoted by \(\text{rot}_F\) on \(F\). Let \(\nabla_s\) denote the surface gradient. Then, for a function \(\varphi\) defined on \(F\), \(\text{rot}_F\) is defined in the distributional sense such that

\begin{equation}
\langle \text{rot}_F \varphi, v \rangle_F := \langle \varphi, \nabla_s v \times \mathbf{n} \rangle_F, \quad \forall v \in H^1_0(F).
\end{equation}

For each subdomain \(\omega \subseteq \Omega\), and \(\varphi\) defined on \(\partial \omega\), \(\text{rot}_{\partial \omega} \varphi\) can be defined similarly,

\begin{equation}
\langle \text{rot}_{\partial \omega} \varphi, v \rangle_{\partial \omega} := \langle \varphi, \nabla_s v \times \mathbf{n} \rangle_{\partial \omega}, \quad \forall v \in H^1(\omega),
\end{equation}

while it can be verified that \(\text{rot}_{\partial \omega} \varphi|_F = \text{rot}_F \varphi\). Here \(\langle \cdot, \cdot \rangle\) denotes the usual pairing between \(H^{-1/2}(\partial \omega) - H^{1/2}(\partial \omega)\), and \(\langle \cdot, \cdot \rangle_F\) defined similarly for \(F \subset \partial \omega\). In particular, the well-known formula states

\begin{equation}
\text{curl} \mathbf{u} \cdot \mathbf{n}_F = \text{rot}_F \mathbf{u}, \quad \text{for } \mathbf{u} \in H(\text{curl}; \omega), \quad \text{on } F \subset \partial \omega,
\end{equation}

where \(\mathbf{n}_F\) is the exterior unit normal vector of \(F\) with respect to \(\omega\).

**Lemma 3.1.** Let \(\omega\) be a Lipschitz polyhedral domain which is simply-connected, and let an interface \(\gamma\) separate \(\omega^+\) and \(\omega^-\) and define a piecewise constant function \(b = b^+ > 0\) in \(\omega^+\). For the data functions \(f \in H(\text{div}; \omega) \cap \ker(\text{div})\), \(h \in L^2(\omega)\) and \(g \in H^{-1/2}(\partial \omega)\) such that the compatibility condition holds:

\begin{equation}
\int_\omega h \, \text{d}x = \langle g, 1 \rangle_{\partial \omega},
\end{equation}
then the following problem admits a unique solution \( \varphi \in H(\text{div}; \omega) \) and \( b\varphi \in H(\text{curl}; \omega) \)

\[(3.8) \quad \text{curl}(b\varphi) = f, \quad \text{div}(\varphi) = h \quad \text{in} \omega, \quad \varphi \cdot n = g \quad \text{on} \partial \omega.
\]

If additionally \( g \in H^{1/2}(\partial \omega) \), \( \omega \) is convex, and \( \gamma \) is a closed surface that is sufficiently smooth and does not intersect the boundary, then \( \varphi \in H^1(\omega^\pm) \).

**Proof.** To handle the data \((h, g)\), we consider the Neumann problem:

\[(3.9) \quad \text{div}(b^{-1}\nabla v) = h \quad \text{in} \omega, \quad b^{-1}\nabla v \cdot n = g \quad \text{on} \partial \omega.
\]

Due to the compatibility condition \((3.7)\), this problem admits a unique solution \( v \in H^1(\omega) \) up to a constant.

Note that the solution \( v \) satisfies \([v] = 0\) and \([b^{-1}\nabla v \cdot n] = 0\) on the interface \( \gamma \).

To handle the data \( f \), we closely follow the argument of [55, Theorem 5.2]. Thanks to \( \text{div}(f) = 0 \), we know by [4, Theorem 3.12] that there exists a vector potential \( w \in H(\text{curl}; \omega) \cap H(\text{div}; \omega) \) such that

\[(3.10) \quad \text{curl} w = f, \quad \text{div} w = 0, \quad \text{in} \omega, \quad w \cdot n = 0, \quad \text{on} \partial \omega,
\]

where all the equations hold in the \( L^2 \) sense. In fact, as \( \omega \) is a Lipschitz polyhedron, by [4, Proposition 3.7] we have \( w \in H^s(\omega) \) with \( s > 1/2 \), and thus \( w \cdot n \in L^2(\partial \omega) \). Therefore, we also have \( b^{-1}w \cdot n = 0 \) on \( \partial \omega \).

Using \( w \), we let \( w \) be the solution of the following elliptic interface problem:

\[(3.11a) \quad \text{div}(b^{-1}\nabla w^\pm) = -\text{div}(b^{-1}w^\pm) \quad \text{in} \omega^\pm,
\]

\[(3.11b) \quad [w]_\gamma = 0, \quad [b^{-1}\nabla w \cdot n]_\gamma = -[b^{-1}w \cdot n]_\gamma,
\]

\[(3.11c) \quad b^{-1}\nabla w \cdot n = 0, \quad \text{on} \partial \omega,
\]

which certainly has a unique solution up to a constant.

Now, we can characterize \( \varphi \) by

\[(3.12) \quad b\varphi = \nabla v + w + \nabla w,
\]

and verify that \( \varphi \) is a solution to \((3.8)\) and belongs to the desired space. First, it follows from the continuity condition in the interface problems \((3.8)\) and \((3.11)\) and \( w \in H(\text{curl}; \Omega) \) that \( b\varphi \in H(\text{curl}; \omega) \). The flux jump condition in \((3.9)\) and \((3.11b)\) implies \( \varphi \in H(\text{div}; \omega) \). Applying \text{curl} to \((3.12)\), we obtain \( \text{curl}(b\varphi) = \text{curl} w = f \). Applying \text{div} to \( \varphi \) and using \((3.9)\) and \((3.11a)\), we obtain \( \text{div} \varphi = h \). On \( \partial \omega \), the boundary condition \( \varphi \cdot n = 0 \) follows from the definition of \( v \) and \( w \) and the fact \( b^{-1}w \cdot n = 0 \).

Suppose \( \omega \) is convex and \( \gamma \) is smooth which does not touch \( \partial \omega \). Theorem 2.17 in [4] implies \( w \in H^1(\omega) \) which leads to \( w \cdot n \in H^{1/2}(\gamma) \). Then, by [7] the elliptic equations in \((3.9)\) and \((3.11)\) both admit solutions \( v, w \in H^1(\Omega) \cap H^2(\Omega^\pm) \). Thus, we obtain \( \varphi \in H^1(\Omega) \cap H^2(\Omega^\pm) \). \( \Box \)

**Lemma 3.2.** Given a simple-connected domain \( \omega \) with Lipschitz boundary, let an interface \( \gamma \) separates \( \omega \) into \( \omega^\pm \) and define a piecewise constant function \( a = \alpha^\pm > 0 \) in \( \omega^\pm \). For the data functions \( f \in H(\text{div}; \omega) \cap \ker(\text{div}), h \in L^2(\omega) \) and \( g \in H^{-1/2}(\partial \omega) \) such that the compatibility condition holds:

\[(3.13) \quad \langle f \cdot n, v \rangle_{\partial \omega} = \langle \text{rot}_{\partial \omega}(n \times g), v \rangle_{\partial \omega} \quad \forall v \in H^1(\omega),
\]

then the following problem admits a solution \( \varphi \in H(\text{curl}; \omega) \) and \( a\varphi \in H(\text{div}; \omega) \)

\[(3.14) \quad \text{curl}(\varphi) = f, \quad \text{div}(a\varphi) = h \quad \text{in} \omega, \quad \varphi \cdot n = g.
\]

**Proof.** Since \( \omega \) is assumed to be simply-connected, the result is given by Theorem 1.2 in [66]. \( \Box \)

Next, we present the complex formed by the new globally defined spaces of \((3.2)\).

**Lemma 3.3.** The following sequence is a complex:

\[(3.15) \quad \mathbb{R} \overset{\rightarrow}{\Rightarrow} H^2(\beta; T_h) \overset{\text{grad}}{\rightarrow} H^1(\text{curl}; \alpha, \beta; T_h) \overset{\text{curl}}{\rightarrow} H^1(\text{div}; \alpha; T_h) \overset{\text{div}}{\rightarrow} H^1(\partial \Omega) \rightarrow 0.
\]

When \( \Omega \) is a simply-connected convex polyhedron, and the interface is also sufficiently smooth not intersecting \( \partial \Omega \), it is also exact.
Proof. We first verify that \( \nabla H^2(\beta; T_h) \subseteq \ker(\text{curl}) \cap H^1(\text{curl}, \alpha, \beta; T_h) \). This is true due to the jump conditions associated with \( \nabla H^2(\beta; T_h) \) and \( \text{curl} \nabla H^2(\beta; T_h) = 0 \). Similarly, \( \text{curl} H^1(\text{curl}, \alpha, \beta; T_h) \subseteq \ker(\text{div}) \cap H(\text{div}, \alpha; T_h) \) due to the jump condition associated with \( H^1(\text{curl}, \alpha, \beta; T_h) \), and \( \text{div} \text{curl} H^1(\text{curl}, \alpha, \beta; T_h) = 0 \). Finally, it is trivial that \( \text{div} H^1(\text{div}, \alpha; T_h) \subseteq H^1(T_h) \). These results together finish the proof.

We then verify the exactness. We first show \( \nabla H^2(\beta; T_h) = \ker(\text{curl}) \cap H^1(\text{curl}, \alpha, \beta; T_h) \). Given each \( u \in \ker(\text{curl}) \cap H^1(\text{curl}, \alpha, \beta; T_h) \), by the classic exact sequence, there exists \( u \in H^1(\Omega) \) such that \( \nabla u = u \), and by the jump conditions associated with \( H^1(\text{curl}, \alpha, \beta; T_h) \) we have \( u \) also satisfies those of \( H^2(\beta; T_h) \). In addition, on each element \( K \), \( \nabla u = u \in H^1(\cup K^\pm) \) implies \( u \in H^2(\cup K^\pm) \). Therefore, \( u \in H^2(\beta; T_h) \).

We then verify \( \text{curl} H^1(\text{curl}, \alpha, \beta; T_h) = \ker(\text{div}) \cap H^1(\text{div}, \alpha; T_h) \). Given a \( u \in \ker(\text{div}) \cap H^1(\text{div}, \alpha; T_h) \), we consider a function \( \varphi \in H(\text{div}; \Omega) \) satisfying

\[
\text{curl} \beta^{-1} \varphi = u, \quad \text{div}(\varphi) = 0 \quad \text{in} \ \Omega, \quad \varphi \cdot n = 0 \quad \text{on} \ \partial\Omega. \tag{3.16}
\]

By Lemma 3.1 with \( \omega = \Omega, \gamma = \Gamma \) and \( b = \beta^{-1} \), we have this system being well-defined with \( \beta^{-1} \varphi \in H(\text{curl}; \Omega) \) and \( \varphi \in H(\text{div}; \Omega) \). By the geometric condition of \( \Omega \) and \( \Gamma \), there also holds \( \varphi \in H^1(\cup \Omega^\pm) \). Thus, we obtain \( v := \beta^{-1} \varphi \in H(\text{curl}; \Omega) \cap H^1(\cup \Omega^\pm) \) and \( \beta v \in H(\text{div}; \Omega) \). Furthermore, on each element \( K \), \( u \in H^1(\text{div}; K^\pm) \) implies \( v \in H^1(\text{curl}; K^\pm) \). In addition, \( [a \text{curl} v \times n]_{\Gamma K} = 0 \) is trivial by the property of \( u \).

We next show \( \text{div} H^1(\text{div}, \alpha; T_h) = H^1(T_h) \). Given each \( f \in H^1(T_h) \). We consider a \( \phi \) satisfying

\[
\begin{align*}
\text{div}(\alpha^{-1} \nabla \phi) &= f, & & \text{in} \ \Omega, \tag{3.17a} \\
[\phi]_\Gamma &= 0, & & [\alpha^{-1} \nabla \phi \cdot n]_\Gamma = 0, \tag{3.17b} \\
\phi &= 0 & & \text{on} \ \partial\Omega. \tag{3.17c}
\end{align*}
\]

By the elliptic regularity [7], we have \( \phi \in H^2(\cup \Omega^\pm) \), and let \( w = \alpha^{-1} \nabla \phi \in H^1(\cup \Omega^\pm) \). On each element \( K \), as \( f \in H^1(\cup K^\pm) \), we have \( w \in H^1(\text{div}; K^\pm) \). At last, (3.17b) leads to \( [a \text{curl} w \times n]_{\Gamma K} = 0 \) and \( [w \cdot n]_{\Gamma K} = 0 \).

Remark 3.1. The classic exact sequence with higher smoothness is given by [42, 67]:

\[
\begin{array}{cccc}
\mathbb{R} & \hookrightarrow & H^2(\Omega) & \overset{\text{grad}}{\rightarrow} H^1(\text{curl}; \Omega) & \overset{\text{curl}}{\rightarrow} H^1(\Omega) & \overset{\text{div}}{\rightarrow} L^2(\Omega) & \longrightarrow & 0.
\end{array}
\tag{3.18}
\]

We note that this space can be simply revised to be

\[
\begin{array}{cccc}
\mathbb{R} & \hookrightarrow & H^2(\Omega) & \overset{\text{grad}}{\rightarrow} H^1(\text{curl}; \Omega) & \overset{\text{curl}}{\rightarrow} H^1(\text{div}; \Omega) & \overset{\text{div}}{\rightarrow} H^1(\Omega) & \longrightarrow & 0.
\end{array}
\tag{3.19}
\]

The revision can be understood immediately from \( \ker(\text{div}) \cap H^1(\Omega) = \ker(\text{div}) \cap H^1(\text{div}; \Omega) \). The proposed new sequence is a further generation of (3.19) in which the jump information is incorporated.

4. Immersed Finite Element Spaces. In this section, we present \( H^1, H(\text{curl}) \) and \( H(\text{div}) \) IFE functions. The basis functions are some piecewise polynomials satisfying the jump conditions in certain sense to ensure the local approximation property. Particularly, the \( H^1 \) IFE space has been developed in [57], but this is the first time that \( H(\text{curl}) \) and \( H(\text{div}) \) IFE spaces are systematically developed. Different from all the IFE spaces in literature, the spaces constructed here serve the purpose for approximation under the VEM framework, thus are not limited by the constraint that DoFs need to be imposed on their associated geometric objects. Instead, the DoFs are handled by the IVE spaces discussed in Section 5. To facilitate a simple presentation, we shall focus on the case that elements are only cut by the interface once, i.e., each edge has at most one cutting point, which is a reasonable assumption employed by many works in the literature [46, 48, 57]. In fact, the interface elements may generally satisfy this assumption if the background mesh is sufficiently fine, namely the interface is locally flat enough. But they can be also constructed for more complicated interface element geometry, which we leave to Appendix A for this general case.

Next, we need a linear approximation to the interface portion \( \Gamma_h \), denoted by \( \Gamma_h^K \). For example, in [46] \( \Gamma_h^K \) is constructed as a plane passing through the three cutting points forming a triangle satisfying the maximum angle condition, see the middle plot in Figure 2.1 for an illustration. The following lemma essentially acknowledges this setting. Another widely-used linear approximation approach is to use \( \Gamma_h : \phi_h(x) = 0 \) with \( \phi_h \) being the linearization of the sign-distance function \( \phi \) of \( \Gamma \) on the same mesh. These choices indicate that the interface can be well-resolved by a mesh that is sufficiently fine. 

8
Lemma 4.1. Suppose the mesh is sufficiently fine such that \( h < h_0 \) for a fixed threshold \( h_0 > 0 \), then on each interface element \( K \in \mathcal{T}_h \), there exist constants \( C_T \) independent of the interface location and mesh size \( h_K \) such that for every point \( X \in \Gamma^K \) with its orthogonal projection \( X^\perp \) onto \( \Gamma^K \),

\[
\| X - X^\perp \| \leq C_T h_K^2.
\]

Let \( \alpha_h \) and \( \beta_h \) be the piecewise constant functions whose jumps are now across \( \Gamma^K \) instead of \( \Gamma^K \). But, here we shall postpone the specific parameters \( \alpha_h \) and \( \beta_h \) in the PDEs to a later discussion, and focus on a generic piecewise constant function denoted as \( c_h \) to present the IFE functions. In the following discussion, \( \mathcal{P}_k(K) \) denotes the polynomial space with degree \( k \) on \( K \). Let \( n_K \) be the normal vector to \( \Gamma^K \) that is approximately in the same direction with \( n_\Gamma \).

Here the discontinuous coefficient \( c_h \) can be viewed as a Hodge star operator. This perspective is the key for computing the projection of the proposed IVE spaces, see Section 5.4.

In order to derive explicit formulas for the functions in the spaces (4.2), we further let \( \hat{t}_K^1 \) and \( \hat{t}_K^2 \) be the two orthonormal tangential unit vectors to \( \Gamma^K \), and denote the matrix \( T = [n_K, \hat{t}_K^1, \hat{t}_K^2] \). Then, we define the matrices:

\[
M_f^{c_h} = T_K \begin{bmatrix} 1 & 0 & 0 \\ 0 & \tilde{c} & 0 \\ 0 & 0 & \tilde{c} \end{bmatrix} (T_K)^\top \quad \text{and} \quad M_c^{c_h} = T_K \begin{bmatrix} \tilde{c} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (T_K)^\top,
\]

where \( \tilde{c} = c_h^+ / c_h^- \). Clearly, both \( M_f^{c_h} \) and \( M_c^{c_h} \) are symmetric and positive definite. Thus, the spaces \( C^c_F(c_h; K) \) and \( C^c_F(c_h; K) \) can be rewritten as

\[
C_F^c(c_h; K) = \{ c : c^\pm = c|_{K^\pm}, \ c^- = M_{c_h}^{c_k} c^+, \ c^+ \in [P_0(K^+_h)]^3 \},
\]

\[
C_F^c(c_h; K) = \{ c : c^\pm = c|_{K^\pm}, \ c^- = M_{c_h}^{c_k} c^+, \ c^+ \in [P_0(K^+_h)]^3 \}.
\]

\( C_F^c(c_h; K) \) and \( C^c_F(c_h; K) \) are subspaces of the piecewise constant vector functions (dimension 6). The values at \( K^\pm \) for any vector in these spaces have corresponding jump conditions (1.5) and (1.7) as constraints, respectively. It can be verified that the dimension of both \( C_F^c(c_h; K) \) and \( C^c_F(c_h; K) \) is 3.

Now, we proceed to present the \( H^1, \ H(\text{curl}) \) and \( \ H(\text{div}) \) IFE functions. We consider \( C_F^c(a_h; K) \) and \( C_F^c(b_h; K) \), formed by two general positive piecewise constant functions \( a_h \) and \( b_h \). Then, all the \( H^1, \ H(\text{curl}) \) and \( \ H(\text{div}) \) IFE functions with the general parameters \( a_h \) and \( b_h \) have simple formulas presented in Table 1 where \( x_K \) is any point on \( \Gamma^K \). One can directly verify that they belong to the corresponding Sobolev spaces and satisfy the associated jump conditions in the table. Note that the normal jump condition in
the \( \mathbf{H}(\text{curl}) \) case and the tangential jump condition in the \( \mathbf{H}(\text{div}) \) case only hold at the point \( x_K \) instead of the entire \( \Gamma_h^K \). This does not violate the necessary continuities for these two spaces to be in \( \mathbf{H}(\text{curl}; K) \) and \( \mathbf{H}(\text{div}; K) \). Of course, different choices of \( x_K \) lead to different spaces, which is characterized by Lemma A.2. In addition, compared with the standard Lagrange, Nédélec, and Raviart-Thomas elements, the only difference for their IFE counterparts is to replace the constant vectors in \( [P_0(K)]^3 \) by the vectors in \( \mathbf{C}_h^I(a_h; K) \) and \( \mathbf{C}_h^e(b_h; K) \), thus providing the necessary piecewise constant approximation to \( \beta \nabla u, \alpha \text{curl } u, \) and \( \beta u \) on interface elements, respectively.

| IFE spaces        | \( S_h^n(b_h; K) \) | \( S_h^e(a_h, b_h; K) \) | \( \mathbf{S}_h^f(a_h; K) \) |
|-------------------|----------------------|--------------------------|-----------------------------|
| Dimension         | 4                    | 6                        | 4                           |
| Sobolev spaces    | \( H^1(K) \)         | \( \mathbf{H}(\text{curl}; K) \) | \( \mathbf{H}(\text{div}; K) \) |
| Function format   | \( b \cdot (x - x_K) + c \) \( a \times (x - x_K) + b \) \( c(x - x_K) + a \) \( c \in P_0(K) \), \( b \in \mathbf{C}_h^I(a_h; K) \), \( a \in \mathbf{C}_h^e(b_h; K) \), \( c \in \mathbf{C}_h^e(a_h; K) \) |
| Jump conditions   | \( [v_h]_{\Gamma_K^h} = 0 \) \( [\mathbf{v}_h \times \mathbf{n}]_{\Gamma_K^h} = 0 \) \( [\mathbf{v}_h]_{\Gamma_K^h} = 0 \) |
|                   | \( [b_h \nabla v_h \cdot \mathbf{n}]_{\Gamma_K^h} = 0 \) \( [a_h \text{curl } \mathbf{v}_h \times \mathbf{n}]_{\Gamma_K^h} = 0 \) \( [a_h \mathbf{v}_h \times \mathbf{n} x_K] = 0 \) |

Table 1: IFE spaces, their dimensions, their function format, the corresponding jump conditions and the Sobolev spaces to which they belong, where \( x_K \) is any point at \( \Gamma_h^K \).

In addition, on each interface element, these spaces admit a local exact sequence established in the following lemma.

**Lemma 4.2.** The following sequence is a complex and is exact:

\[
\begin{align*}
\mathbb{R} & \longrightarrow S_h^n(b_h; K) \xrightarrow{\text{grad}} S_h^e(a_h, b_h; K) \xrightarrow{\text{curl}} \mathbf{S}_h^f(a_h; K) \xrightarrow{\text{div}} P_0(K) \longrightarrow 0.
\end{align*}
\]

Furthermore, the constant vector spaces \( \mathbf{C}_h^I(a_h; K) \) and \( \mathbf{C}_h^e(b_h; K) \), respectively, are the curl-free and div-free subspaces of \( \mathbf{S}_h^e(a_h, b_h; K) \) and \( S_h^f(a_h; K) \):

\[
\begin{align*}
\text{(4.7a)} & \quad \mathbf{C}_h^I(b_h; K) = \text{grad } S_h^n(b_h; K) = S_h^e(a_h, b_h; K) \cap \ker(\text{curl}), \\
\text{(4.7b)} & \quad \mathbf{C}_h^f(a_h; K) = \text{curl } S_h^n(a_h, b_h; K) = S_h^f(a_h; K) \cap \ker(\text{div}).
\end{align*}
\]

**Proof.** It can be verified directly. \( \square \)

**Remark 4.1.** In computation of Maxwell interface problem, the IVE functions and their curls are projected to the constant spaces \( \mathbf{C}_h^I(b_h; K) \) and \( \mathbf{C}_h^e(\alpha_h; K) \). To ensure an optimal first order convergence, the projections need not be to the full IFE spaces \( S_h^n(\beta_h; K) \), \( S_h^e(\alpha_h, \beta_h; K) \) and \( \mathbf{S}_h^f(\beta_h; K) \). But these spaces will be useful in the computation procedure of the projections.

5. **Immersed Virtual Element Spaces.** It is generally not possible to construct conforming piecewise polynomial spaces to the Sobolev spaces in (3.2). Traditionally, Lagrange, Nédélec and Raviart-Thomas elements are conforming to \( H^1 \), \( \mathbf{H}(\text{curl}) \) and \( \mathbf{H}(\text{div}) \) spaces, yet they cannot provide sufficient approximation when a mesh-cutting interface is present. The IFE spaces introduced above can capture the jump information, but at the cost of losing the conformity. In this section, we construct immersed virtual element (IVE) spaces based on solutions to local interface problems. IVE spaces can be both conforming and satisfy interface conditions perfectly.
Given an interface element $K \in \mathcal{T}_h$, we let $\mathcal{N}_K$ and $\mathcal{E}_K$ be the collection of all the nodes and edges in this triangulation. Note that the nodes include the vertices in the background mesh and cutting points, and the edges include all the sub-edges cut by the interface and newly-added edges. Then $\mathcal{F}_K$ denotes the resulting triangular faces. Given a $T$ which may be a cube, square, tetrahedron or triangle, we let $\mathcal{N}_D_0(T)$ and $\mathcal{R}_T_0(T)$ be the first family of Nédélec polynomial space and the Raviart-Thomas polynomial space of the lowest degree on $T$. The Lagrange space is simply the first-degree polynomial space $\mathcal{P}_1(T)$.

Next, we also need two weighted projections onto the piecewise constant vector spaces $\mathcal{C}_h^e(c_h; K)$ and $\mathcal{C}_h^e(c_h; K)$ which will be used in the definition of the IVE spaces as well as the computation:

\begin{equation}
\Pi_{K}^{e, c_h} : \mathbf{H}(\text{curl}; K) \to \mathcal{C}_h^e(c_h; K), \quad \text{satisfying}
\int_K c_h \Pi_{K}^{e, c_h} v_h \cdot \mathbf{p}_h \, dx = \int_K c_h v_h \cdot \mathbf{p}_h \, dx, \quad \forall \mathbf{p}_h \in \mathcal{C}_h^e(c_h; K),
\end{equation}

\begin{equation}
\Pi_{K}^{f, c_h} : \mathbf{H}(\text{div}; K) \to \mathcal{C}_h^f(c_h; K), \quad \text{satisfying}
\int_K c_h \Pi_{K}^{f, c_h} v_h \cdot \mathbf{p}_h \, dx = \int_K c_h v_h \cdot \mathbf{p}_h \, dx, \quad \forall \mathbf{p}_h \in \mathcal{C}_h^f(c_h; K).
\end{equation}

The super-scripts, $e$ and $f$, still emphasize the distinct Sobolev spaces, i.e., the images of $\Pi_{K}^{e, c_h}$ and $\Pi_{K}^{f, c_h}$ belong to $\mathbf{H}(\text{curl}; K)$ and $\mathbf{H}(\text{div}; K)$, respectively. In computation, only $\Pi_{K}^{e, c_h}$ and $\Pi_{K}^{f, c_h}$ are needed, while some other weights will be also used (only) for definition of the IVE spaces.

5.1. The $H^1$ IVE Space. We first consider the $H^1$ case. Given the boundary triangulation, we define the boundary function space:

\begin{equation}
\mathcal{B}_h^0(\partial K) = \{ v_h \in C(\partial K) : v_h|_T \in \mathcal{P}_1(T), \forall T \in \mathcal{F}_K \}.
\end{equation}

Then, on an interface element $K$, the $H^1$ IVE spaces involving the discontinuous coefficient $\beta$ is defined as

\begin{equation}
\mathcal{V}_h^n(K) = \{ v_h : \nabla \cdot (\beta \nabla v_h) = 0, \; |v_h|_{1,K} = 0, \; [\beta \nabla v_h \cdot \mathbf{n}]_{\Gamma_K} = 0, \; v_h|_{\partial K} \in \mathcal{B}_h^0(\partial K) \},
\end{equation}

Clearly $\mathcal{V}_h^n(K) \subseteq H^1(K)$. On the boundary $\partial K$ we use the continuous $\mathcal{P}_1$ finite element space on the body-fitted surface triangulation. In the interior we use $\beta$-harmonic extension so that the shape functions satisfy the jump condition on the interface.

The property of the nodal DoFs is given by the following Lemma.

Lemma 5.1. The space $\mathcal{V}_h^n(K)$ has nodal DoFs $\{ v_h(z), z \in \mathcal{N}_K \}$.

Proof. First, $v_h \in \mathcal{V}_h^n(K)$ is uniquely determined by the boundary condition in $\mathcal{B}_h^0(\partial K)$. The space $\mathcal{B}_h^0(\partial K)$ has the nodal DoFs associated with the nodes in $\mathcal{N}_K$. So functions in $\mathcal{V}_h^n(K)$ are uniquely determined by their nodal values.

On any non-interface element, the standard Lagrange FE space, i.e., $\mathcal{P}_1(K)$, is used. Thus, with the nodal DoFs, we are able to define the global $H^1$-conforming IVE space as

\begin{equation}
\mathcal{V}_h^n = \{ v_h \in H^0_0(\Omega) : v_h|_K \in \mathcal{V}_h^n(K), \forall K \in \mathcal{T}_h^i, \quad \text{and} \quad v_h|_K \in \mathcal{P}_1(K), \forall K \in \mathcal{T}_h^e \}.
\end{equation}

Functions in $\mathcal{V}_h^n$ is piecewise linear on the element boundary triangulation and in general non-polynomial inside interface element, which is the key to capture both the jump condition and conformity.

As standard VEMs, the function values in the interior are not needed, projections to certain spaces with approximation properties are computed instead. In the following paragraph, we show how to compute $\Pi_{K}^{e, \beta_h} \nabla v_h$ for $v_h \in \mathcal{V}_h^n(K)$. For every $\mathbf{p}_h \in \mathcal{C}_h^e(\beta_h; K)$, by (4.3) there holds $\beta_h \mathbf{p}_h \in \mathcal{C}_h^e(\beta_h^{-1}; K) \subseteq \mathbf{H}(\text{div}; K) \cap \ker(\text{div})$. Then, the integration by parts shows

\begin{equation}
\int_K \beta_h \Pi_{K}^{e, \beta_h} \nabla v_h \cdot \mathbf{p}_h \, dx = \int_K \nabla v_h \cdot (\beta_h \mathbf{p}_h) \, dx = \int_{\partial K} v_h (\beta_h \mathbf{p}_h \cdot \mathbf{n}) \, ds,
\end{equation}

of which the right-hand side is computable. The $L^2$ projection of $v_h$ to $\mathcal{P}_1(K)$ is not computable by the definition of the current space. However, this is not needed as $v_h$ itself can be approximated by the formula in Table 1 using the gradient obtained from (5.5). Denote this weighted $H^1$ projection by $\bar{v}_h$, the constant $c$ in Table 1 can be chosen such that $\int_{\partial K} \bar{v}_h = \int_{\partial K} v_h$. This constraint gives compactness, thus a sufficient approximation for computing the right-hand side term to guarantee the first order optimal convergence.
5.2. The H(curl) IVE Space. For the H(curl) case, the boundary space is defined as
\begin{equation}
B_h^0(\partial K) = \{ v_h : v_h|_T \in \mathcal{N}D_0(T), \forall T \in F_K, (v_h \cdot t)|_e \text{ is continuous on each } e \in E_K \}.
\end{equation}

Each \( \mathcal{N}D_0(T) \) contains the 2D vector polynomials tangentially defined on the planar surface \( T \). By formulations from trace finite element on triangulated surfaces \cite{63}, \( B_h^0(\partial K) \) is a well-defined finite element space on its own, and has the DoFs of \( \int_{\partial K} v_h \cdot t \, ds, e \in E_K \). With this boundary space, we first introduce an auxiliary H(curl) IVE space:
\begin{equation}
\tilde{V}_h^e(K) = \{ v_h \in \mathbf{H}(\text{curl}; K) : \beta v_h \in \mathbf{H}(\text{div}; K), \text{ div}(\beta v_h) = 0, \alpha \text{ curl } v_h \in \mathbf{H}(\text{curl}; K), \text{ curl } \alpha \text{ curl } v_h \in C_h^1(\beta^{-1}; K), v_h^e \in B_h^0(\partial K) \}.
\end{equation}

Here, \( \alpha \) and \( \beta \) can be also understood as Hodge star operators exactly mimicking the second and the third vertical mappings in the desired diagram \cite{33} between different Sobolev spaces.

The following lemma gives the well-posedness and DoFs of \( \tilde{V}_h^e(K) \).

**Lemma 5.2.** \( \tilde{V}_h^e(K) \) is unisolvent with respect to the DoFs \( \{ \int_{\partial K} v_h \cdot t \, ds, e \in E_K \} \) and \( \{ \int_K \beta_h v_h \cdot p_h \, dx, p_h \in C_h^1(\beta; K) \} \).

**Proof.** Let \( q_h \in C_h^1(\beta^{-1}; K) \) and \( r_h \in B_h^0(\partial K) \) be some arbitrary data functions. Let us formulate the following local interface problem arising from the definition in \( \tilde{V}_h^e(K) \):
\begin{equation}
\begin{align*}
\text{curl } \alpha \text{ curl } v_h &= q_h, & \text{div}(\beta v_h) &= 0, & \text{in } K, & v_h^e &= r_h & \text{on } \partial K.
\end{align*}
\end{equation}

Note that the solutions \( v_h \) of \( \text{(5.8)} \) form the space \( \tilde{V}_h^e(K) \). We need to show the well-posedness of this problem; namely, with the given data functions, it admits a unique solution \( v_h \) satisfying \( v_h \in \mathbf{H}(\text{curl}; K), \beta v_h \in \mathbf{H}(\text{div}; K) \) and \( \alpha \text{ curl } v_h \in \mathbf{H}(\text{curl}; K) \). Consider a potential \( \psi_h \) such that
\begin{equation}
\begin{align*}
\text{curl } \alpha \psi_h &= q_h, & \text{div}(\psi_h) &= 0, & \text{in } K, & \psi_h \cdot n &= \text{rot}_{\partial K} r_h, & \text{on } \partial K.
\end{align*}
\end{equation}

Note that
\begin{equation}
\int_{\partial K} \text{rot}_{\partial K} r_h \, ds = \sum_{F \in F_K} \int_{\partial F} r_h \cdot t \, ds = 0,
\end{equation}
by the tangential continuity along each edge that is the intersection of two faces in \( F_K \), as well as \( \partial K \) being a closed surface. As \( q_h \) is div-free, by Lemma 3.1, we have that \( \psi_h \) uniquely exists, i.e., \( \text{(5.9)} \) is well-defined. Then, it is easy to see that \( \text{(5.8)} \) can be equivalently written as
\begin{equation}
\text{curl } v_h = \psi_h, & \text{div}(\beta v_h) = 0 & \text{in } K, & v_h^e &= r_h & \text{on } \partial K.
\end{equation}

Note that the boundary condition in \( \text{(5.10)} \) is equivalent to \( v_h \times n = r_h \times n \). As \( n \times (r_h \times n) = r_h \), with integration by parts on each \( F \in F_K \), we have for any \( v \in H^1(\omega) \),
\begin{equation}
\begin{align*}
\int_{\partial K} (n \times (r_h \times n))(\nabla v \times n) \, ds &= \int_{\partial K} r_h(\nabla v \times n) \, ds = \int_{\partial K} (\text{rot}_{\partial K} r_h)v \, ds,
\end{align*}
\end{equation}
and then the boundary condition in \( \text{(5.9)} \) shows that the compatibility condition in \( \text{(3.13)} \) indeed holds. Thus, the well-posedness follows from Lemma 3.2. The well-posedness implies that the dimension of the solution space is equal to the (finite) dimension of the space of possible data functions; the boundary data function \( r_h \) is uniquely determined by \( \{ \int_{\partial K} r_h \cdot t \, ds, e \in E_K \} \); the right hand side \( q_h \in C_h^1(\beta^{-1}; K) \) which has dimension 3. Therefore, the dimension of solution space matches the DoFs count \( 3 + |E_K| \) for this virtual space.

In the rest of the proof, it needs to be established that the given moments on edges and element interior are indeed DoFs. To this end, it suffices to show that a function with vanishing DoFs is trivial in this space. Noticing \( \text{curl } \alpha \text{ curl } v_h \in C_h^1(\beta^{-1}; K) = \beta_h C_h^1(\beta; K) \) by \( \text{(4.3)} \), thus from the interior DoFs in Lemma 5.2 we have
\begin{equation}
0 = \int_{\partial K} v_h \cdot \text{curl } \alpha \text{ curl } v_h \, ds = \int_{\partial K} \alpha \text{ curl } v_h \cdot \text{ curl } v_h \, ds - \int_{\partial K} \tilde{v}_h^e \cdot (\alpha \text{ curl } v_h \times n) \, ds,
\end{equation}
where we have used integration by parts in the second equality. As the edge moments are zero, we know \( \tilde{v}_h^e = 0 \). So, we have \( \int_K \alpha \text{ curl } v_h \cdot \text{ curl } v_h \, ds = 0 \) which implies \( \text{curl } v_h = 0 \) as \( \alpha \) is positive. In addition, by \( \text{div}(\beta v_h) = 0 \) and the vanishing trace, we derive from Lemma 3.2 that \( v_h = 0 \). □
Similar to the classical VEM, both \( \mathbf{v}_h \) and \( \text{curl} \mathbf{v}_h \) are not computable. But we shall see that their weighted projections to the IFE spaces are computable. We first address the projection of \( \text{curl} \mathbf{v}_h \) which will be then used to develop a new IVE space as a subspace of \( \widetilde{V}_h^\alpha \) that only has the edge DoFs.

By the Hodge star property (4.3), we argue that \( \Pi_K^{\alpha_h} \text{curl} \mathbf{v}_h \) defined in (5.1b) is computable for any positive \( \alpha_h \). In particular, for each \( p_h \in C^1_h(a_h; K) \) we know \( a_h p_h \in H(\text{curl}; K) \cap \ker(\text{curl}) \). Then, applying the projection \( \Pi_K^{\alpha_h} \) to \( \text{curl} \mathbf{v}_h \) with the integration by parts, we obtain

\[
\int_K a_h \Pi_K^{\alpha_h} \text{curl} \mathbf{v}_h \cdot p_h \, dx = \int_K \text{curl} \mathbf{v}_h \cdot (a_h p_h) \, dx = \int_{\partial K} (\mathbf{v}_h \times \mathbf{n}) (a_h p_h)^\tau \, ds.
\]

The right-hand side above is only computable through only the edge DoFs \( \int_{\partial K} \mathbf{v}_h \cdot \mathbf{t} \, ds \), \( e \in \mathcal{E}_K \), since the boundary triangulation is known. In computation, \( \Pi_K^{\alpha_h} \text{curl} \mathbf{v}_h \) provides a sufficient approximation order on elements intersecting the interface.

Now, we let \( S_h^\alpha(a_h; 0; K) \) be the subspace of \( S_h^\alpha(a_h, b_h; K) \) that has the fixed \( b = 0 \) in Table 1. Namely, the functions in \( S_h^\alpha(a_h; 0; K) \) are just \( w_h = a \times (x - x_K) \), \( a \in C^1_h(a_h; K) \), and thus the space only has the dimension 3. Then, based on the auxiliary space \( \widetilde{V}_h^\alpha(K) \), we introduce its subspace:

\[
V_h^\beta(K) = \{ v_h \in \widetilde{V}_h^\alpha(K) : \int_K \text{curl} v_h \cdot w_h \, dx = 0, \forall w_h \in S_h^\beta(\beta_h^{-1}; 0; K) \}.
\]

Clearly, \( V_h^\beta(K) \subset \widetilde{V}_h^\alpha(K) \), and the following lemma ensures that only the edge DoFs are needed for the unisolvency of this subspace due to the extra constraint.

**Lemma 5.3.** \( V_h^\beta(K) \) is unisolvent with respect to the edge DoFs \( \{ \int_e \mathbf{v}_h \cdot \mathbf{t} \, ds, e \in \mathcal{E}_K \} \).

**Proof.** It suffices to show that this extra condition (5.14), in fact, makes the interior DoFs \( \int_K \beta_h \mathbf{v}_h \cdot p_h \, dx \), \( p_h \in C^1_h(\beta_h; K) \) fixed thus not degrees of freedom anymore. To see this, for each \( p_h \in C^1_h(\beta_h; K) \), we have \( \beta_h p_h \in C^1_h(\beta_h^{-1}; K) \). Therefore, by the local exact sequence and (4.3), there exists \( w_h \in S_h^\beta(\beta_h^{-1}; 0; K) \) such that \( \text{curl} w_h = \beta_h p_h \). Then, the integration by parts shows

\[
\int_K \beta_h \mathbf{v}_h \cdot p_h \, dx = \int_K \mathbf{v}_h \cdot \text{curl} w_h \, dx = \int_K \text{curl} v_h \cdot w_h \, dx - \int_{\partial K} v_h^\tau \cdot (w_h \times \mathbf{n}) \, ds.
\]

Note that (I) = 0 by the extra condition in the definition. For (II), \( V_h^\beta(\partial K) \) is also solely determined by the edge DoFs. Therefore, under the extra constraint, the right-hand side of (5.15) is computable for every pair of \( p_h \) and \( w_h \), as long as the edge DoFs \( \{ \int_e \mathbf{v}_h \cdot \mathbf{t} \, ds, e \in \mathcal{E}_K \} \), are given. \( \square \)

For the new space \( V_h^\beta(K) \), the identity (5.15) also gives a simple formula for computing \( \Pi_K^{\beta_h} \mathbf{v}_h \):

\[
\int_K \beta_h \Pi_K^{\beta_h} \mathbf{v}_h \cdot p_h \, dx = \int_K \beta_h \mathbf{v}_h \cdot p_h \, dx = - \int_{\partial K} v_h^\tau \cdot (w_h \times \mathbf{n}) \, ds,
\]

with \( w_h = (\beta_h p_h/2) \times (x - x_K) \) given by Table 1.

With the edge DoFs, we are able to define a global \( H(\text{curl}) \)-conforming space that uses (5.26) on interface elements and standard Nédélec elements on non-interface elements:

\[
V_h = \{ \mathbf{v}_h \in H(\text{curl}) : \mathbf{v}_h|_K \in V_h(K), \ K \in T_h^i \text{ and } \mathbf{v}_h|_K \in N\mathcal{D}_0(K), \ K \in T_h^o \}.
\]

### 5.3. The \( H(\text{div}) \) IVE Space.

For the \( H(\text{div}) \) case, the boundary function space is defined as

\[
B_h^l(\partial K) = \{ v_h : v_h|_F \in \mathcal{P}_0(F), \forall F \in \mathcal{F}_K \}.
\]

Now, similar to the \( H(\text{curl}) \) space, let us define the IVE space in the following:

\[
V_h^l(K) = \{ \mathbf{v}_h \in H(\text{div}; K) : \text{div}(\mathbf{v}_h) \in \mathcal{P}_0(K), \ \text{div}(\mathbf{v}_h \cdot \mathbf{n}) \in B_h^l(\partial K), \ \alpha \mathbf{v}_h \in H(\text{curl}; K), \ \text{curl}(\alpha \mathbf{v}_h) = 0 \}.
\]

Again, the discontinuous coefficient \( \alpha \) serves as a Hodge star operator that is to mimic the third vertical mapping in Diagram 3.3. The well-posedness and DoFs of this space is given by the following lemma.
**Lemma 5.4.** $V_h^f(K)$ is unisolvent with respect to the DoFs $\{v_h|_{F} \cdot n_F, F \in F_K\}$.

**Proof.** We let $c_h \in P_0(K)$ and $r_h \in B_h^p(\partial K)$ be some arbitrary data functions for the definition (5.19) satisfying the compatibility condition

\[(5.20) \quad \int_K c_h \, dx = \int_{\partial K} r_h \, ds.\]

Then, we consider the following div-curl system:

\[(5.21) \quad \text{div}(v_h) = c_h, \quad \text{curl} \alpha v_h = 0 \quad \text{in} \ K, \quad \text{and} \quad v_h \cdot n = r_h \quad \text{on} \ \partial K \]

whose solutions $v_h$ form the space $V_h^f(K)$. Due to the compatibility condition (5.20), by Lemma 3.1 the system in (5.21) is well-posed and admits a unique solution $v_h \in H(\text{div}; K)$ with $\alpha v_h \in H(\text{curl}; K)$. The dimension of $V_h^f(K)$ is just the dimension of space of the independent data functions, $|F_K|$, where we note that $c_h$ does not count as it is determined by $r_h$ from (5.20).

Next, to show that the given moments are indeed DoFs, we suppose they all vanish, and immediately conclude $\text{div}(v_h) = 0$. Then, Lemma 3.1 shows $v_h = 0$. \qed

Now, we discuss the computability of the proposed $H(\text{div})$ IVE space. The identity (5.20), in fact, yields a formula for computing $\text{div}(v_h)$:

\[(5.22) \quad \text{div}(v_h) = |K|^{-1} \int_{\partial K} v_h \cdot n \, ds \]

which is computable. As for $v_h$ itself, we then argue that $\Pi^{f,a_h} v_h$ defined in (5.1b) is always computable for any positive $a_h$. Similar to (5.13), given each $p_h \in C_h^f(a_h; K)$, we have $a_h p_h \in C_h^c(a_h^{-1}; K)$. Then, there exists $\psi_h = (a_h p_h) \cdot (x - x_K) \in S_h^0(a_h^{-1}; K)$ such that $\nabla \psi_h = a_h p_h$, and thus we can derive

\[(5.23) \quad \int_K a_h \Pi^{f,a_h} v_h \cdot p_h \, dx = \int_K a_h v_h \cdot p_h \, dx = \int_K v_h \cdot \nabla \psi_h \, dx = - \int_K \text{div}(v_h) \psi_h \, dx + \int_{\partial K} v_h \cdot n \psi_h \, ds \]

which is computable. In computation, $a_h$ in this space is set to $\alpha_h$ to provide sufficient approximation to $\text{curl} u$.

Thanks to the face DoFs, we can define a global $H(\text{div})$-conforming space:

\[(5.24) \quad V_h^f = \{v_h \in H(\text{div}; \Omega) : v_h \in V_h^f(K), \ K \in T_h^x \text{ and } v_h|_K \in \mathcal{R} T_0(K), \ K \in T_h^n \}. \]

### 5.4. Some Comments and Alternative Definitions

The proposed $H(\text{curl})$ and $H(\text{div})$ IVE spaces above are exactly the extension of the classical virtual spaces in literature [11, 12, 13, 17, 15] to the case of discontinuous coefficients. The modification also includes the source terms and the boundary conditions of the local interface problems, by which the weighted projections are computable.

Next, let us summarize the relation between the involved spaces and weights in the computation of projections, which may be unified as

\[(5.25) \quad \int_K c_h \Pi^{s,c_h} v_h \cdot p_h \, dx = \int_K v_h \cdot \left( \sum_{c_h p_h} \right) \, dx, \quad \text{where} \quad \left\{ \begin{array}{l} (s, s', c_h) = (e, f, \beta_h), \text{ if } (v_h, p_h) \in H(\text{curl}; K) \times C_h^c(c_h; K), \\ (s, s', c_h) = (f, e, \alpha_h), \text{ if } (v_h, p_h) \in H(\text{div}; K) \times C_h^f(c_h; K). \end{array} \right. \]

By the language of differential forms, in order for the wedge product of a $k$-form and $l$-form to be scalar, there needs $k + l = 3$ in the 3D case. Note that $v_h$ and $p_h$ both belong to the $k$-form, $k = 1, 2$, so $c_h$ acts as a Hodge star operator (4.3) mapping $p_h$ to the $(3 - k)$-form for the desired wedge product. Here, the value of $c_h = \alpha_h$ or $\beta_h$ depends on the $H(\text{curl})$ or $H(\text{div})$ spaces matching the underlying Maxwell’s equations.

At last, we provide an alternative definition of the $H(\text{curl})$ IVE spaces being a different subspace of $\tilde{V}_h^f(K)$, which has some nice mathematical properties. The key is also to impose suitable conditions to assign the interior DoFs.
For the $\mathbf{H}(\text{curl})$ space, we may consider

$$
V_h^\beta(K) = \{ v_h \in \widetilde{V}_h^\beta(K) : \int_K \text{curl} v_h \cdot w_h \, dx = \int_K \Pi_K^{\beta} \text{curl} v_h \cdot w_h \, dx, \forall w_h \in S_h^\beta(\beta^{-1}_h, 0; K) \}.
$$

Then, the interior DoFs can be determined also through integration by parts:

$$
\int_{\partial K} \beta_h v_h \cdot p_h \, ds = \int_K \text{curl} v_h \cdot w_h \, dx - \int_{\partial K} \tilde{v}_h \cdot (w_h \times n) \, ds
$$

where $w_h = (\beta_h p_h/2) \times (x - x_K)$ from Table 1 makes the space $V_h^\beta(K)$ only have the edge DoFs. The identity (5.27) also gives the formula for computing $\Pi_K^{\beta} v_h$. But, compared with (5.16), (5.27) needs to compute the extra term $\int_K \Pi_K^{\beta} \text{curl} v_h \cdot w_h \, dx$, which is slightly more expensive.

This approach to determine the subspaces is similar to the one in [3, 22] for the classical $H^1$ virtual spaces. Here, the benefit is to have the new spaces free of the choice of $x_K$. Note that the spaces in (5.14) depends on the choice of the point $x_K \in \Gamma_h^K$ which can be arbitrary on the plane $\Gamma_h^K$ with a distance $O(h_K)$ to the element $K$. However, the new space in (5.26) is invariant with respect to the various $x_K \in \Gamma_h^K$, even though the underlying IFE spaces $S_h^\beta(\beta^{-1}_h, 0; K)$ are not.

**Lemma 5.5.** On each interface element $K$, the IVE spaces $V_h^\beta(K)$ in (5.14) is invariant with respect to the choice of the points $\{x_{K,m}\}_{m=1}^{M-1}$ at $\Gamma_h^{K,m}$.

**Proof.** The result directly follows from Lemma A.2. \qed

6. **A discrete de Rham Complex.** The proposed IVE spaces inherit the de Rham complex properties of standard finite element spaces including the exact sequence and commutativity.

Thanks to the nodal, edge and face DoFs of the proposed IVE spaces, let us first define the corresponding interpolations:

(6.1a) $I_h^n : H^1(\beta; \mathcal{T}_h) \rightarrow V_h^n$ satisfying $I_h^n u(x) = u(x), \forall x \in \mathcal{N}_h$.

(6.1b) $I_h^\alpha : H^1(\text{curl}, \alpha, \beta; \mathcal{T}_h) \rightarrow V_h^\beta$ satisfying $\int_{\mathcal{E}_e} I_h^\alpha u \cdot t \, ds = \int_{\mathcal{E}_e} u \cdot t \, ds, \forall e \in \mathcal{E}_h$.

(6.1c) $I_h^f : H^1(\text{div}, \alpha; \mathcal{T}_h) \rightarrow V_h^f$ satisfying $\int_{\mathcal{F}_f} I_h^f u \cdot n \, ds = \int_{\mathcal{F}_f} u \cdot n \, ds, \forall F \in \mathcal{F}_h$.

We further need the standard $L^2$ projection denoted by $\Pi_h^0 : L^2(K) \rightarrow \mathcal{P}_0(K)$, and define the global one as $\Pi_h^0$ such that $\Pi_h^0|_T = \Pi_T^0$, i.e., $\Pi_h^0 : L^2(\Omega) \rightarrow Q_h$ where

$$
Q_h = \{ v_h : v_h \in \mathcal{P}_0(K), \forall K \in \mathcal{T}_h(K) \}.
$$

These operators together with the IVE spaces will be used to formulate the continuous and discrete de Rham complex in (1.9). In fact, Lemma 3.3 already shows that exactness in the continuous level. So our focus will be on the discrete one in the lower part of (1.9).

**Lemma 6.1.** When $\Omega$ is topologically trivial, the following complex is exact

$$
\mathbb{R} \xrightarrow{\text{grad}} V_h^n \xrightarrow{\text{curl}} V_h^\beta \xrightarrow{\text{div}} Q_h \xrightarrow{\text{grad}} 0.
$$

**Proof.** The argument for showing the sequence being a complex is basically the same as Lemma 3.3. To show the exactness, we can look at the DoFs which form a co-chain exact complex on the cell-complex defined by the mesh. For the completeness, we include a detailed proof below.

First verify $\nabla V_h^n = \ker(\text{curl}) \cap V_h^n$. By the classic exact sequence, given each $v_h \in \ker(\text{curl}) \cap V_h^n$, there exists $v_h \in H^1(\Omega)$ such that $\nabla v_h = v_h$. Given an interface element $K$, the jump conditions associated with $V_h^\beta(K)$ implies $v_h$ also satisfies those of $V_h^n(K)$. In addition, let $F$ be one of its face in the boundary triangulation, since $\text{rot}_F v_h = 0$ and $v_h|_{F} \in ND_0(F)$, we have $\nabla_F v_h = v_h|_F \in [\mathcal{P}_0(F)]^2$, which implies
$v_h \in \mathcal{P}_1(F)$. Hence, $v_h \in V^a_h(K)$. On each non-interface element $K$, $v_h$ is just a constant vector, so $v_h \in \mathcal{P}_1(K)$. Therefore, we conclude $v_h \in V^a_h$.

Second, we now, to the end, prove $\text{div}(\nabla_h^f) = Q_h$. Given each $q \in Q_h$, there exists a regular potential $\mathbf{u} \in H^1(\Omega)$ s.t. $\text{div} \mathbf{u} = q$. Then, we define $u_h = I_h^f \mathbf{u}$ and

$$|K| \frac{q}{|K|} = \int_K \text{div}(\mathbf{u}) \, dx = \int_{\partial K} \mathbf{u} \cdot n \, ds = \int_{\partial K} I_h^f \mathbf{u} \cdot n \, ds = \int_K \text{div}(I_h^f \mathbf{u}) \, dx$$

which implies $\text{div}(I_h^f \mathbf{u})|_K = q|_K$ on each element $K$ finishing the proof.

To verify $\text{curl} \nabla_h^f = \nabla_h^f \cap \ker(\text{div})$, we can use a dimension count. Denote by $\#V_h, \#E_h, \#F_h, \#T_h$ the number of vertices, edges, faces, and elements, respectively. From the surjectivity $\text{div}(\nabla_h^f) = Q_h$, we know $\dim(\nabla_h^f \cap \ker(\text{div})) = \dim V_h^f - \dim Q_h = \#F_h - \#T_h$. On the other hand, $\dim \text{curl} V_h^e = \dim V_h^e - \dim (\ker(\text{curl}) \cap V_h^e) = \dim V_h^e - \dim (\text{dim curl} V_h^e) = \#E_h - \#V_h + 1$. Then by Euler's formula, we get $\dim \text{curl} V_h^f = \dim (\nabla_h^f \cap \ker(\text{div}))$. As $\text{curl} V_h^f \subseteq \nabla_h^f \cap \ker(\text{div})$, we conclude that they are equal. \hfill \Box

7. The Immersed Virtual Schemes. Based on the previously established spaces and projections, in this section we are ready to present the IVE schemes. With the exact sequence, we also develop fast solvers for the $H(\text{curl})$ interface problem. At last, we present a data structure that can facilitate an efficient and vectorized implementation of the proposed method.

7.1. The IVE Schemes. We shall focus on the $H^1$ and $H(\text{curl})$ interface problems due to their vast applications. For simplicity, we let $(\cdot, \cdot)_D$ be the standard $L^2$ inner product on $D$. For the $H^1$ case, we define a local bilinear form as

$$b_K(u_h, v_h) = (\beta_h \Pi^\varepsilon \nabla u_h, \Pi^\beta_h \nabla v_h)_K + S_K((I - \Pi^\varepsilon \nabla) \nabla u_h, (I - \Pi^\beta_h) \nabla v_h),$$

where the projection $\Pi^\varepsilon \nabla \beta_h$ is given in (5.1a) on interface elements $K$ and simply assumed to be the identity operator on non-interface elements as the standard FE spaces are used and computable.

The choice of the stabilization term $S_K$ is one of the keys for the proposed method working satisfactorily. Various choices have been proposed in the literature [14, 22, 26] based on different norms on the boundary. In this work, we will employ the following surface $H^1$ stabilization:

$$S_K(w_h, z_h) := \gamma h_K \sum_{F \in F_K} (w_h^T, z_h^T)|_F,$$

where $w_h|_F$ and $z_h|_F$ are the tangential components on the face $F$. In particular, we note that $(\nabla v_h)|_F = \nabla v_h - (\nabla v_h \cdot n_F) n_F = \nabla_F v_h$ is the surface gradient of $v_h$, and it is computable since the trace of $v_h$ on $\partial K$ belongs to the standard FE space and is known. As the standard FE spaces are defined on the boundary triangulation, the stabilization in (7.2) must be piecewise computed.

Then, the global bilinear form is defined as

$$b_h(u_h, v_h) = \sum_{K \in T_h} b_K(u_h, v_h).$$

The proposed IVE scheme is to find $u_h \in V^a_h$ such that

$$b_h(u_h, v_h) = \sum_{K \in T_h} \int_K f \tilde{v}_h \, dx, \quad \forall v_h \in V^a_h.$$  

Note that the projection of $v_h$ itself is not computable for the current space, and thus we simply employ the approximated gradient $\Pi^\varepsilon \nabla \Pi^\beta h$ and the formula in Table 1 to form $\tilde{v}_h = (\Pi^\varepsilon \nabla \Pi^\beta h) \cdot (x - x_K) + c$ with the constant $c$ chosen such that $\int_{\partial K} \tilde{v}_h \, ds = \int_{\partial K} v_h \, ds$. In fact, the $H^1$ interface problem is not our focus, and this discretized system needs not to be solved. Only the stiffness matrix for the $H^1$ interface problem is needed for the auxiliary space preconditioner in the fast solver for the $H(\text{curl})$ interface problems.

As for the $H(\text{curl})$ interface problem, we need to deal with the terms of curl curl $\mathbf{u}$ and $\mathbf{u}$ separately. For the curl curl term, we introduce

$$a_K^1(u_h, v_h) = (\alpha_h \Pi^\varepsilon \nabla \Pi^\beta_h \text{curl} u_h, \Pi^\varepsilon \nabla \Pi^\beta_h \text{curl} v_h)_K + S^1_K((I - \Pi^\varepsilon \nabla) \text{curl} u_h, (I - \Pi^\varepsilon \nabla) \text{curl} v_h),$$

where $S^1_K$ is a stabilization term defined in a similar way to $S_K$. For the curl term, we introduce

$$a_K^2(u_h, v_h) = (\alpha_h \Pi^\varepsilon \nabla \Pi^\beta_h \text{curl} u_h, \Pi^\varepsilon \nabla \Pi^\beta_h \cdot n) \cdot v_h)_K + S^2_K((I - \Pi^\varepsilon \nabla) \cdot \text{curl} u_h, (I - \Pi^\varepsilon \nabla) \cdot \text{curl} v_h),$$

where $S^2_K$ is another stabilization term defined in a similar way to $S_K$. These two terms are combined into a single bilinear form $a_K(u_h, v_h)$, which is defined as

$$a_K(u_h, v_h) = a_K^1(u_h, v_h) + a_K^2(u_h, v_h).$$
where, similarly, \( \Pi^{f,\alpha}_{K} \) is chosen as (5.1b) on interface elements but just the identity operator on non-interface elements. The stabilization term is defined as

\[
S^{1}_{K}(w_{h}, z_{h}) = \gamma_{1}h \sum_{F \in \mathcal{F}_{K}} (w_{h} \cdot n_{F}, z_{h} \cdot n_{F})_{F},
\]

where we note that \( \text{curl} u_{h} \cdot n \) for any function \( u_{h} \in V_{h}^{e} \) can be computed through the formula in (3.6) with \( \text{rot} F u_{h} = |F|^{-1} \int_{\partial F} u_{h} \cdot t \, ds \) on each triangular face \( F \). In addition, the bilinear form for the weighted \( L^{2} \) inner product is defined as

\[
a^{0}_{K}(u_{h}, v_{h}) = (\beta_{h} \Pi^{e,\beta}_{K} u_{h} \cdot \Pi^{e,\beta}_{K} v_{h})_{K} + S^{0}_{K}((I - \Pi^{e,\beta}_{K})u_{h}, (I - \Pi^{e,\beta}_{K})v_{h}),
\]

where \( \Pi^{e,\beta}_{K} \) is defined in (5.16) on interface element and the identity on non-interface elements. The stabilization is given by

\[
S^{0}_{K}(w_{h}, z_{h}) = \gamma_{0} \sum_{F \in \mathcal{F}_{K}} (w^{T}_{h}, z^{T}_{h})_{F},
\]

where \( w^{T}_{h} \) and \( z^{T}_{h} \) still denote the tangential components onto each face \( F \). With the triangulation on faces, \( w^{T}_{h} \in V_{h}^{e} \) is computable through the edge DoFs. We highlight that the scaling \( h^{0} = 1 \) in the stabilization \( S^{0}_{K}(w_{h}, z_{h}) \) is different from the usual \( h \) in classical VEM in [15, 12, 13, 16], and this is also the key for the proposed method to produce optimal convergent solutions. The theoretical justification has been given for the 2D case in [28]. Then, we can define the global bilinear form

\[
a_{h}(u_{h}, v_{h}) = \sum_{K \in T_{h}} a^{1}_{K}(u_{h}, v_{h}) + a^{0}_{K}(u_{h}, v_{h}).
\]

The proposed IVE scheme for the \( \textbf{H} \)(curl) interface problem is to find \( u_{h} \in V_{h}^{e} \) such that

\[
a_{h}(u_{h}, v_{h}) = \sum_{K \in T_{h}} \int_{K} \mathbf{f} \cdot \Pi^{e,\beta}_{K} v_{h} \, dx, \quad \forall v_{h} \in V_{h}^{e}.
\]

**7.2. Implementation.** Inherited from the classical VEM, the implementation of the proposed algorithm is highly vectorized. Computing the projections from IVE spaces and assembly of matrices significantly outperform the classical IFE methods. To see this, following [32], we describe a \texttt{face2elem} and a \texttt{face} data structure that can greatly facilitate the implementation. \texttt{face2elem} is a vector mapping from each (local) face’s index to its mother element’s index. \texttt{face} is a matrix containing each face’s DoFs (node or edge) on its rows. Here, we use the tetrahedral interface elements in Figure 7.1 to illustrate the data structures. In Figure 7.1, the red and blue segments are, respectively, cutting edges by the interface and newly added edges for the surface triangulation. The indices are shown on the two plots for all the vertices and edges. Suppose the index of this element is 1, and then the desired data structures of \texttt{face2elem} and \texttt{face} are shown in (7.11).

**Fig. 7.1:** Indices of nodes (left) and edges (right) of an interface element.
Here $\Pi_{K}^{H}$ denotes the auxiliary space preconditioner (HX preconditioner) based on the auxiliary space framework in \cite{50}. The second complex, the first one is the algebraic multigrid (AMG) solver $B^{\text{curl}}$ for a scalar Laplacian matrix, which consists of the following three components:

- a smoother $S^{\text{curl}}$ of the H(curl) matrix $A$,
- an algebraic multigrid (AMG) solver $B^{\text{grad}}$ for a scalar Laplacian matrix,
• an AMG solver $B^{\text{grad}}$ for a vector Laplacian matrix.

We simply employ the incidence matrix associated with the operator $\nabla : \overline{S}_H^\alpha \rightarrow \overline{S}_h^\alpha$ as the discrete gradient matrix $G$ which resembles that from the lowest order Nédéléc element on simplicial meshes. $G$ maps the nodal DoFs (columns) to edge DoFs (rows). There are two nonzero entry, $\pm 1$, on each row. The columns of these entries correspond to the nodes of the edge. The sign is determined by the global orientation of an edge. The node-to-edge transfer matrix is denoted by $\Pi : \prod_{i=1}^{3} \mathbb{R}^{\text{node}} \rightarrow \mathbb{R}^{\text{edge}}$. Note that these two matrices being well-defined are based on the node and edge DoFs of the $H^1$ and $H(\text{curl})$ IVE spaces.

For $H(\text{curl})$ problems, it is known that a multigrid solver for Poisson-type equations is not sufficient since the discrete operator corresponding to $\text{curl}(\alpha \text{curl}) + \beta I$ behaves differently for a gradient field and a solenoidal field (see e.g., [6]). When sufficient piecewise regularity is assumed, we have by [39]

$$\|\alpha \text{grad} u\|^2 \approx \|\alpha \text{curl} u\|^2 + \|\alpha \text{div} u\|^2.$$ 

Hence, if $u = \text{curl} w \in (\ker(\text{curl}))^\perp$, for some suitable $w$ such that $\text{div} u = 0$, then

$$(\alpha \text{curl} u, \text{curl} u) + (\beta u, u) \approx (\alpha \text{grad} u, \text{grad} u) + (\beta u, u),$$

which corresponds to the following operator:

(7.15) \quad $B^{\text{grad}} u := -\text{div}(\alpha \nabla u) + \beta u$

that will be assembled as an auxiliary matrix and can be solved by an AMG solver for the vector $H^1$-interface problem. On the other hand, if $u, v \in \ker(\text{curl})$, i.e., $u = \nabla p$ and $v = \nabla q$, for some suitable $p, q$, then

$$(\alpha \text{curl} u, \text{curl} v) + (\beta u, v) = (\beta \nabla p, \nabla q),$$

thus we can formulate the matrix problem for the gradient part of the solution by $B^{\text{grad}} = G^T AG$, which corresponds to the following operator:

(7.16) \quad $B^{\text{grad}} u := -\text{div}(\beta \nabla p)$

that can be again solved efficiently by an AMG solver for the $H^1$-interface problem.

Next, we present a block diagonal smoother (preconditioner). A block matrix is formed by the edge DoFs in the neighborhood expanding from the interface. We begin with the collection of the DoFs that is near the interface:

(7.17) \quad $\mathcal{D}_1 = \{ e \in \mathcal{E}_h : \exists K \in T_h^l \text{ such that } e \in \mathcal{E}_K \}$.

Then, starting from $\mathcal{D}_1$ we iteratively define

(7.18) \quad $\mathcal{D}_l = \{ e \in \mathcal{E}_h : e \text{ has at least one node belonging to the edges of } \mathcal{D}_{l-1} \}$.

Let $A_l$ be the matrix of the entries in $A$ associated with the DoFs in $\mathcal{D}_l$. Then, we rewrite (7.13) into

(7.19) \quad $A_l u := \begin{pmatrix} A_N & A_{NI} \\ A_{IN} & A_I \end{pmatrix} \begin{pmatrix} u_N \\ u_I \end{pmatrix} = \begin{pmatrix} f_N \\ f_I \end{pmatrix}$

Here, the key is to solve the $A_l$ block by a direct solver, which is indeed the price to be paid by the proposed method. However, since the size of $A_l$, i.e., $\#\mathcal{D}_l$, is in the order of $O(\#\text{total DoF}^{2/3})$ for reasonably small $l$, this direct solver is generally efficient. The expanding width can reduce the number of iterations required for the resulting solver, see Table 3 for the comparison. Meanwhile, our numerical experience suggests that the increased cost is negligible for small $l$’s, e.g., the expanding width $l = 1 \text{ or } 2$ is enough. It is almost equivalent to directly solving a 2D linear system which can be efficiently handled by “backslash” (mldivide) in Matlab. Furthermore, since the direct solver will be called multiple times, we opt to store the $LU$-factorization of $A_l$ in the inner iteration (preconditioning) to be more efficient. The residual equation of the $A_N$ block can be efficiently solved using a block or point-wise Gauss-Seidel smoother. At last, we summarize the algorithm in the following for a fixed $l$, and denote $A_{l} := A_l$. 

19
Algorithm 7.1 An HX preconditioned CG

Require: $u^{(0)}$, $\mathbf{tol}$, $M$, $l$, block form of $A$.
Ensure: $u^{\text{MG}}$.

1: $k = 0$.
2: $r_I \leftarrow f_I - A_I u^{(0)}$.
3: while True do
4: $e_I \leftarrow \text{LUSolve}(A_I, r_I)$.
5: $r_N \leftarrow \text{Smoothen}(A_N, r_N)$.
6: $r \leftarrow [r_N, r_I]$.
7: $r \leftarrow r + \Pi(AuxSolve(A^{aux}, \Pi^T r))$.
8: $r_c \leftarrow G(AuxSolve(A, G^T r))$.
9: $r \leftarrow r + r_c$.
10: $u^{\text{MG}} \leftarrow \text{CG}(A, r)$.
11: if $k > M$ or norm($r$) < $\mathbf{tol}$ then
12: Break.
13: end if.
14: $k \leftarrow k + 1$.
15: end while.

Fig. 8.1: Plots of interface and triangulation: sphere (left) and two twisted torus (right). For the spheric interface, the boundary triangulation of interface element is plotted in blue on the left half of the sphere.

8. Numerical Examples. In this section, we present a class of numerical examples to validate the aforementioned advantages of the proposed method. The background unfitted mesh is generated by cutting $\Omega$ into $N^3$ cubes and each cube is then cut into several tetrahedra with the mesh size be $h = 1/N$.

8.1. The $H^1$-interface problem. We first consider the $H^1$-interface problem given by (1.2) for a spheric interface shown in Figure 8.1 on the domain $\Omega = (-1,1)^3$. The exact solution is constructed as

$$
(8.1) \quad u(x) = \begin{cases} 
\exp((\|x\|^2 - r^2)/\beta^-), & \text{if } x \in \Omega^-; \\
\sin((\|x\|^2 - r^2)/\beta^+) + 1, & \text{if } x \in \Omega^+,
\end{cases}
$$

where the source term of (1.2) as well as the boundary conditions are computed accordingly. The numerical experiment is carried on the meshes of $N = 10, 20, 30, ..., 160$. We first report the CPU time to compute the projections of IVE functions and the related matrix assembling in Table 2 for the meshes $N = 60$ to 160. Note that at the finest level, there are approximately 153600 interface elements, and based on the proposed data structure, computing the IVE projections is highly efficient. As for the matrix assembling, we observe that the majority of time is devoted to the stabilization term, and we believe this is due to a larger number of triangular faces from the boundary triangulation. Certainly, these computation are highly parallelizable. In addition, we show the numerical errors for $\beta^- = 1$, $\beta^+ = 10$ and $\beta^- = 1$, $\beta^+ = 100$ in the left two plots of Figure 8.2. Due to the geometric errors caused by coarse meshes, the convergence orders indicated on the graph are computed by incorporating only the errors from $N = 60$ to 160, but it clearly shows the
asymptotic optimal convergence. Remarkably, the optimal convergence is even achieved for the $L^\infty$ norm which is a demanding property for interface problems.

| Total # DoFs | 1367631 | 1771561 | 2248091 | 2803221 | 3442951 | 4173281 |
|-------------|---------|---------|---------|---------|---------|---------|
| Interface # DoFs | 160926 | 191322 | 224682 | 259818 | 298866 | 330774 |
| Time(s) for projection | 2.96 | 4.00 | 4.88 | 37.92 | 5.65 | 7.13 |
| Time(s) for matrix assembling | 14.21 | 18.14 | 21.14 | 30.47 | 37.58 | 42.61 |

Table 2: CPU time for computing the projections of IFE functions and the generation of stiffness matrices including the stabilization terms.

Fig. 8.2: Numerical errors and convergence order for the $H^1$ interface problem. Left two plots are for the spheric interface: $(\beta^-, \beta^+) = (1, 10)$ and $(\beta^-, \beta^+) = (1, 100)$, and right two plots are for the toroidal interface: $(\beta^-, \beta^+) = (1, 10)$ and $(\beta^-, \beta^+) = (1, 100)$. The black dashed lines indicate the expected $O(h^2)$ convergence for the $L^\infty$ and $L^2$ errors and $O(h)$ for the $H^1$ errors.

The second example concerns a more complicated interface shape that has two torus twisted with each other, see the right plot in Figure 8.1. The level-set functions of the two torus are $\phi_1 = (((x_1+0.3)^2+x_2^2)^{1/2} - 0.2)^2 + x_3^2 - (\pi/5)^2$ and $\phi_2 = (((x_1-0.3)^2+x_2^2)^{1/2} - 0.2)^2 + x_3^2 - (\pi/5)^2$, and then the level-set function of this interface is given by $\phi(x) = \min(\phi_1(x), \phi_2(x))$. The domain inside the two torus is $\Omega^- = \{x : \phi(x) < 0\}$ and the outside one is $\Omega^+ = \{x : \phi(x) > 0\}$. The exact solution is given by

$$u(x) = \begin{cases} 
1, & \text{if } x \in \Omega^-, \\
\cos(\phi_1(x)\phi_2(x)), & \text{if } x \in \Omega^+. 
\end{cases}$$

In this case, the computational domain is $\Omega = (-1.3, 1.3)^3$. The numerical solutions and errors are reported in the right two plots of Figure 8.2. As the interface has much larger curvature which requires the finer mesh to resolve. The convergence orders are estimated from the mesh size $N = 60$ to 160 which indicate the optimal convergence even for the $L^\infty$ errors.

**8.2. The $\bf{H}(\text{curl})$ interface problem.** Now, let us consider the $\bf{H}(\text{curl})$ interface problem. It is known that solving the linear system from the Maxwell’s equations is much more challenging. So here we first test the fast solver developed in Section 7.3 for an extreme case that each interface element has small-cut subelements. For this purpose, we consider the domain $\Omega = (-1,1)^3$ with a flat interface $x_1 = 5 \times 10^{-2}r$. Fix the mesh size as $N = 20$ and the parameters as $(\alpha^-, \alpha^+) = (1, 10)$ and $(\beta^-, \beta^+) = (1, 10)$. If $r = 0$, i.e., $x_1 = 0.05$, the interface plane cuts all the interface elements exactly through the center and thus, each subelement has regular shape. In computation, we let $r = 1, 2, 3, 4$, i.e., the subelements on the left-side of the interface will become small accordingly. We report the condition numbers, the number of iterations and CPU time in Table 3. We can see that small-cut interface elements can indeed make the conditioning worse, which may significantly increase the iteration numbers, see the results for $l = 0$. However, the effect of small-cut interface elements can be successfully eliminated by the proposed block diagonal smoother $A_l$ in (7.19). For this extreme case, $l = 2$ seems sufficient to make the convergence completely independent of small subelements, but our numerical experience suggests that $l = 1$ is good enough in general.
Table 3: Condition numbers of the $H(\text{curl})$ linear system with various interface location, and the related CPU time and # iterations for the expanding width $l = 0, 1, 2$, where $l = 0$ means no block matrix used.

| Interface location $r$ | Condition numbers $6.4 \times 10^6$ | $1.7 \times 10^7$ | $1.7 \times 10^9$ | $1.7 \times 10^{11}$ | $1.7 \times 10^{13}$ |
|------------------------|---------------------------------------|------------------|------------------|------------------|------------------|
| $l = 0$                | # iteration                           | 44               | 53               | 107              | 327              | 842              |
|                        | Time(s)                               | 12               | 16               | 27               | 84               | 220              |
| $l = 1$                | # iteration                           | 43               | 44               | 43               | 73               | 91               |
|                        | Time(s)                               | 11               | 11               | 11               | 21               | 24               |
| $l = 2$                | # iteration                           | 43               | 44               | 43               | 43               | 42               |
|                        | Time(s)                               | 11               | 12               | 12               | 11               | 12               |

Next, we consider the spheric interface and slightly modify the benchmark example from [47] of which the analytical solution is given by

\[
\mathbf{u} = \begin{cases} 
\frac{1}{\beta} \mathbf{x} + \frac{1}{\alpha} n_1 R_1(\mathbf{x})[(x_2-x_3)\mathbf{e}_1, (x_3-x_1)\mathbf{e}_2, (x_1-x_2)\mathbf{e}_3]^\top & \text{in } \Omega^-, \\
\frac{1}{\beta} \mathbf{x} + \frac{1}{\alpha} n_2 R_2(\mathbf{x})[(x_2-x_3)\mathbf{e}_1, (x_3-x_1)\mathbf{e}_2, (x_1-x_2)\mathbf{e}_3]^\top & \text{in } \Omega^+,
\end{cases}
\]

where $\mathbf{x} = [x_1, x_2, x_3]^\top$ and $R_1(\mathbf{x}) = r_1^2 - ||\mathbf{x}||^2$, $R_2(\mathbf{x}) = r_2^2 - ||\mathbf{x}||^2$. The numerical experiment is carried on the meshes of $N = 10, 20, 30, ..., 80$. In particular, the computational time and number of iterations are presented in Table 4. From the table, we can conclude that both the block-diagonal smoother and the HX preconditioner are important for reducing the iteration number for convergence. The results also show that the direct solver at each iteration does not cost significant computational time compared with the total cost of the iterative solver. Next, we report the numerical errors in both the $H^1$ and $H(\text{curl})$ norms in the left two plots of Figure 4, and the estimated convergence orders are also indicated in the plots which clearly demonstrate the optimality.

Table 4: CPU time and number of iterations for solving the $H(\text{curl})$ linear system with the spheric interface and $(\alpha^-, \beta^-) = (1, 1)$ and $(\alpha^+, \beta^+) = (100, 200)$: block-diagonal HX (BD-HX) with $l = 0$ and 1 and the simple block-diagonal PCG (BD-PCG). The CPU time with respect to DoFs are approximatly $O((\text{#DoF})^{1.19})$, $O((\text{#DoF})^{0.80})$, $O((\text{#DoF})^{0.82})$ for BD-PCG, BD-HX($l = 0$) and BD-HX($l = 1$), respectively.

| BD-PCG | Total # DoFs | 80554 | 244424 | 547074 | 1028452 | 1734626 | 2703384 | 3980338 |
|--------|--------------|-------|--------|--------|---------|---------|---------|---------|
|        | # iteration  | 962   | 1465   | 1932   | 2385    | 2828    | 3238    | 3661    |
|        | Time(s)      | 42.72 | 166.46 | 427.70 | 905.39  | 1693.95 | 2871.83 | 4542.63 |
| BD-HX  | $l = 0$      |       |        |        |         |         |         |         |
|        | # iteration  | 144   | 142    | 146    | 140     | 148     | 142     | 145     |
|        | Time(s)      | 38.89 | 69.09  | 117.97 | 214.89  | 401.81  | 533.28  | 808.19  |
| BD-HX  | $l = 1$      |       |        |        |         |         |         |         |
|        | # iteration  | 75    | 76     | 81     | 77      | 80      | 83      | 90      |
|        | Time(s)      | 22.16 | 41.14  | 70.12  | 132.54  | 241.93  | 321.93  | 529.66  |

In the second example, we also consider the twisted torus in the right plot of Figure 8.1 on the domain $\Omega = (-1.3, 1.3)^3$. To construct a function that satisfies the corresponding jump condition on the torus surface, we let $f(\mathbf{x}) = \phi_1(\mathbf{x})\phi_2(\mathbf{x})((x_1+0.3)^2 + x_2^2)((x_1-0.3)^2 + x_2^2)$. Then, the exact solution is then defined as

\[
\mathbf{u} = \frac{1}{\beta} \nabla f(\mathbf{x}) + \frac{1}{\alpha} \cos(f(\mathbf{x}))\mathbf{v}_0, \quad \text{with } \mathbf{v}_0 = [0, 0, 1]^\top,
\]

where the boundary conditions and the source term are computed accordingly. The numerical results are reported in the right two plots of Figure 8.3 which also shows the clear optimal convergence rate. These results demonstrate that the IVE method works well for complex surfaces.

At last, we note that the penalty-type method cannot achieve optimal convergence numerically including both the IFE method [47] and the interface-penalty method [30, 29]. Therefore, we believe the present method has distinguishing advantages in computational electromagnetism.
A. IFE Spaces on Complicated Geometry. Here, we describe the IFE spaces for complicated interface element geometry, i.e., $\Gamma^K$ may have multiple components. Let us assume that $\Gamma^K$ consists of the multiple components $\Gamma^{K,m}$, $m = 1, 2, ..., M - 1$, each of which is a simply-connected smooth surface. As $\Gamma$ is supposed not to intersect itself, $\Gamma^{K,m}$’s then do not intersect with one another. Without loss of generality, we assume the subelement containing $A_1$ is $K_1$ and $\Gamma^{K,1} = \partial K_1 \setminus \partial K$. Then, $K_m$ is the subelement bounded by $\partial K$, $\Gamma^{K,m}$ and $\Gamma^{K,m+1}$, $m = 2, ..., M - 1$, and the remaining one is denoted by $K_M$. We present a 2D illustration of the geometry by the right plot in Figure 2.1. The parameters associated with the subelement $K_m$ are denoted as $\alpha_m$ and $\beta_m$, which should take the values of $\alpha^-$ and $\alpha^+$ alternatively.

For each $\Gamma^{K,m}$, we let $\Gamma^{K,m}$ be its planar approximation. Similarly, define the subelement containing $A_1$ by $K_{h,1}$, and the others $K_{h,2}...,K_{h,M}$ are defined in a similar manner as their counterparts $K_1,...,K_M$. Note that each of $K_{h,1},...,K_{h,M}$ is a polyhedron. Let $\alpha_h$ and $\beta_h$ be the piecewise constant functions defined on these polyhedral subelements, and denote $\alpha_{h,m} = \alpha_h|_{K_{h,m}}$ and $\beta_{h,m} = \beta_h|_{K_{h,m}}$, $m = 1,...,M$.

Similar to (4.5), we are able to derive explicit formulas for the functions in the spaces (4.2). For each linear interface component $\Gamma^K_h$, we further let $\mathbf{t}_m^1$ and $\mathbf{t}_m^2$ be the two orthogonal tangential unit vectors to $\Gamma^{K,m}_h$, and denote the matrix $T_m = [\mathbf{n}_m, \mathbf{t}_m^1, \mathbf{t}_m^2]$. Then define the transformation matrices:

$$M^{e,c_h}_{K,m} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \tilde{c}_m & 0 \\ 0 & 0 & \tilde{c}_m \end{bmatrix} (T_m)^\top$$

and

$$M^{e,c_h}_{K,m} = \begin{bmatrix} \tilde{c}_m & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (T_m)^\top,$$

where $\tilde{c}_m = c_{h,m}/c_{h,m+1}$, with $m = 1, 2, ..., M - 1$. Then, define the spaces $\mathbf{C}^e_h(c_h; K)$ and $\mathbf{C}^f_h(c_h; K)$ as

(A.2a) \[ \mathbf{C}^e_h(a_h; K) = \{ \mathbf{c} : \mathbf{c}_m = \mathbf{c}|_{K_{h,m}} \in \mathcal{P}_0(K_{h,m}), m = 1, ..., M, \mathbf{c}_{m+1} = M^{e,c_h}_{K,m} \mathbf{c}_m, m = 1, ..., M - 1 \}, \]

(A.2b) \[ \mathbf{C}^f_h(b_h; K) = \{ \mathbf{c} : \mathbf{c}_m = \mathbf{c}|_{K_{h,m}} \in \mathcal{P}_0(K_{h,m}), m = 1, ..., M, \mathbf{c}_{m+1} = M^{e,c_h}_{K,m} \mathbf{c}_m, m = 1, ..., M - 1 \}. \]

Again, the constant vectors at different cut regions are related by the jump conditions and thus the dimension of both $\mathbf{C}^e_h(c_h; K)$ and $\mathbf{C}^f_h(c_h; K)$ is also 3. In this case, the formulas of IFE functions are slightly more complicated which are presented in the following lemma.

**Lemma A.1.** Let $\mathbf{x}_{K,m}$ be an arbitrary point at $\Gamma^{K,m}_h$. Let $\mathbf{a} = \mathbf{a}_m$ and $\mathbf{b} = \mathbf{b}_m$ be two arbitrary vectors in $\mathbf{C}^f_h(a_h; K)$ and $\mathbf{C}^f_h(b_h; K)$, and let $c$ be an arbitrary constant. Then, the formulas for the functions in $\mathbf{S}^e_h(a_h, b_h; K)$ and $\mathbf{S}^f_h(a_h, b_h; K)$, respectively, are

(A.3) \[ v^e_h = \begin{cases} \mathbf{b}_m \cdot (\mathbf{x} - \mathbf{x}_{K,1}) + c & \text{in } K_{h,m}, \ m = 1, 2, \\ \mathbf{b}_m \cdot (\mathbf{x} - \mathbf{x}_{K,m-1}) + c + \sum_{l=2}^{m-1} \mathbf{b}_l \cdot (\mathbf{x}_{K,l} - \mathbf{x}_{K,l-1}) & \text{in } K_{h,m}, \ m \geq 3, \end{cases} \]

(A.4) \[ v^f_h = \begin{cases} \mathbf{a}_m \times (\mathbf{x} - \mathbf{x}_{K,1}) + \mathbf{b}_m & \text{in } K_{h,m}, \ m = 1, 2, \\ \mathbf{a}_m \times (\mathbf{x} - \mathbf{x}_{K,m-1}) + \mathbf{b}_m + \xi_m & \text{in } K_{h,m}, \ m \geq 3, \end{cases} \]

with $\xi_m := \sum_{l=2}^{m-1} (M^{e,b_h}_{K,m-1} \cdots M^{e,b_h}_{K,l}) [\mathbf{a}_l \times (\mathbf{x}_{K,l} - \mathbf{x}_{K,l-1})]$, and

(A.5) \[ v^f_h = \begin{cases} c(\mathbf{x} - \mathbf{x}_{K,1}) + \mathbf{a}_m & \text{in } K_{h,m}, \ m = 1, 2, \\ c(\mathbf{x} - \mathbf{x}_{K,m-1}) + \mathbf{a}_m + \eta_m & \text{in } K_{h,m}, \ m \geq 3, \end{cases} \]
with \( \eta_m := c \sum_{l=2}^{m-1} (M_{K,m-1}^{a,b} \cdots M_{K,l}^{a,b}) (x_{K,l} - x_{K,l-1}) = c \sum_{l=2}^{m-1} (\prod_{n=m-1}^{l} M_{K,n}^{a,b}) (x_{K,l} - x_{K,l-1}) \).

The formed IFE spaces also have the dimension 4, 6, and 4 for the \( H^1 \), \( \mathbf{H}(\text{curl}) \) and \( \mathbf{H}(\text{div}) \) cases, respectively.

**Proof.** One can directly verify that these piecewisely-defined functions satisfy the corresponding jump conditions shown in Table 1 but on each \( \Gamma^K \). The dimension can be simply counted by the number of free variables of \( a, b \) and \( c \) in the formulas above.

Note that in Lemma A.1, the points \( \{x_{K,m}\}_{m=1}^{M-1} \) should be chosen and fixed. The following lemma gives the difference between two edge IFE spaces defined with different sets of such points.

**Lemma A.2.** On each interface element \( K \), let \( \{x_{K,m}\}_{m=1}^{M-1} \) and \( \{\hat{x}_{K,m}\}_{m=1}^{M-1} \) be two different sets of points on \( \{\Gamma^K\}_{m=1}^{M-1} \), and let \( \bar{v}_h \) and \( \bar{v}_{h} \) be the corresponding IFE functions in (A.4). Then, there holds

\[
\bar{v}_h - \bar{v}_{\hat{h}} \in C_h^e(b_h; K).
\]

**Proof.** Denote \( \bar{t}_m = \bar{x}_{K,m} - x_{K,m} \) which is parallel to the plane \( \Gamma^K \). We first have

\[
\bar{v}_h - \bar{v}_{\hat{h}} = \begin{cases} \bar{a}_m \times \bar{t}_1, & \text{in } K_{h,m}, \ m = 1, 2, \\
\bar{a}_m \times \bar{t}_{m-1} + \sum_{l=2}^{m-1} (\prod_{n=m-1}^{l} M_{K,n}^{e,b_h}) \bar{a}_l \times (\bar{t}_{l-1} - \bar{t}_l), & \text{in } K_{h,m}, \ m \geq 3. \end{cases}
\]

Now, for each \( \bar{a}_m \times \bar{t}_{m-1} \) with \( m = 2, \ldots, M \), by the property of \( \bar{a} \in A(a_h; K) \), we have

\[
\bar{a}_{m-1} \cdot (\bar{a}_m \times \bar{t}_{m-1}) = \bar{a}_{m-1} \cdot (\bar{a}_m \times \bar{t}_{m-1}) = \bar{n}_{m-1} \cdot (\bar{a}_m \times \bar{t}_{m-1} - \bar{t}_{m-1} - \bar{t}_{m-1}) \bar{a}_m = \bar{n}_{m-1} \cdot (\bar{a}_m \times \bar{t}_{m-1}).
\]

and similarly,

\[
\bar{n}_{m-1} \times (\bar{a}_m \times \bar{t}_{m-1}) = \bar{n}_{m-1} \cdot (\bar{a}_m \times \bar{t}_{m-1}) = \bar{n}_{m-1} \times (\bar{a}_m \times \bar{t}_{m-1}).
\]

(A.8) and (A.9) together imply \( \bar{a}_m \times \bar{t}_{m-1} = M_{K,m-1}^{e,b_h} (\bar{a}_m \times \bar{t}_{m-1}) \). Therefore, for \( m \geq 4 \), we can have

\[
\begin{align*}
(\bar{v}_h - \bar{v}_{\hat{h}})|_{K_{h,m}} &= M_{K,m-1}^{e,b_h} (\bar{a}_m \times \bar{t}_{m-1}) + \sum_{l=2}^{m-1} (\prod_{n=m-1}^{l} M_{K,n}^{e,b_h}) \bar{a}_l \times (\bar{t}_{l-1} - \bar{t}_l) \\
&= M_{K,m-1}^{e,b_h} \left( \bar{a}_m \times \bar{t}_{m-2} + \sum_{l=2}^{m-2} (\prod_{n=m-1}^{l} M_{K,n}^{e,b_h}) \bar{a}_l \times (\bar{t}_{l-1} - \bar{t}_l) \right) \\
&= M_{K,m-1}^{e,b_h} (\bar{v}_h - \bar{v}_{\hat{h}})|_{K_{h,m-1}}.
\end{align*}
\]

With the similar idea, we can verify \( (\bar{v}_h - \bar{v}_{\hat{h}})|_{K_{h,m}} = M_{K,m-1}^{e,b_h} (\bar{v}_h - \bar{v}_{\hat{h}})|_{K_{h,m-1}} \) for \( m = 2, 3 \). \( \square \)

**REFERENCES**

[1] Gabriel Acosta and Ricardo G. Durán. The Maximum Angle Condition for Mixed and Nonconforming Elements: Application to the Stokes Equations. *SIAM J. Numer. Anal.*, 37(1):18–36, 1999.

[2] Slimane Adjerid, Nabil Chaabane, and Tao Lin. An immersed discontinuous finite element method for Stokes interface problems. *Comput. Methods Appl. Mech. Engrg.*, 293:170–190, 2015. in press.

[3] B. Ahmed, A. Alsaedi, F. Brezzi, L.D. Marini, and A. Russo. Equivalent projectors for virtual element methods. *Comput. Math. Appl.*, 66(3):376–391, 2013.

[4] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault. Vector potentials in three-dimensional non-smooth domains. *Math. Meth. Appl. Sci.*, 21, 1998. 6, 7.

[5] D. N. Arnold, R. S. Falk, and R. Winther. Preconditioning in \( H(\text{div}) \) and applications. *Math. Comp.*, 66(219), 1997.

[6] D. N. Arnold, R. S. Falk, and R. Winther. Multigrid in \( H(\text{div}) \) and \( H(\text{curl}) \). *Numerische Mathematik*, 85(2):197–217, 2000.

[7] I. Babuška. The finite element method for elliptic equations with discontinuous coefficients. *Computing (Arch. Elektron. Rechen).* 5:207–213, 1970.

[8] I. Babuška and A. K. Aziz. On the Angle Condition in the Finite Element Method. *SIAM J. Numer. Anal.*, 13(2):214–226, 1976.

[9] I. Babuška, G. Caloz, and J. E. Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM J. Numer. Anal.*, 31(4):945–981, 1994.
