

# ON A GENERAL IMPLEMENTATION OF $h$ - AND $p$ -ADAPTIVE CURL-CONFORMING FINITE ELEMENTS

MARC OLM<sup>†</sup>, SANTIAGO BADIA<sup>†,‡</sup>, AND ALBERTO F. MARTÍN<sup>†,‡</sup>

ABSTRACT. Edge (or Nédélec) finite elements are theoretically sound and widely used by the computational electromagnetics community. However, its implementation, especially for high order methods, is not trivial, since it involves many technicalities that are not properly described in the literature. To fill this gap, we provide a comprehensive description of a general implementation of edge elements of first kind within the scientific software project **FEMPAR**. We cover into detail how to implement arbitrary order (i.e.,  $p$ -adaptive) elements on hexahedral and tetrahedral meshes. First, we set the three classical ingredients of the finite element definition by Ciarlet, both in the reference and the physical space: cell topologies, polynomial spaces and moments. With these ingredients, shape functions are automatically implemented by defining a judiciously chosen polynomial *pre-basis* that spans the local finite element space combined with a change of basis to automatically obtain a canonical basis with respect to the moments at hand. Next, we discuss global finite element spaces putting emphasis on the construction of global shape functions through oriented meshes, appropriate geometrical mappings, and equivalence classes of moments, in order to preserve the inter-element continuity of tangential components of the magnetic field. Finally, we extend the proposed methodology to generate global curl-conforming spaces on *non-conforming* hierarchically refined (i.e.,  $h$ -adaptive) meshes with arbitrary order finite elements. Numerical results include experimental convergence rates to test the proposed implementation.

**Keywords:** edge finite elements, curl-conforming spaces, adaptive mesh refinement, implementation

## 1. INTRODUCTION

Edge elements were originally proposed in the seminal work by Nédélec [1]. They are a natural choice in electromagnetic finite element (FE) simulations due to their sound mathematical structure [2]. In short, edge FE spaces represent curl-conforming fields with continuous tangential components and discontinuous normal components. It is recognized to be their greatest advantage against Lagrangian FEs [3].<sup>1</sup> The curl operator has null

---

*Date:* March 18, 2019.

<sup>†</sup> Centre Internacional de Mètodes Numèrics en Enginyeria, Esteve Terrades 5, E-08860 Castelldefels, Spain. <sup>‡</sup> Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, E-08034 Barcelona, Spain. MO gratefully acknowledges the support received from the Catalan Government through the FI-AGAUR grant. SB gratefully acknowledges the support received from the Catalan Government through the ICREA Acadèmia Research Program. This work has partially been funded by the European Union through the FP7 project FORTISSIMO, under the grant agreement 609029. E-mails: [molm@cimne.upc.edu](mailto:molm@cimne.upc.edu) (MO), [sbadia@cimne.upc.edu](mailto:sbadia@cimne.upc.edu) (SB), [amartin@cimne.upc.edu](mailto:amartin@cimne.upc.edu) (AM).

<sup>1</sup>There are different approaches to design discretizations of the Maxwell equations that rely on Lagrangian FEs and can theoretically converge to singular solutions (see, e.g., [4]). However, these methods are not robust for complex electromagnetic problems. In our experience [5], the usage of edge elements is of imperative importance in the modelling of fields near singularities by allowing normal components to jump across interfaces between two different media with highly contrasting properties.

space, the gradient of any scalar field. To remove its kernel, one needs to include a zero-order term, a transient problem or a divergence-free constraint over the magnetic field times resistivity<sup>2</sup> (also known as Coulomb gauge) [6].

The edge FE method has the status of the method of choice in the computational electromagnetics community. Nevertheless, other discretization techniques are available for electromagnetic problems. Apart from the aforementioned nodal FEs [4], nodal discontinuous Galerkin methods [7, 8] and isogeometric analysis [9] have also been considered for the Maxwell problem. Furthermore, the virtual element method (VEM) [10], which is designed to handle meshes consisting of arbitrary polygonal or polyhedral elements, has also been defined for curl-conforming spaces [11]. Local edge VEM spaces [11] rely on a space formulated on the boundary of the element, thus they naturally allow to design adaptive methods with conforming, arbitrary meshes.

Edge FEs are widespread in the computational electromagnetics community, mainly for the lowest order case. In fact, they receive the name of edge elements because for first order approximations each degree of freedom (DoF) is associated with an edge of the mesh, and DoFs can be understood as circulations along edges. However, its implementation, especially for high order, is not a trivial task and involves many technicalities. On the other hand, there are few works addressing the implementation details of arbitrary-order edge FE schemes.

The construction of local bases for edge elements has been addressed by few authors. A basis for high order methods, expressed in terms of the (barycentric) affine coordinates of the simplex (i.e., tetrahedra) is described in [12, 13], including also implementation details in [14]. The definition of affine-related coordinates is used to define expressions for the bases in tetrahedra, prisms and pyramids in [15]. Furthermore, hierarchical bases of functions can be computed for hexahedra by tensor product of 1D Legendre polynomials and the so-called 1D  $H(\text{curl})$ -shape functions [16, 17], which are in turn obtained by differentiation of the former ones. Nevertheless, we prefer to rely on the FE definition by Ciarlet, which induces the basis of shape functions as the dual basis with respect to suitable edge DoFs [18, 19]. In this work, arbitrary order edge basis functions are automatically implemented by defining polynomial *pre-bases* that span the local finite element spaces combined with a change of basis.

Better documented is the construction of global curl-conforming spaces [20, 21], which relies on the Piola mapping to achieve the global continuity of the tangent components by using moments defined in the reference cell. In the case of unstructured hexahedral meshes, where non-affine geometrical mappings must be used in general, optimal convergence of *standard* edge elements in the  $H(\text{curl})$  norm cannot be achieved [22]. Complex geometries can still be considered using, e.g., unfitted FE techniques [23] on octree background meshes to avoid non-affine mappings.

Special care has to be taken in the definition of edge DoFs to enforce the right continuity across cells. To prevent the so-called sign conflict, different alternatives have been proposed. A simple sign flip [24] can assure consistency, i.e., all local DoFs shared by two or more tetrahedra represent the same global DoF, for first order 3D FEs only. For higher order methods, a remedy can be based on the construction of all possible shape functions combined with a permutation that depends on local edge/face orientations [15, 25]. Our preferred solution, for simplicity and ease of implementation, is to *orient* tetrahedral meshes, requiring that local nodes numbering within every element are always sorted based on their global indices [21]. On the other hand, although there are hexahedral meshes that cannot be oriented in 3D [26], we restrict ourselves to forest-of-octrees meshes [27] such that that the initial coarse mesh can be oriented, e.g., a uniform, structured mesh.

---

<sup>2</sup>Edge FEs provide solutions that are pointwise divergence-free in the element interiors. However, the normal component of the field can freely jump across inter-element faces. The zero inter-element jump constraint has to be enforced by a Lagrange multiplier [4] to eliminate the kernel.

Few open source FE software projects include the implementation of edge FEs of arbitrary order. FEniCS supports arbitrary order FEs [28], but is restricted to tetrahedra and has its own domain specific language for weak forms to automatically generate the corresponding FE code, which may prevent  $hp$ -adaptivity. On the other hand, the deal.ii library also provides arbitrary order edge FEs [29, 30], even on  $h$ -adaptive meshes, but it is restricted to hexahedral meshes. Recently, a C++ plugin that defines high order edge FE ingredients [14] has been developed for the FreeFEM++ library [31], but it is restricted to up to third order and tetrahedral meshes. As far as we know, the most complete approaches in the literature are the implementations in MFEM [32], Netgen/NGSolve [33, 34] and hp2D [16]/hp3D [17]. MFEM provides arbitrary order tetrahedral/hexahedral edge FEs and  $h$ -adaptivity on non-conforming hexahedral meshes. Netgen/NGSolve and hp3D implement arbitrary order edge prism and pyramidal FEs, besides the tetrahedral and hexahedral ones, and hp3D provides the  $hp$ -adaptive method on non-conforming hexahedral meshes.

The usage of non-nodal based (compatible or structure-preserving) FEs, especially in multi-physics applications in combination with incompressible fluid and solid mechanics [35, 36, 37], is still scarce. It is motivated by the fact that most codes in computational mechanics are designed from inception for being used with nodal FEs, and the extension to other types of FEs seems to be a highly demanding task. Although non-nodal based FEs are theoretically well known, the poor, fractioned, and spread information on key implementation issues does not contribute to its greater diffusion in the computational mechanics community.

The motivation of this work is to provide *a comprehensive description of a novel and general implementation of edge FEs of arbitrary order on tetrahedral and hexahedral<sup>3</sup> non-conforming meshes*, confronted with typical hard-coded implementations of the shape functions that preclude high order methods. With this aim, we will address three critical points that are especially complex in the construction of arbitrary order edge FEs: 1) an automatic manner of generating the associated polynomial spaces; 2) a general solution for the so-called sign conflict; and 3) a construction of the discrete  $H(\text{curl})$ -conforming FE spaces atop *non-conforming* meshes. With regard to 1), we generate arbitrary order edge FE shape functions in a simple but extensible manner. First, we propose a *pre-basis* of vector-valued polynomial functions spanning arbitrary order local edge FE spaces for both tetrahedral and hexahedral topologies; see [39] for a related approach on tetrahedra. Next, we compute a change of basis from the *pre-basis* to the canonical basis with respect to the edge FE DoFs, i.e., the basis of shape functions. To address 2) and properly determine the inter-element continuity, we rely on oriented meshes. Finally, in order to address 3), we combine a standard adaptive mesh refinement (AMR) nodal-based implementation of constraints on non-conforming geometrical objects (a.k.a. as hanging) [40] with the relation between a Lagrangian *pre-basis* and the edge basis, to avoid the evaluation of edge moments on the refined cells. Besides, the same machinery can also be applied to other polynomial based FEs, like Raviart-Thomas FEs, Brezzi-Douglas-Marini FEs [41], second kind edge FEs [42], or recent divergence-free FEs [43]. An alternative approach, more suitable for hierarchical bases of functions, is to compute the constraints on non-conforming objects by comparison of the two representations of a function (atop the coarse or the refined cell) at uniformly distributed collocation points [16].

In the course of an  $hp$ -adaptive finite element procedure, an error estimator indicates at which mesh cells the error of the computed FE solution is higher, which will be potential candidates to be refined. For curl-conforming problems, several types of *a posteriori* error

---

<sup>3</sup>Although this work is restricted to hexahedra and tetrahedra, one can also find definitions for prism and pyramid edge elements [38] with optimal rates of convergence of the numerical solution towards the exact solution in the  $H(\mathbf{curl}, \Omega)$  norm.

estimators have been developed and analyzed. Efficient and reliable residual-based error estimators were introduced in [44] and have been further developed and analyzed in [45, 46]. Among others, we also find hierarchical error estimators [47], equilibrated estimators [48] or recovery-based estimators [49].

The proposed approach is the result of the experience gained by the authors after the implementation in **FEMPAR** [50, 51], a scientific software project for the simulation of problems governed by partial differential equations (PDEs). In any case, we present an automatic, comprehensive strategy rather than focusing on a particular software implementation. Consequently, thorough definitions and examples will be provided, but the exact software implementation will not be shown. The reader is directed to [50] for code details, where an exhaustive introduction to the software abstractions of **FEMPAR** is presented.

The text is organized as follows. In Sect. 2, we introduce some notation and a general problem with the curl-curl formulation. In Sect. 3, we will cover the definition and construction of the edge FE of arbitrary order, focusing on an straightforward manner of generating the involved polynomial spaces. Sect. 4 is devoted to the construction of global curl-conforming FE spaces. In Sect. 5, a strategy to deal with  $h$ -adaptive spaces with the edge element will be described. Finally, in Sect. 6, we will show some convergence results, which will validate the implementation of the edge elements.

## 2. PROBLEM STATEMENT

**2.1. Notation.** In this section, we introduce the model problem to be solved and some mathematical notation. Bold characters will be used to describe vector functions or tensors, while regular ones will determine scalar functions or values. No difference is intended by using upper-case or lower-case letters for functions.

Let  $\Omega \subset \mathbb{R}^d$  be a bounded domain with  $d = 2, 3$  the space dimension. In 3D, the rotation operator of a vector field  $\mathbf{v}$  is defined as:

$$\nabla \times \mathbf{v} = \begin{bmatrix} \partial_2 v_3 - \partial_3 v_2 \\ \partial_3 v_1 - \partial_1 v_3 \\ \partial_1 v_2 - \partial_2 v_1 \end{bmatrix}.$$

We use the standard multi-index notation for derivatives, with  $\boldsymbol{\alpha} := (\alpha_1, \dots, \alpha_d)^T \in \mathbb{Z}_+^d$  and  $|\boldsymbol{\alpha}|_1 = \sum_{i=1}^d |\alpha_i|$ . Let us define the spaces

$$\begin{aligned} (1) \quad H^r(\Omega) &:= \{v \in L^2(\Omega) \mid \partial^{\boldsymbol{\alpha}} v \in L^2(\Omega) \text{ for all } |\boldsymbol{\alpha}|_1 \leq r\}, \\ (2) \quad H^r(\mathbf{curl}, \Omega) &:= \{\mathbf{v} \in H^r(\Omega)^d \mid \nabla \times \mathbf{v} \in H^r(\Omega)^d\}. \end{aligned}$$

The space  $H^0(\mathbf{curl}, \Omega)$  is represented with  $H(\mathbf{curl}, \Omega)$ . We also consider the subspaces

$$\begin{aligned} (3) \quad H_0^1(\Omega) &:= \{v \in H^1(\Omega) \mid v = 0 \text{ in } \partial\Omega\}, \\ (4) \quad H_0(\mathbf{curl}, \Omega) &:= \{\mathbf{v} \in H(\mathbf{curl}, \Omega) \mid \mathbf{n} \times \mathbf{v} = \mathbf{0} \text{ in } \partial\Omega\}, \end{aligned}$$

where  $\mathbf{n}$  denotes the outward unit normal to the boundary of the domain  $\Omega$ . The space  $H(\mathbf{curl}, \Omega)$  in (2) is equipped with the norm

$$\|\mathbf{u}\|_{H(\mathbf{curl}, \Omega)} := \left( \|\mathbf{u}\|_{L^2(\Omega)^d}^2 + \|\nabla \times \mathbf{u}\|_{L^2(\Omega)^d}^2 \right)^{\frac{1}{2}}.$$

In 2D, a scalar version of the curl operator can be defined as  $\text{curl}(\mathbf{v}) \doteq \partial_1 v_2 - \partial_2 v_1$ . We can similarly define  $H^r(\text{curl}, \Omega)$  and its related subspaces. In the following, we consider the notation for the 3D case.

**2.2. Formulation.** The model problem consists in finding a vector field  $\mathbf{u}$  (magnetic field) solution of

$$(5) \quad \nabla \times \alpha \nabla \times \mathbf{u} + \beta \mathbf{u} = \mathbf{f} \quad \text{in } \Omega,$$

where  $\mathbf{f}$  is a given source term. Problem parameters  $\alpha, \beta$  can range from scalar, positive values for isotropic materials to positive definite tensors for anisotropic materials. Besides, (5) needs to be supplied with appropriate boundary conditions. The boundary of the domain  $\partial\Omega$  is divided into its Dirichlet boundary part, i.e.,  $\partial\Omega_D$ , and its Neumann boundary part, i.e.,  $\partial\Omega_N$ , such that  $\partial\Omega_D \cup \partial\Omega_N = \partial\Omega$  and  $\partial\Omega_D \cap \partial\Omega_N = \emptyset$ . Then, boundary conditions for the problem at hand read

$$\begin{aligned} \mathbf{u} \times \mathbf{n} &= \mathbf{g}_D && \text{on } \partial\Omega_D, \\ \mathbf{n} \times (\alpha \nabla \times \mathbf{u}) &= \mathbf{g}_N && \text{on } \partial\Omega_N, \end{aligned}$$

where  $\mathbf{n}$  is a unit normal to the boundary. Dirichlet (essential) boundary conditions prescribe the tangent component of the field  $\mathbf{u}$  on the boundary of the domain. On the other hand, Neumann (natural) boundary conditions arise in the integration by parts of the curl-curl term

$$(6) \quad \int_{\Omega} (\nabla \times \alpha \nabla \times \mathbf{u}) \cdot \mathbf{v} = \int_{\Omega} (\alpha \nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) - \int_{\Omega_N} (\alpha \nabla \times \mathbf{u}) \cdot (\mathbf{n} \times \mathbf{v}).$$

Consider now two different non-overlapping regions on the domain  $\Omega$  corresponding to two different media (usual case in electromagnetic simulations), namely  $\Omega_1$  and  $\Omega_2$ , such that  $\Omega_1 \cup \Omega_2 = \Omega$ , and let us define the *interface* as  $\Gamma := \Omega_1 \cap \Omega_2$ . Let us denote by  $\mathbf{n}_1, \mathbf{n}_2$  the unit normal pointing outwards of  $\Omega_1$  and  $\Omega_2$  on  $\Gamma$ , resp. The *transmission* conditions (in absence of other sources) for (5) are stated as follows:

$$\mathbf{n} \times (\alpha_1 \nabla \times \mathbf{u}_1 - \alpha_2 \nabla \times \mathbf{u}_2) = 0 \quad \text{on } \Gamma,$$

where  $\mathbf{n}$  can, e.g., be  $\mathbf{n}_1 = -\mathbf{n}_2$ . For the sake of simplicity in the presentation (not in the implementation), let us consider  $\partial\Omega_D = \partial\Omega$ , so that the Neumann boundary term (6) can be removed from the weak form, and homogeneous Dirichlet boundary conditions. Hence, the variational form of the double curl formulation (5) reads: find  $\mathbf{u} \in H_0(\mathbf{curl}, \Omega)$  such that

$$(7) \quad (\alpha \nabla \times \mathbf{u}, \nabla \times \mathbf{v}) + (\beta \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in H_0(\mathbf{curl}, \Omega).$$

### 3. EDGE FES

Let  $\mathcal{T}_h$  be a quasi-uniform partition of  $\Omega$  into a set of hexahedra (quadrilaterals in 2D) or tetrahedra (triangles in 2D) geometrical cells  $K$ . For every  $K \in \mathcal{T}_h$ , we denote by  $h_K$  its diameter and set the characteristic mesh size as  $h = \max_{K \in \mathcal{T}_h} h_K$ . Let us denote by  $v \in \mathcal{N}$ ,  $e \in \mathcal{E}$  and  $f \in \mathcal{F}$  the components and global sets of vertices, edges and faces of  $\mathcal{T}_h$ , with cardinality  $n_{\mathcal{N}}, n_{\mathcal{E}}$  and  $n_{\mathcal{F}}$ , resp. Using Ciarlet's definition, a FE is represented by the triplet  $\{K, \mathcal{V}, \Sigma\}$ , where  $\mathcal{V}$  is the space of functions on  $K$  and  $\Sigma$  is a set of linear functionals on  $\mathcal{V}$ . The elements of  $\Sigma$  are called DoFs (moments) of the FE. We will denote the number of functionals on the cell as  $n_{\Sigma}$ , and  $\Sigma \doteq \{\sigma_a\}_{a=1}^{n_{\Sigma}}$ .  $\Sigma$  is a basis for  $\mathcal{V}'$ , which is dual to the so-called basis of shape functions  $\{\phi^a\}_{a=1}^{n_{\Sigma}}$  for  $\mathcal{V}$ , i.e.,  $\sigma_a(\phi^b) = \delta_{ab}$ ,  $\forall a, b \in \{1, \dots, n_{\Sigma}\}$ .

At this point, we must distinguish between the *reference* FE, built on a reference cell, and the FE in the physical space. Our implementation of the space of functions and moments is based on a unique reference FE  $\{\hat{K}, \hat{\mathcal{V}}, \hat{\Sigma}\}$ . Then, in the physical space, the FE triplet on a cell  $K$  relies on its reference FE, a geometrical mapping  $\Phi_K$  such that  $K = \Phi(\hat{K})$  and a linear bijective function mapping  $\hat{\Psi}_K : \hat{\mathcal{V}} \rightarrow \mathcal{V}$ . It is well known that quadrilateral FEs may result in a loss of accuracy on general meshes, e.g., when elements are given as images of hexahedra under invertible bilinear maps, in comparison with the

accuracy achieved with squares or cubes [52]. In this work, hexahedra in the physical space are obtained with affine transformations from the reference square or cube, thus optimal convergence properties are guaranteed. Nevertheless, this fact does not restrict the simulations to simple geometries, since unfitted approaches [53] may be employed to model complex geometries with structured background meshes. Let us denote by  $\mathbf{J}_K$  the Jacobian of the geometrical mapping, i.e.,  $\mathbf{J}_K \doteq \frac{\partial \Phi_K}{\partial \mathbf{x}}$ . The functional space is defined as  $\mathcal{V} \doteq \{\hat{\Psi}_K(\hat{v}) \circ \Phi_K^{-1} : \hat{v} \in \hat{\mathcal{V}}\}$ ; we will also use the mapping  $\Psi_K : \hat{\mathcal{V}} \rightarrow \mathcal{V}$  defined as  $\Psi_K(\hat{v}) \doteq \hat{\Psi}_K(\hat{v}) \circ \Phi_K^{-1}$ . Finally, the set of DoFs on the physical FE is defined as  $\Sigma \doteq \{\hat{\sigma} \circ \Psi_K^{-1} : \hat{\sigma} \in \hat{\Sigma}\}$  from the set of the reference FE linear functionals. In the following subsections, we go into detail into these concepts and provide some practical examples.

**3.1. Reference cell.** A *polytope* is mathematically defined as the convex hull of a finite set of points (vertices). The concept of polytope may be of practical importance, because it allows one to develop codes that can be applied to any topology of arbitrary dimension that fits into the framework, see [50] for a thorough exposition. For the sake of ease, we restrict ourselves to two possible polytopes:  $d$ -cubes and  $d$ -simplices (with  $d = 2, 3$ ), defined as the convex hull of a set of  $n_v = 2^d$  and  $n_v = d + 1$  geometrically distinct points, respectively. An *ordered* set of vertices  $\{\mathbf{v}_1, \dots, \mathbf{v}_{n_v}\}$  not only defines the topology of the cell  $\hat{K}$  but an *orientation*. Edges and faces are polytopes of lower dimension, and an ordered set of their vertices also defines their orientation. For the sake of illustration, the local indexing at the cell level is depicted in Figs. 1a and 2a for a 3-cube and a 3-simplex, resp. The 2D cases follow by simply considering the restriction of the 3D cell to one of its faces.

**3.2. Polynomial spaces.** Local FE spaces are usually spanned by polynomial functions. Let us start by defining basic polynomial spaces that will be needed in the forthcoming definitions. We define here all polynomial spaces in an arbitrary polytope  $K$ , but they will only be used for the reference cell in the next sections. The space of polynomials of degree less than or equal to  $k > 0$  in all the variables  $\{x_i\}_{i=1}^d$  is denoted by  $\mathcal{Q}_k(K)$ . Analogously, we can define the space of polynomials of degree less than or equal to  $\{k_i\}_{i=1}^d$  for the variable  $\{x_i\}_{i=1}^d$ , denoted by  $\mathcal{Q}_{\mathbf{k}}(K)$ , with  $\mathbf{k} = [k_1, \dots, k_d]$ . Clearly, the dimension of this space, denoted by  $\dim(\mathcal{Q}_{\mathbf{k}}(K))$ , is  $\prod_{i=1}^d (k_i + 1)$ . Let us also define the corresponding truncated polynomial space  $\mathcal{P}_k(K)$  as the span of the monomials with degree less than or equal to  $k$ . To determine the dimension of the truncated space, we note that the number of components can be expressed with the triangular or tetrahedral number  $T_n^d$ ,  $d = \{2, 3\}$ , such that

$$(8) \quad \dim(\mathcal{P}_k(K)) = T_{k+1}^d, \quad T_n^d = \frac{\prod_{i=1}^d (n + i - 1)}{d!}.$$

**3.2.1. Construction of polynomial spaces.** Given an order  $k$ , it is trivial to form the set of monomials  $q^k = \{x^i\}_{i=0}^k$ , that spans the 1D space  $\mathcal{Q}_k(K)$ . We construct a basis spanning the multi-dimensional space  $\mathcal{Q}_{\mathbf{k}}(K)$  with a Cartesian product of the monomials for each dimension, i.e.,  $\{q_1^{k_1} \times \dots \times q_d^{k_d}\}$ , thus we have  $\mathcal{Q}_{\mathbf{k}}(K) = \text{span}\{\prod_{i=1}^d x_i^{\alpha_i} \text{ s.t. } \alpha_i \leq k_i\}$ . Let us denote by  $|\alpha|$  the summation of the exponents for a given monomial. Then, the multi-dimensional truncated space is defined as  $\mathcal{P}_k(K) = \text{span}\{\prod_{i=1}^d x_i^{\alpha_i} \text{ s.t. } |\alpha| \leq k\}$ .

Let us also define Lagrangian polynomials spanning  $\mathcal{Q}_{\mathbf{k}}(K)$ , which will be used in the definition of moments in Sect. 3.3. Given an order  $k$  and a set  $\mathcal{N}_k$  of different nodes in  $\mathbb{R}$ , usually equidistant in the interval  $[x_0, x_k]$ , we can define the set of Lagrangian polynomials

$\{\ell_i\}_{i=0}^k$  as,

$$(9) \quad \ell_i := \frac{\prod_{n \in \mathcal{N}_k \setminus i} (x - x_n)}{\prod_{n \in \mathcal{N}_k \setminus i} (x_i - x_n)},$$

where we indistinctly represent nodes by their index  $i$  or its position  $x_i$ . The set of all polynomials  $\{\ell_i^k\}_{i=0}^k$  defines the Lagrangian basis  $L^k$ . In the grad-conforming (Lagrangian) FE, the definition of moments  $\{\sigma_a\}_{a=1}^k$  simply consists in the evaluation of the functions on the given points  $x \in \mathcal{N}_k$ . Clearly, polynomials in (9) evaluated at points  $x_a$  satisfy the duality  $\sigma_a(\ell_i) = \ell_i(x_a) = \delta_{ai}$  for every  $a, i \in \{0, \dots, k\}$ , i.e., they are shape functions. For multi-dimensional spaces, we define the set of nodes as the Cartesian product of 1D nodes. Given a  $d$ -dimensional space of order  $\mathbf{k} = [k_1, \dots, k_d]$ , the set of nodes is defined as  $\mathcal{N}^{\mathbf{k}} = \mathcal{N}^{k_1} \times \dots \times \mathcal{N}^{k_d}$ .

**3.2.2. Hexahedra.** The space of functions on the cell  $\mathcal{V}_k(K)$  for this sort of elements is defined as the space of gradients for the scalar polynomial space  $\mathcal{Q}_k(K)$ , i.e.,

$$(10) \quad \mathcal{V}_k(K) := \{\mathcal{Q}_{k-1,k}(K) \times \mathcal{Q}_{k,k-1}(K)\},$$

$$(11) \quad \mathcal{V}_k(K) := \{\mathcal{Q}_{k-1,k,k}(K) \times \mathcal{Q}_{k,k-1,k}(K) \times \mathcal{Q}_{k,k,k-1}(K)\},$$

in 2D and 3D, resp. Let us illustrate the polynomial space with a couple of examples.

**Example 3.1.** *The polynomial space for the lowest order ( $k = 1$ ) edge hexahedral element is defined as*

$$\mathcal{V}_k(K) = \{\mathcal{Q}_{0,1,1}(K) \times \mathcal{Q}_{1,0,1}(K) \times \mathcal{Q}_{1,1,0}(K)\}$$

which can be represented as the span of the set of vector-valued functions

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_2 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_3 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} x_2 x_3 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ x_3 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1 x_3 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ x_1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ x_2 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ x_1 x_2 \end{bmatrix} \right\}.$$

**Example 3.2.** *The polynomial space for the second order quadrilateral element is defined as*

$$\mathcal{V}_k(K) = \{\mathcal{Q}_{1,2}(K) \times \mathcal{Q}_{2,1}(K)\},$$

which can be represented by the spanning set

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ 0 \end{bmatrix}, \begin{bmatrix} x_2 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 x_2 \\ 0 \end{bmatrix}, \begin{bmatrix} x_2^2 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 x_2^2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1^2 \end{bmatrix}, \begin{bmatrix} 0 \\ x_2 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1 x_2 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1^2 x_2 \end{bmatrix} \right\}.$$

**3.2.3. Tetrahedra.** The polynomial space for tetrahedral elements is slightly more involved. For the sake of brevity, let us omit the cell (i.e.,  $K$ ) in the notation for the local FE spaces. Let us start by defining the homogeneous polynomial space  $[\tilde{\mathcal{P}}_k]^d := [\mathcal{P}_k]^d \setminus [\mathcal{P}_{k-1}]^d$ , where  $[\mathcal{P}_k]^2 = \mathcal{P}_k \times \mathcal{P}_k$  and  $[\mathcal{P}_k]^3 = \mathcal{P}_k \times \mathcal{P}_k \times \mathcal{P}_k$ . The space  $[\tilde{\mathcal{P}}_k]^d$  has dimension  $d \cdot \dim(\tilde{\mathcal{P}}_k)$ , and using (8) we have:

$$\dim(\tilde{\mathcal{P}}_k) = \dim(\mathcal{P}_k) - \dim(\mathcal{P}_{k-1}) = T_{k+1}^d - T_k^d = T_{k+1}^{d-1}.$$

The function space of order  $k$  on a tetrahedron is then defined as

$$(12) \quad \mathcal{V}_k(K) = [\mathcal{P}_{k-1}]^d \oplus \mathcal{S}_k,$$

where  $\mathcal{S}_k$  is the space of polynomials

$$\mathcal{S}_k := \{p(\mathbf{x}) \in [\tilde{\mathcal{P}}_k]^d \text{ such that } p(\mathbf{x}) \cdot \mathbf{x} = 0\}.$$

Note that if  $p(\mathbf{x}) \in [\tilde{\mathcal{P}}_k]^d$ , then  $p \cdot \mathbf{x} \in \tilde{\mathcal{P}}_{k+1}$  and any polynomial in  $\tilde{\mathcal{P}}_{k+1}$  may be written in this way. The dimension of the space  $\mathcal{S}_k$  is

$$(13) \quad \dim(\mathcal{S}_k) = d \cdot \dim(\tilde{\mathcal{P}}_k) - \dim(\tilde{\mathcal{P}}_{k+1}),$$

which leads to  $(k+2)k$  in 3D and  $k$  in 2D, resp. Among all the possible forms of representing the space  $\mathcal{S}_k$  in 3D, we consider the following spanning set

$$(14a) \quad \mathcal{S}_k = \text{span} \left\{ \bigcup_{\beta=1}^k \bigcup_{\alpha=1}^{k+1-\beta} \left( \begin{bmatrix} -x_1^{\alpha-1} x_2^{k-\alpha-\beta+2} x_3^{\beta-1} \\ x_1^\alpha x_2^{k-\alpha-\beta+1} x_3^{\beta-1} \\ 0 \end{bmatrix}, \begin{bmatrix} -x_1^{k-\alpha-\beta+1} x_2^{\beta-1} x_3^\alpha \\ 0 \\ x_1^{k-\alpha-\beta+2} x_2^{\beta-1} x_3^{\alpha-1} \end{bmatrix} \right) \right\},$$

$$(14b) \quad \bigcup_{\alpha=1}^k \begin{bmatrix} 0 \\ -x_1^0 x_2^{\alpha-1} x_3^{k-\alpha+1} \\ x_1^0 x_2^\alpha x_3^{k-\alpha} \end{bmatrix} \left. \right\}.$$

**Proposition 3.1.** *The set of vector functions defined in (14) forms a basis of the space  $\mathcal{S}_k$ .*

*Proof.* First, we note that the proposed basis contains  $\{p_i(\mathbf{x})\}_{i=1}^{(k+2)k}$  vector functions. Clearly, the total degree of the monomials found on each component for all functions is  $k$ , thus  $S \subset \tilde{\mathcal{P}}_k^d$ . Further, it is easy to check that  $p_i(\mathbf{x}) \cdot \mathbf{x} = 0$  for any  $p_i(\mathbf{x}) \in S$ . The proof is completed by showing that all the functions are linearly independent. It is trivial to see that all the functions in the first set of functions are indeed linearly independent; the two sets have different non-zero components. Finally, vector functions from the last subset (14b) are independent of  $x_1$ , whereas all functions in the previous subset in (14a) do depend on  $x_1$  and are linearly independent among them.  $\square$

In two dimensions, the analytical expression of the spanning set simply reduces to

$$\mathcal{S}_k = \text{span} \left\{ \bigcup_{\alpha=1}^k \begin{bmatrix} -x_1^{\alpha-1} x_2^{k-\alpha+1} \\ x_1^\alpha x_2^{k-\alpha} \end{bmatrix} \right\}.$$

The dimension of the space  $\mathcal{V}_k(K)$  for tetrahedra can be obtained by adding (8) and (13),

$$(15) \quad \dim(\mathcal{V}_k(K)) = \frac{k \prod_{i=2}^d (k+i)}{(d-1)!}.$$

Let us give some examples of polynomial bases for tetrahedral edge elements spanning the requested spaces.

**Example 3.3.** *A set of polynomial spanning  $\mathcal{V}_1(K) = [P_0]^3 \oplus \mathcal{S}_1$  (i.e., lowest order tetrahedral element) is*

$$\left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} -x_2 \\ x_1 \\ 0 \end{bmatrix}, \begin{bmatrix} -x_3 \\ 0 \\ x_1 \end{bmatrix}, \begin{bmatrix} 0 \\ -x_3 \\ x_2 \end{bmatrix} \right\}.$$

**Example 3.4.** *A polynomial base spanning  $\mathcal{V}_2(K) = [P_1]^2 \oplus \mathcal{S}_2$  (i.e., second-order triangular element) is*

$$\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} x_1 \\ 0 \end{bmatrix}, \begin{bmatrix} x_2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ x_1 \end{bmatrix}, \begin{bmatrix} 0 \\ x_2 \end{bmatrix}, \begin{bmatrix} -x_2^2 \\ x_1 x_2 \end{bmatrix}, \begin{bmatrix} -x_1 x_2 \\ x_1^2 \end{bmatrix} \right\}.$$

In both cases, vector functions that span the subspaces  $[\mathcal{P}_{k-1}]^d \subset \mathcal{V}_k(K)$  and  $\mathcal{S}_k \subset \mathcal{V}_k(K)$  can easily be identified from the full sets of vector-valued functions in Exs. 3.3 and 3.4.

Note that local spaces  $\mathcal{V}_k(K)$  lie between the full polynomial spaces of order  $k-1$  and  $k$ , i.e.,  $[\mathcal{Q}_{k-1}(K)]^d \subset \mathcal{V}_k(K) \subset [\mathcal{Q}_k(K)]^d$ ,  $[\mathcal{P}_{k-1}(K)]^d \subset \mathcal{V}_k(K) \subset [\mathcal{P}_k(K)]^d$  for hexahedra and tetrahedra, resp. This kind of elements are called edge FEs of the first kind [1]. Another edge FE, the so-called second kind, was introduced also by Nédélec in [42]. It follows similar ideas but considers full polynomial spaces, i.e.,  $[\mathcal{Q}_k(K)]^d$  or  $[\mathcal{P}_k(K)]^d$ , instead of anisotropic polynomial spaces (see (11)) or incomplete polynomial spaces (see (12))



for hexahedra and tetrahedra, resp., which clearly simplifies the implementation of the polynomial spaces. Edge elements of the second kind offer better constants in the error estimates at the cost of increasing the number of DoFs (see detailed definitions in [54, Ch. 2]).

**3.3. Edge moments in the reference element.** In order to complete the definition of edge FEs, it remains to define a set of (linearly independent) functionals to be applied on  $\mathcal{V}_k(\hat{K})$ . Edge moments are integral quantities over geometrical sub-entities of the cell against some test functions. These moments involve directions in 1D entities (edges) and orientations in 2D (faces) or 3D (volumes). Thus, we need to define local orientations for the lower dimension geometrical entities of the reference cell, see Figs. 1a and 2a. Let us distinguish between three different subsets of moments  $\hat{\Sigma}$ , namely DoFs related to edges  $\sigma_{\hat{e}}$ , to faces  $\sigma_{\hat{f}}$  and volume  $\sigma_{\hat{K}}$  such that  $\hat{\Sigma} = \sigma_{\hat{e}} \cup \sigma_{\hat{f}} \cup \sigma_{\hat{K}}$ .

**3.3.1. Hexahedra.** The set of functionals, for the reference element, that form the basis in two dimensions reads ([2, Ch. 6]):

$$(16a) \quad \sigma_{\hat{e}}(\hat{\mathbf{u}}_h) := \int_{\hat{e}} (\hat{\mathbf{u}}_h \cdot \hat{\boldsymbol{\tau}}) \hat{q} \quad \forall \hat{q} \in \mathcal{P}_{k-1}(\hat{e}), \quad \forall \hat{e} \in \hat{K},$$

$$(16b) \quad \sigma_{\hat{K}}(\hat{\mathbf{u}}_h) := \int_{\hat{K}} \hat{\mathbf{u}}_h \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in \mathcal{Q}_{k-1,k-2}(\hat{K}) \times \mathcal{Q}_{k-2,k-1}(\hat{K}).$$

We will have  $4k$  DoFs associated to edges and  $2k(k-1)$  inner DoFs. Therefore, the complete set of functionals has cardinality  $n_{\hat{\Sigma}} = 2k(k+1)$ . In the case of  $k=1$ , only edge DoFs  $\sigma_{\hat{e}}$  appear. In three dimensions the set is defined as

$$(17a) \quad \sigma_{\hat{e}}(\hat{\mathbf{u}}_h) := \int_{\hat{e}} (\hat{\mathbf{u}}_h \cdot \hat{\boldsymbol{\tau}}) \hat{q} \quad \forall \hat{q} \in \mathcal{P}_{k-1}(\hat{e}), \quad \forall \hat{e} \in \hat{K}$$

$$(17b) \quad \sigma_{\hat{f}}(\hat{\mathbf{u}}_h) := \int_{\hat{f}} (\hat{\mathbf{u}}_h \times \hat{\mathbf{n}}) \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in \mathcal{Q}_{k-2,k-1}(\hat{\mathcal{F}}) \times \mathcal{Q}_{k-1,k-2}(\hat{\mathcal{F}}), \quad \forall \hat{\mathcal{F}} \in \hat{K}$$

$$\sigma_{\hat{K}}(\hat{\mathbf{u}}_h) := \int_{\hat{K}} \hat{\mathbf{u}}_h \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in \mathcal{Q}_{k-1,k-2,k-2}(\hat{K}) \times \mathcal{Q}_{k-2,k-1,k-2}(\hat{K}) \times \mathcal{Q}_{k-2,k-2,k-1}(\hat{K}),$$

where  $\hat{\boldsymbol{\tau}}$  is the unit vector along the edge and  $\hat{\mathbf{n}}$  the unit normal to the face. In this case, we have  $12k$  DoFs associated to edges,  $6 \cdot 2k(k-1)$  DoFs associated to the 6 faces of the cell and  $3k(k-1)^2$  inner DoFs. We have a total number of  $n_{\hat{\Sigma}} = 3k(k+1)^2$  DoFs. Note that in the case of the lowest order elements, i.e.,  $k=1$ , only DoFs associated to edges appear. For higher order elements, i.e.,  $k \geq 2$ , all kinds of DoFs occur in both dimensions.

DoFs are labelled in the reference cell as follows. First, for every DoF, we can determine the geometrical entity that *owns* it, e.g., an edge of the reference FE in (17a). Within every geometrical entity, there is a one-to-one map between DoFs and test functions. Thus, we can number DoFs by the numbering of the test functions in the test space, e.g., the test functions in  $\mathcal{P}_{k-1}(\hat{e})$  for the DoFs in (17a). We note that all the test spaces in the DoF definitions can be built using a nodal-based (Lagrangian) basis, and thus, every DoF in a geometrical object can be conceptually associated to one and only one node. The numbering of the nodes in the geometrical object is determined by its orientation in the reference FE. The composition of a geometrical object numbering (see Figs. 1–2) and the object-local node numbering provides the local numbering of DoFs within the reference cell.

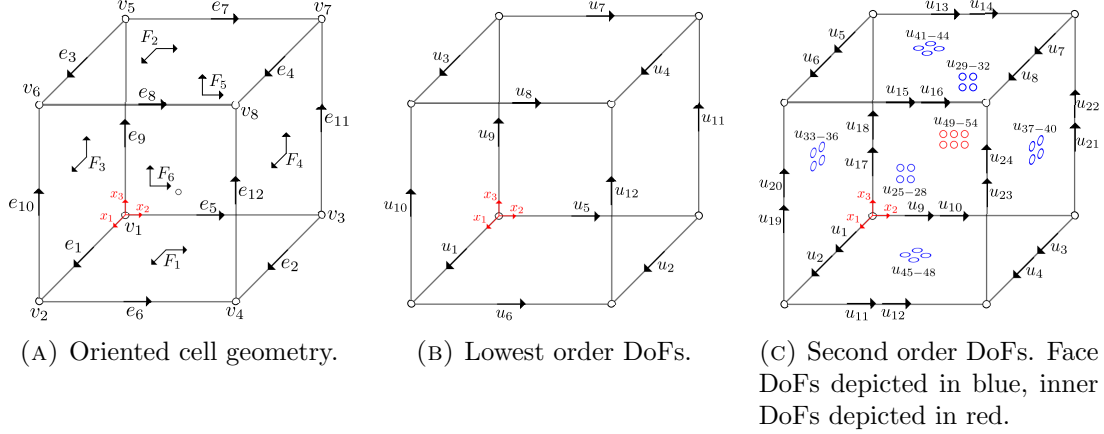


FIGURE 1. 3D hexahedral reference FE.

3.3.2. *Tetrahedra.* The set of functionals described in this section follows [2, Ch. 5]. In the 2D case, the set of moments for the reference FE is defined as

$$(18a) \quad \sigma_{\hat{e}}(\hat{\mathbf{u}}_h) := \int_{\hat{e}} (\hat{\mathbf{u}}_h \cdot \hat{\boldsymbol{\tau}}) \hat{q} \quad \forall \hat{q} \in \mathcal{P}_{k-1}(\hat{e}), \quad \forall \hat{e} \in \hat{K}$$

$$(18b) \quad \sigma_{\hat{K}}(\hat{\mathbf{u}}_h) := \int_{\hat{K}} \hat{\mathbf{u}}_h \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in [\mathcal{P}_{k-2}(\hat{K})]^2.$$

Clearly, we have  $3k$  edge DoFs and  $k(k-1)$  inner DoFs, which lead to a total number of  $n_{\hat{\Sigma}} = k(k+2)$  DoFs. In three dimensions, the set  $\hat{\Sigma}$  is defined as

$$(19a) \quad \sigma_{\hat{e}}(\hat{\mathbf{u}}_h) := \int_{\hat{e}} (\hat{\mathbf{u}}_h \cdot \hat{\boldsymbol{\tau}}) \hat{q} \quad \forall \hat{q} \in \mathcal{P}_{k-1}(\hat{e}), \quad \forall \hat{e} \in \hat{K}$$

$$(19b) \quad \sigma_{\hat{f}}(\hat{\mathbf{u}}_h) := \frac{1}{\|\hat{\mathcal{F}}\|} \int_{\hat{\mathcal{F}}} \hat{\mathbf{u}}_h \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in [\mathcal{P}_{k-2}(\hat{K})]^3 \text{ s.t. } \hat{\mathbf{q}} \cdot \hat{\mathbf{n}} = 0, \quad \forall \hat{f} \in \hat{K}$$

$$(19c) \quad \sigma_{\hat{K}}(\hat{\mathbf{u}}_h) := \int_{\hat{K}} \hat{\mathbf{u}}_h \cdot \hat{\mathbf{q}} \quad \forall \hat{\mathbf{q}} \in [\mathcal{P}_{k-3}(\hat{K})]^3,$$

where  $\hat{\mathbf{n}}$  in (19b) is the unit normal to the face. The set of moments (19) contains  $6k$  edge DoFs,  $4k(k-1)$  face DoFs and  $\frac{k(k-1)(k-2)}{2}$  inner DoFs. Therefore, the tetrahedral edge element has a total number of  $n_{\hat{\Sigma}} = \frac{k(k^2+5k+6)}{2}$  DoFs. The local numbering of DoFs is analogous as for the hexahedral case.

**Remark 3.1.** *The face moments in the definition (19b) seem to differ from the rest of definitions in (19), which are not scaled with a geometrical entity measure. We follow here [2, Ch. 5], where this expression of face moments is used to prove affine equivalence, i.e., proving that DoFs are affine invariant under the transformation from the reference to the physical element.*

Note that in the case of the lowest order elements, i.e.,  $k=1$ , only DoFs associated to edges occur. For second order elements, i.e.,  $k=2$ , we also find DoFs related to faces (inner DoFs in 2D), and it is in orders higher than two where all kinds of DoFs occur. Note that vector-valued test functions  $\hat{\mathbf{q}}$  involved in (18b), (19b) and (19c) can be understood as products between linearly independent vectors  $\{\hat{\boldsymbol{\tau}}_i\}_1^d$  that form a basis in the geometrical entity of dimension  $d$  and scalar polynomials  $\hat{q} \in \mathcal{P}_{k-d}$ .

3.4. **Construction of edge shape functions.** Usually, low order edge elements are implemented via hard-coded expressions of their respective shape functions. However, such an approach is not suitable for high order edge FEs, which involve complex analytical

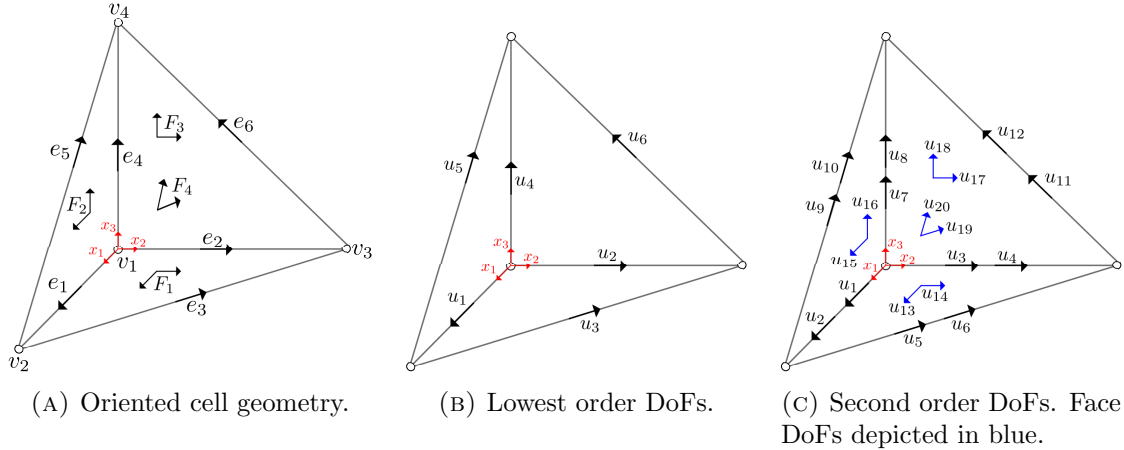


FIGURE 2. 3D tetrahedral reference FE.

expressions of the shape functions. In this section, we provide an automatic generator of arbitrary order shape functions.

The definition of the moments for the edge FE (in (16a,16b), (17a,17b,??) or (18a,18b), (19a-19c) for hexahedra and tetrahedra, resp) requires the selection of functions spanning the requested polynomial space in each case. First, we generate a *pre-basis*  $\{\varphi^a\}_{a=1}^{n_\Sigma}$  that spans the local FE space  $\mathcal{V}_k(\hat{K})$ . To this end, we consider the tensor product of Lagrangian polynomial basis (see Sect. 3.2.1) for hexahedra or the combination of a Lagrangian basis of one order less in (12) plus the based of monomials in (14) for tetrahedra. Our goal is to build another (canonical) basis  $\{\phi^a\}_{a=1}^{n_\Sigma}$  that spans the same space and additionally satisfies  $\sigma_a(\phi^b) = \delta_{ab}$  for  $a, b \in \{1, \dots, n_\Sigma\}$ , i.e., the basis of shape functions for the edge element. Thus, we are interested in obtaining a linear combination of the functions of the *pre-basis* such that the duality between moments and functions is satisfied. As a result, an edge shape function  $\phi^a$  can be written as  $\phi^a = \sum_{b=1}^{n_\Sigma} Q_{ab} \varphi^b$ . Let us make use of Einstein's notation to provide the definition of the change of basis between the two of them. We have:

$$(20) \quad \phi^b = Q_{bc} \varphi^c, \quad \sigma_a(\phi^b) = \sigma_a(Q_{bc} \varphi^c), \quad \delta_{ab} = \sigma_a(\varphi^c) Q_{bc},$$

or in compact form  $\mathbf{I} = \mathbf{C} \mathbf{Q}^T$ , thus  $\mathbf{Q}^T = \mathbf{C}^{-1}$ . As a result, the edge shape functions are obtained as  $\phi^a = Q_{ab} \varphi^b = C_{ba}^{-1} \varphi^b$ . For the sake of illustration, we provide some examples of full sets of edge shape functions for different orders. To make the visualization easy, we provide only 2D examples for first and second order square (Figs. 3 and 4) and triangular elements (Figs. 5 and 6). In these figures, we put a circle on top of the node corresponding to the DoF dual to the shape function. The superscript indicates the local moment numbering on the particular geometrical entity.

#### 4. GLOBAL FE SPACES AND CONFORMITY

A FE space is  $H(\text{curl})$ -conforming if the tangential components at the interface between elements are continuous, i.e., they do not have to satisfy normal continuity over element faces. The discrete global FE space where the magnetic field solution  $\mathbf{u}_h$  lies is defined as

$$\mathcal{ND}_k(\Omega) = \{\mathbf{v}_h \in H(\text{curl}, \Omega) \text{ such that } \mathbf{v}_h|_K \in \mathcal{V}_k(K) \forall K \in \mathcal{T}_h\},$$

where  $\mathcal{V}_k$  has been defined for the hexahedral and tetrahedral edge FEs in Sect. 3.2.

**4.1. Moments in the physical space and the Piola map.** In this section we define a set of DoFs (moments) in the physical space that allow one to interpolate analytical functions onto the edge FE space, e.g., to enforce Dirichlet data. One can check that

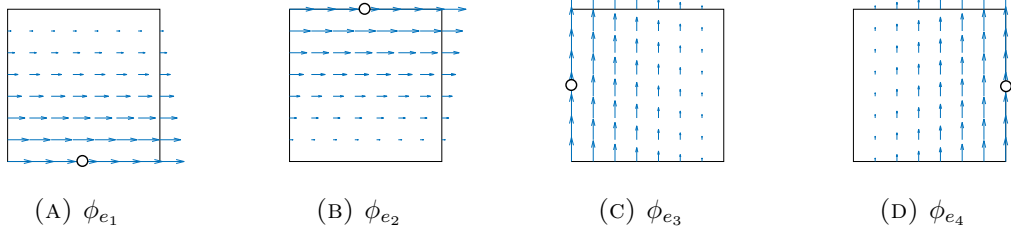


FIGURE 3. Vector-field plots of the shape functions in the lowest order 2D hexahedra. The indices of edges follow the geometrical entities indices for the reference hexahedron in Fig. 1a restricted to the plane  $z = 0$ . Auxiliary circles denote the geometrical entity where the moment is defined.

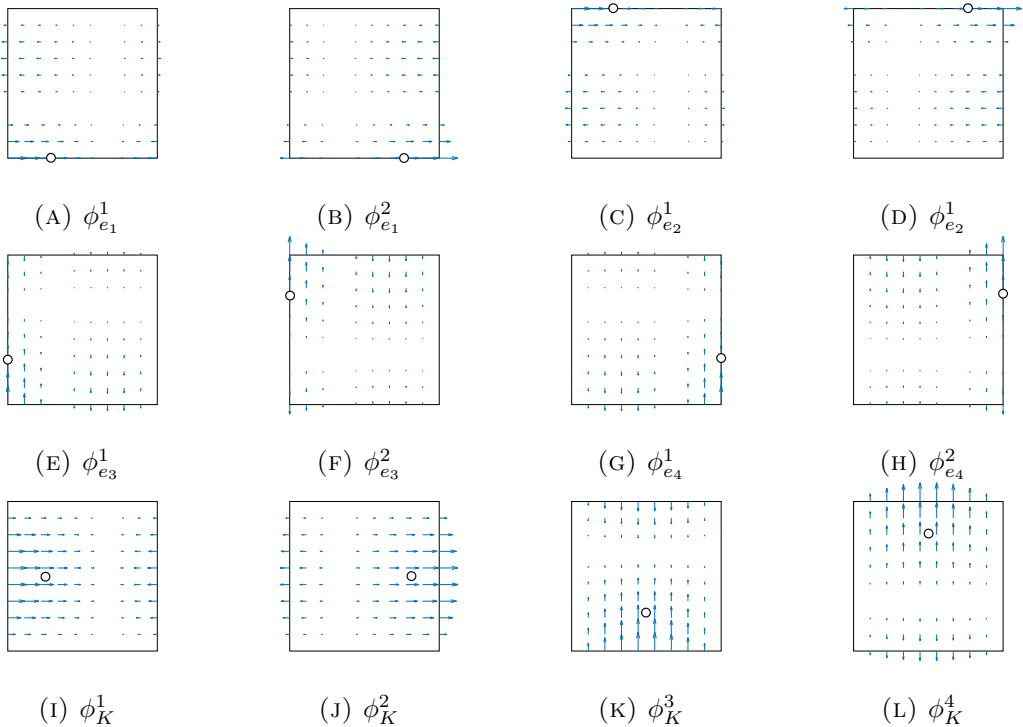


FIGURE 4. Vector-field plots of the shape functions in the second order hexahedral edge element. The indices of edges follow the geometrical entities indices for the reference hexahedron in Fig. 1a restricted to the plane  $z = 0$ . Auxiliary circles denote the geometrical entity where the moment is defined.

the continuity of those DoFs which lay at the boundary between cells provides the desired continuity of the tangential component. On top of that, using the so-called covariant Piola mapping  $\Psi_K(\hat{\mathbf{v}}) \doteq \hat{\Psi}_K(\hat{\mathbf{v}}) \circ \Phi_K^{-1}$ , where

$$\hat{\Psi}_K(\hat{\mathbf{v}}) \doteq \mathbf{J}_K^{-T} \hat{\mathbf{v}},$$

it can be checked that a DoF in the reference space of a function  $\hat{\mathbf{v}}$  on  $\hat{K}$  is equal to the corresponding DoF in the physical space for  $\Psi_K(\hat{\mathbf{v}})$  on  $K$  (see [2, Ch. 5] for details). Thus, the Piola mapping, which preserves tangential traces of vector fields [21], is the key to achieve a curl-conforming global space by using a FE space relying on a reference FE definition.

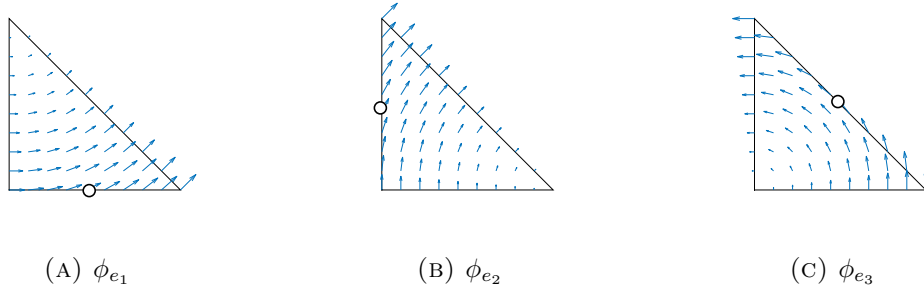


FIGURE 5. Vector-field plots of the shape functions in the lowest order tetrahedral element. The indices of edges follow the geometrical entities indices for the reference tetrahedron in Fig. 2a restricted to the plane  $z = 0$ . Auxiliary circles denote the geometrical entity where the moment is defined.

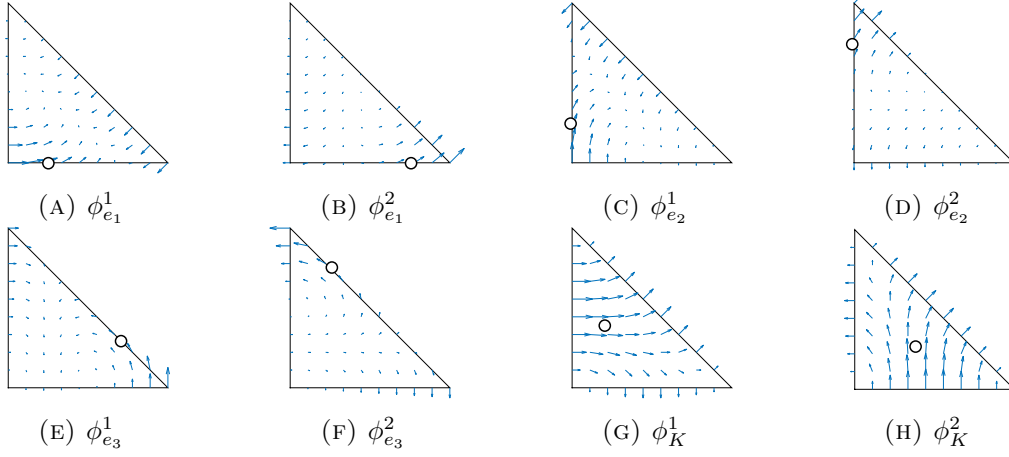


FIGURE 6. Vector-field plots of the shape functions in the second order edge element. The indices of edges follow the geometrical entities indices for the reference tetrahedron in Fig. 2a restricted to the plane  $z = 0$ . Auxiliary circles denote the geometrical entity where the moment is defined.

4.1.1. *Hexahedra.* Let us now define the moments for a curl-conforming FE defined on a general hexahedron. Given an integer  $k \geq 1$ , the set of functionals that forms the basis in two dimensions reads:

$$\begin{aligned} \sigma_e(\mathbf{u}_h) &:= \int_e (\mathbf{u}_h \cdot \boldsymbol{\tau}) q \quad \forall q \in \mathcal{P}_{k-1}(e), \quad \forall e \in K, \\ \sigma_K(\mathbf{u}_h) &:= \int_K \mathbf{u}_h \cdot \mathbf{q}, \quad \forall \mathbf{q} \text{ obtained by mapping } \mathbf{q} = \left( \frac{1}{\det(\mathbf{J}_K)} \right) \mathbf{J}_K \hat{\mathbf{q}}, \\ &\quad \hat{\mathbf{q}} \in \mathcal{Q}_{k-1, k-2}(\hat{K}) \times \mathcal{Q}_{k-2, k-1}(\hat{K}), \end{aligned}$$

where  $\boldsymbol{\tau}$  is the unit vector along the edge. In 3D, the set of functionals reads

$$\begin{aligned}\sigma_e(\mathbf{u}_h) &:= \int_e (\mathbf{u}_h \cdot \boldsymbol{\tau}) q \quad \forall q \in \mathcal{P}_{k-1}(e), \quad \forall e \in K \\ \sigma_f(\mathbf{u}_h) &:= \int_{\mathcal{F}} (\mathbf{u}_h \times \mathbf{n}) \cdot \mathbf{q} \quad \forall \mathbf{q} \text{ obtained by mapping } \mathbf{q} = \mathbf{J}_K^{-T}(\mathbf{J}_{\hat{\mathcal{F}}}\hat{\mathbf{q}}), \\ &\quad \hat{\mathbf{q}} \in \mathcal{Q}_{k-2,k-1}(\hat{\mathcal{F}}) \times \mathcal{Q}_{k-1,k-2}(\hat{\mathcal{F}}), \quad \forall \mathcal{F} \in K \\ \sigma_K(\mathbf{u}_h) &:= \int_K \mathbf{u}_h \cdot \mathbf{q} \quad \forall \mathbf{q} \text{ obtained by mapping } \mathbf{q} = \left( \frac{1}{\det(\mathbf{J}_K)} \right) \mathbf{J}_K \hat{\mathbf{q}}, \\ &\quad \hat{\mathbf{q}} \in \mathcal{Q}_{k-1,k-2,k-2}(\hat{K}) \times \mathcal{Q}_{k-2,k-1,k-2}(\hat{K}) \times \mathcal{Q}_{k-2,k-2,k-1}(\hat{K})\end{aligned}$$

where  $\mathbf{n}$  is the unit normal to the face. The definition of the face moments in (??) requires to transfer either the 3D vector  $(\mathbf{u}_h \times \mathbf{n})$  to the face  $\mathcal{F}$  or the vector  $\hat{\mathbf{q}}$ , contained in a reference face, to the 3D physical cell. We choose this latter option, which implies the transformation of the vector  $\hat{\mathbf{q}}$  to the reference cell through the face Jacobian  $\mathbf{J}_{\hat{\mathcal{F}}} = \frac{\partial \Phi_{\hat{K}}}{\partial \mathbf{x}_{\hat{\mathcal{F}}}}$ .

4.1.2. *Tetrahedra.* On a general tetrahedron  $K$ , we define the moments for a curl-conforming FE as follows. Given an integer  $k \geq 1$ , the set of functionals that forms the basis in two dimensions reads

$$\begin{aligned}\sigma_e(\mathbf{u}_h) &:= \int_e (\mathbf{u}_h \cdot \boldsymbol{\tau}) q \quad \forall q \in \mathcal{P}_{k-1}(e), \quad \forall e \in K \\ \sigma_K(\mathbf{u}_h) &:= \int_K \mathbf{u}_h \cdot \mathbf{q} \quad \forall \mathbf{q} \text{ obtained by mapping} \\ &\quad \mathbf{q} = \left( \frac{1}{\det(\mathbf{J}_K)} \right) \mathbf{J}_K \hat{\mathbf{q}}, \quad \hat{\mathbf{q}} \in [\mathcal{P}_{k-2}(\hat{K})]^2.\end{aligned}$$

Clearly, we have  $3k$  edge DoFs and  $k(k-1)$  inner DoFs, which lead to a total number of  $n_{\Sigma} = k(k+2)$  DoFs. In three dimensions the set  $\Sigma$  is defined as

$$\begin{aligned}\sigma_e(\mathbf{u}_h) &:= \int_e (\mathbf{u}_h \cdot \boldsymbol{\tau}) q \quad \forall q \in \mathcal{P}_{k-1}(e), \quad \forall e \in K \\ \sigma_f(\mathbf{u}_h) &:= \frac{1}{\|\mathcal{F}\|} \int_{\mathcal{F}} \mathbf{u}_h \cdot \mathbf{q} \quad \forall \mathbf{q} = \mathbf{J}_K \hat{\mathbf{q}} \quad \text{s.t. } \hat{\mathbf{q}} \cdot \hat{\mathbf{n}} = 0, \\ &\quad \hat{\mathbf{q}} \in [\mathcal{P}_{k-2}(\hat{f})]^3 \quad \forall f \in K \\ \sigma_K(\mathbf{u}_h) &:= \int_K \mathbf{u}_h \cdot \mathbf{q} \quad \forall \mathbf{q} \text{ obtained by mapping} \\ &\quad \mathbf{q} = \left( \frac{1}{\det(\mathbf{J}_K)} \right) \mathbf{J}_K \hat{\mathbf{q}}, \quad \hat{\mathbf{q}} \in [\mathcal{P}_{k-3}(\hat{K})]^3.\end{aligned}$$

4.2. **Nedelec interpolator.** The space of edge FE functions can be represented as the range of an interpolation operator  $\pi^h$  that is well defined for sufficiently smooth functions  $\mathbf{u} \in H(\mathbf{curl}, \Omega)$  by

$$(25) \quad \pi^h(\mathbf{u}) := \sum_a u^a(\mathbf{u}) \phi^a$$

where  $u^a(\mathbf{u}) = \sigma^a(\mathbf{u})$  are the evaluation of the moments for the function  $\mathbf{u}$  described for the hexahedra and tetrahedra cases in Sect. 4.1 for all  $e \in \mathcal{E}$ ,  $F \in \mathcal{F}$  and  $K \in \mathcal{T}_h$ . Note that Dirichlet boundary conditions can be strongly imposed in the resulting system (usual implementation in FE codes) by evaluating the moments corresponding to edges and/or faces on the Dirichlet boundary given the analytical expression of the tangential trace.

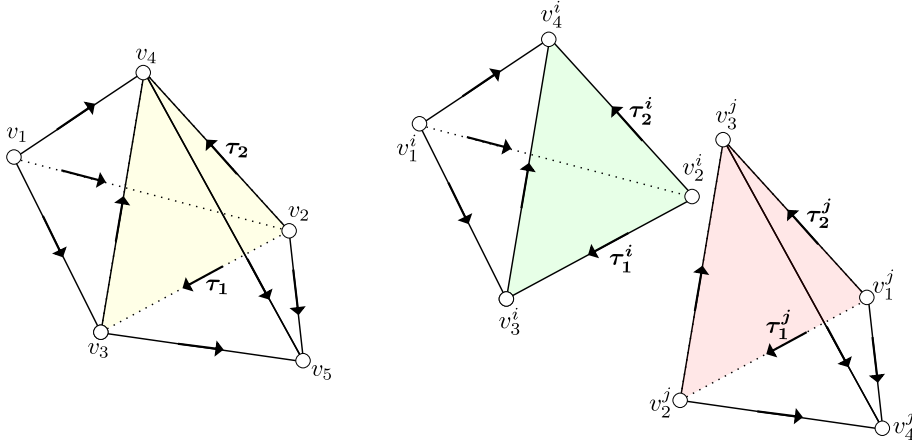


FIGURE 7. Oriented mesh. Global information (left) and its local, oriented, counterpart (right) for two elements  $K_i$  and  $K_j$ . Common edges for two adjacent tetrahedra will always agree in its direction, as well as chosen tangent vectors  $\tau_1$  and  $\tau_2$  to the common face.

**4.3. Global DoFs.** In order to guarantee global inter-element continuity with Piola-mapped elements, special care has to be taken with regard to the orientation of edges and faces at the cell level. Let us consider a global numbering for the vertices in a mesh and a local numbering at the cell level. Given an edge/face, sorting its vertices with respect to the local (resp. global) index of their vertices, one determines the local (resp. global) orientation of the edge/face. A mesh in which the local and global orientation of all its edges and faces coincide is called an *oriented mesh*. For oriented meshes, common edges or faces for two adjacent tetrahedra will always agree in its orientation, thus represent the same global DoF. Tetrahedral meshes are oriented with a simple local renumbering [21], and we restrict ourselves to hexahedral octree meshes that are oriented by construction.

Local DoFs are uniquely determined by the cell in which they are defined, the geometrical entity within the reference cell that owns them and the local numbering of DoFs within the geometrical entity (see Sect. 3.3.1). The local numbering only depends on the orientation of the geometrical entity through the ordering of its vertices.

Global DoFs are defined as an equivalence class over the union of the local DoFs for all cells. Two local DoFs are the same global DoF if and only if they belong to the same geometrical object and the same local numbering within the geometrical object in their respective cells. The previous equivalence class leads to a curl-conforming FE space if local orientations of geometrical objects coincide with a global orientation, i.e., on oriented meshes only.

For the sake of illustration, see the simple mesh depicted in Fig. 7, composed by two tetrahedra defined by the global vertices  $v_1, v_2, v_3, v_4$  and  $v_5$ . The two elements share a common face, defined by 3 vertices that have different local indices for all  $v_i \in K_i$  and  $v_j \in K_j$ . The ordering convention, local indexing according to ascending global indices, ensures that both triangles agree on the direction of the common edges and the common face. Note that the Jacobian of the transformation may become negative with the orientation procedure. We only need to make sure that the absolute value of the Jacobian is taken whenever the measure of the change of basis is applied. The situation is much more involved for general hexahedral meshes. In any case, our implementation of hexahedral meshes relies on octree meshes, where consistency is ensured. It is easy to check that an octree mesh in which a cell inherits the orientation of its parent is *oriented* by construction.

5. EDGE FES IN  $h$ -ADAPTIVE MESHES

In this section, we expose an implementation procedure for *non-conforming* meshes and edge FEs of arbitrary order. In the following, we restrict ourselves to hexahedral meshes, even though the extension to tetrahedral meshes is straightforward. Our procedure can also be readily extended to any FE space that relies on a *pre-basis* of Lagrangian polynomial spaces plus a change of basis, e.g., Raviart-Thomas FEs.

**5.1. Hierarchical AMR on octree-based meshes.** The AMR generation relies on hierarchically refined octree-based hexahedral meshes [55] and the p4est [27] library is used for such purpose. In this method, one must enforce constraints to ensure the conformity of the global FE space, which amounts to compute constraints between coarser and refined geometrical entities shared by two cells with different level of refinement [16]. For Lagrangian elements, restriction operators in geometrical sub-entities are generally obtained by evaluating the shape functions associated with the coarse side of the face at the interpolation points of the shape functions on the refined side of the face [40]. The idea is conceptually equivalent for edge elements but considerably more complex to implement because DoFs do not represent nodal values but moments on top of geometrical entities. In this work, we follow an approach based on the relation between a Lagrangian *pre-basis*, where moments are trivial to evaluate, and the edge basis, so we can avoid the evaluation of edge moments on the refined cell to compute constraints.

Let  $\mathcal{T}'_h$  be a conforming partition of  $\Omega$  into a set of hexahedra (quadrilateral in 2D) geometrical cells  $K$ .  $\mathcal{T}'_h$  can, e.g., be as simple as a single quadrilateral or hexahedron. Starting from  $\mathcal{T}'_h$ , hierarchical AMR is a multi-step process in which at each step, some cells of the input mesh are marked for refinement. A cell marked for refinement is partitioned into four (2D) or eight (3D) children cells by bisecting all cell edges  $\mathcal{E}_K$ . Let us denote by  $\mathcal{T}_h$  the resulting partition of  $\Omega$ .  $\mathcal{T}_h$  can be thought as a collection of quads (2D) or octrees (3D) where the cells of  $\mathcal{T}'_h$  are the roots of these trees, and children cells branch off their parent cells. The leaf cells in this hierarchy form the mesh in the usual meaning, i.e.,  $\mathcal{T}_h$ . Thus, for every cell  $K \in \mathcal{T}_h$  we can define  $\ell(K)$  as the level of  $K$  in the aforementioned hierarchy, where  $\ell(K) = 0$  for the root cells, and  $\ell(K) = \ell(\text{parent}(K)) + 1$  for any other cell. Clearly, the cells in  $\mathcal{T}_h$  can be at different levels of refinement. Thus, these meshes are *non-conforming*. In order to complete the definition of  $\mathcal{T}_h$ , let us introduce the concept of *hanging* geometrical entities. For every cell  $K \in \mathcal{T}_h$ , consider its set of vertices  $\mathcal{N}_K$ , edges  $\mathcal{E}_K$  and faces  $\mathcal{F}_K$ . We can represent the set of geometrical entities that have lower dimension than the cell by  $\mathcal{G}_K = \mathcal{N}_K \cup \mathcal{E}_K \cup \mathcal{F}_K$ . Its global counterpart is defined as  $\mathcal{G}_{\mathcal{T}} = \cup_{K \in \mathcal{T}_h} \mathcal{G}_K$ . For every geometrical entity  $s \in \mathcal{G}_{\mathcal{T}}$ , we can represent by  $\mathcal{T}_s$  the set of cells  $K \in \mathcal{T}_h$  such that  $s \in \mathcal{G}_K$ . Additionally,  $\tilde{\mathcal{T}}_s$  is defined as the set of cells  $K \in \mathcal{T}_h$  such that  $s \subsetneq s'$  for some  $s' \subset \mathcal{G}_K$ . Roughly speaking, one set contains all the cells where the geometrical entity is found while the other set contains all the cells where the geometrical entity is a strict subset of a coarser geometrical entity. A geometrical entity  $s \in \mathcal{G}_{\mathcal{T}}$  is *hanging* (or *improper*) if and only if there exists at least one neighbouring cell in  $\tilde{\mathcal{T}}_s$ . We represent the set of hanging geometrical entities with  $\mathcal{G}_{\mathcal{T}}^{\text{hg}}$ . The definition of the proposed AMR approach is completed by imposing the so-called 2:1 balance restriction, which is used in mesh adaptive methods by a majority of authors as a reasonable trade-off between performance gain and complexity of implementation [55, 27, 16].

**Definition 5.1.** A  $d$ -tree ( $d = \{2, 3\}$ ) is 2:1  $k$ -balanced if and only if, for any cell  $K \in \mathcal{T}_h$  there is no  $s \in \mathcal{G}_K$  of dimension  $m \in [k, d)$  having non-empty intersection with the closed domain of another finer cell  $K' \in \tilde{\mathcal{T}}_h$  such that  $\ell(K') - \ell(K) > 1$ .

For edge FEs it is enough to consider 1-balance since vertices do not contain any associated DoF. For the sake of clarity, in Figs. 8a and 8b, allowed *hanging* geometrical entities



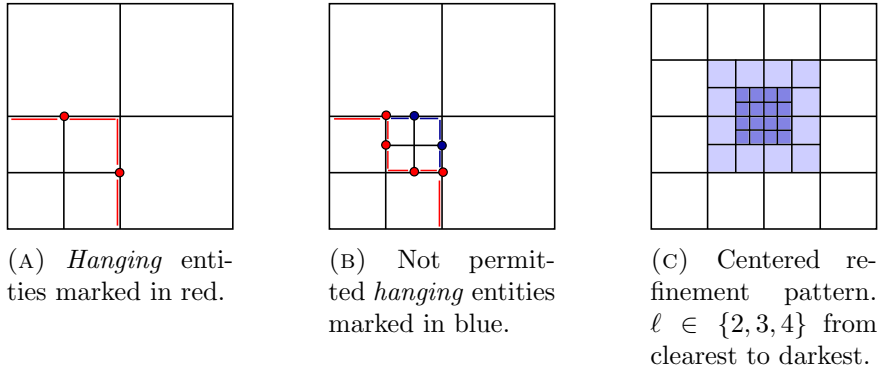


FIGURE 8.  $h$ -adaptive refined (single octree) *non-conforming* meshes.

are depicted in red, whereas in Fig. 8b, not allowed ones are shown in blue. Clearly, the latter mesh is the result of a refinement process that does not accomplish the 2:1 balance, thus not permitted in our AMR approach. Note that in order to enforce the 2:1 balance in the situation depicted in Fig. 8b, one would need to apply additional refinement to some cells with lower values for  $\ell(K)$  until the restriction (5.1) is satisfied.

**5.2. Conformity of the global FE space.** In order to preserve the conformity of the FE space  $\mathcal{N}\mathcal{D}_k(\Omega)$ , we cannot allow an arbitrary value for DoFs placed on top of *hanging* geometrical entities, which will be denoted by *hanging* DoFs. Our approach is to eliminate the *hanging* DoFs of the global system by defining a set of constraints such that curl-conformity is preserved. We propose an algorithm that computes the edge FE constraints by relying on the ones of Lagrangian FEs and the change of basis in Sect. 3.4. This way, one can reuse existing ingredients in a nodal-based AMR code and work already required to define the edge FE shape functions.

The computation of constraints requires some preliminary work at the geometrical level, i.e., independent of the FE space being used. Let us first compute the set of hanging geometrical entities  $\mathcal{G}_{\mathcal{T}}^{\text{hg}}$ . For every  $g \in \mathcal{G}_{\mathcal{T}}^{\text{hg}}$ , let us compute the coarser geometrical entity  $G(g) \in \mathcal{G} \setminus \mathcal{G}_{\mathcal{T}}^{\text{hg}}$  that contains it. Let us assign to every  $g \in \mathcal{G}_{\mathcal{T}}$  one cell such that  $K(g) \in \mathcal{T}_g$ . With this information, for every  $g \in \mathcal{G}_{\mathcal{T}}^{\text{hg}}$ , we can extract the fine cell  $K_h \equiv K(g)$  and the coarse cell  $K_{2h} \equiv K(G(g))$  (see Fig. 9a).

The coarse cell  $K_{2h}$  can be refined (once) to meet the level of refinement of  $K_h$ . (The coarse cell is only refined for the computation of constraints so the original mesh is not affected.) Let us consider its children cells  $\{K_h^s\}_{s=1}^{2^d}$  obtained after isotropic refinement by the procedure exposed in Sect. 5.1, see Fig. 9b. We represent the patch of subcells with  $\tilde{K}_h = \bigcup_{i=1}^{2^d} K_h^s$ . We can determine a subcell index  $s(g)$  such that  $g \in \mathcal{G}_{K_h^{s(g)}}$ ;  $s(g)$  is not unique in general, see Fig. 9b.

**5.2.1. Constraints for Lagrangian FEs.** In this section, we compute the constraints for Lagrangian FE spaces. Using the geometrical information above, given a hanging geometrical entity  $g$ , let us consider the Lagrangian FE spaces  $\mathcal{L}_{\mathbf{k}}(K_{2h})$ ,  $\mathcal{L}_{\mathbf{k}}(\tilde{K}_h)$  and  $\mathcal{L}_{\mathbf{k}}(K_h^s)$  (see Sect. 4 for the definitions, Figs. 10a and 10b for an illustration). The objective is to compute the constraints over DoFs on  $g \in \mathcal{G}_{\mathcal{T}}^{\text{hg}}$  that enforce global continuity. This continuity is enforced by evaluating the coarse cell  $K_{2h}$  shape functions on the fine cell  $K_h$  nodes (DoFs) on  $g$ . Since both cells share the FE order, such set of constraints leads to full continuity across  $g$  [56, Ch. 3]. Let us explain the steps followed to compute these constraints.

Clearly,  $\mathcal{L}_{\mathbf{k}}(K_{2h}) \subset \mathcal{L}_{\mathbf{k}}(\tilde{K}_h)$ , so we can apply the Lagrangian interpolant to inject  $\mathbf{u}^{2h} \in \mathcal{L}_{\mathbf{k}}(K_{2h})$  into  $\mathcal{L}_{\mathbf{k}}(\tilde{K}_h)$ . For this purpose, consider the set of original Lagrangian

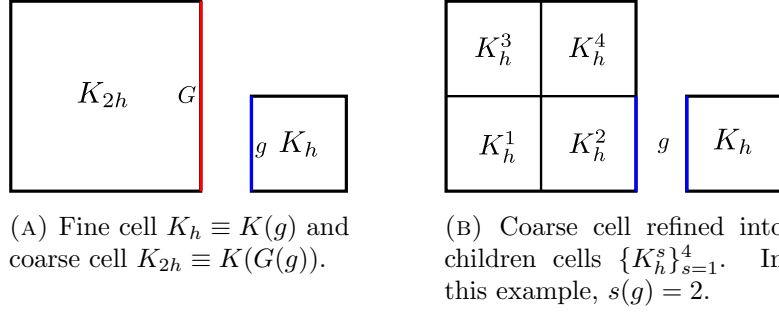


FIGURE 9. The coarse cell  $K_{2h}$  is refined in order to meet the highest level of refinement. Then, the index  $s(g)$  can be determined.

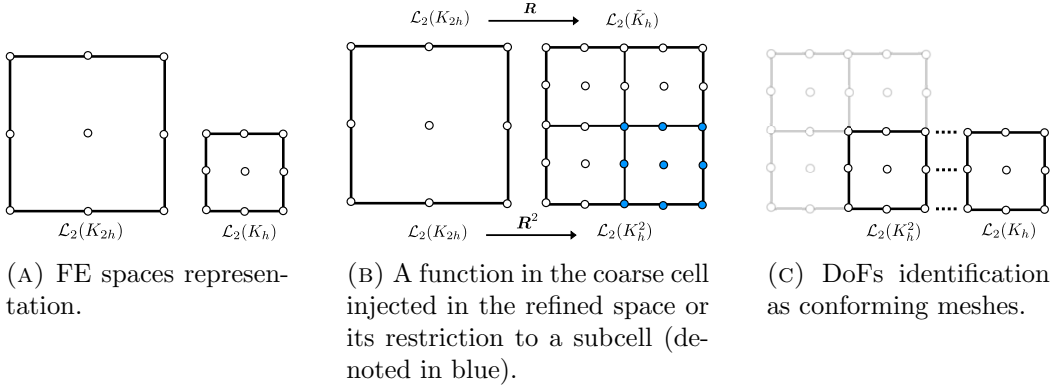


FIGURE 10. Scheme for enforcing continuity at common entities for two cells with different level of refinement with second order Lagrangian FEs.

basis functions  $\{\varphi^j\}_{j=1}^{n_k}$  spanning  $\mathcal{L}_{\mathbf{k}}(K_{2h})$ , which has cardinality  $n_k = \prod_i^d (k_i + 1)$  (see Sect. 3.2 for details). Further, consider the set of moments  $\{\sigma_i\}_{i=1}^{\tilde{n}_k}$  uniquely defining a solution in  $\mathcal{L}_{\mathbf{k}}(\tilde{K}_h)$ , which has cardinality  $\tilde{n}_k = \prod_i^d (2k_i + 1)$  (see Sect. 3.2.1). Given the vector of DoF values of  $\mathbf{u}^{2h}$ , the restriction operator  $\mathbf{R} : \mathcal{L}_{\mathbf{k}}(K_{2h})' \rightarrow \mathcal{L}_{\mathbf{k}}(\tilde{K}_h)'$

$$R_{ij} \doteq \sigma_i(\varphi^j), \quad i = 1, \dots, \tilde{n}_k, \quad j = 1, \dots, n_k,$$

provides the DoFs of the interpolated function. Further, note that given a *hanging* DoF  $u^{h,i}$  on top of  $g$ , it is easy to check that  $R_{ij} = 0$  if DoF  $u^{2h,j}$  is not on  $G(g)$ .

For every subcell  $K_h^s$  one can readily determine the map  $w_s(\cdot)$  such that, given the local cell index  $i \in \{1, \dots, n_k\}$  for moments defined on  $\mathcal{L}_{\mathbf{k}}(K_h^s)'$  ( $s \in \{1, \dots, 2^d\}$ ), returns a global moment index  $i' \in \{1, \dots, \tilde{n}_k\}$  in the space  $\mathcal{L}_{\mathbf{k}}(\tilde{K}_h)'$ . It leads to the subcell restriction operator (of DoF values)  $\mathbf{R}^s : \mathcal{L}_{\mathbf{k}}(K_{2h})' \rightarrow \mathcal{L}_{\mathbf{k}}(K_h^s)'$  defined as

$$R_{ij}^s \doteq R_{w_s(i)j} \quad i = 1, \dots, n_k, \quad j = 1, \dots, n_k.$$

A representation of the action of  $\mathbf{R}$  and  $\mathbf{R}^s$  can be seen in Fig. 10b. Note that they are independent of the level of refinement and the cell in the physical space; they are computed only once at the reference cell.

We can identify every DoF  $i \in \mathcal{L}_{\mathbf{k}}(K_h)'|_g$  with a DoF  $i' \in \mathcal{L}_{\mathbf{k}}(K_h^{s(g)})'|_g$  as for conforming Lagrangian meshes; two local DoF values must be identical if their corresponding nodes are located at the same position (see Fig. 10c). Finally, *the DoF value  $i \in \mathcal{L}_{\mathbf{k}}(K_h)'|_g$  is constrained by the DoF values of the coarse cell through the row  $i'$  of  $\mathbf{R}^{s(g)}$ .*

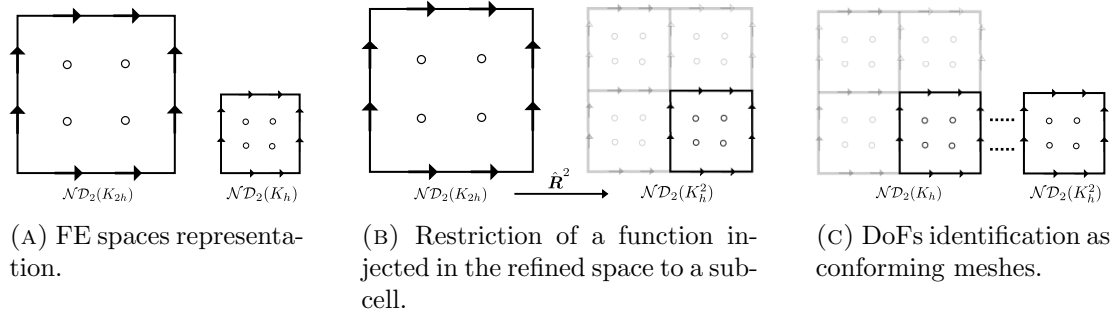


FIGURE 11. Scheme for enforcing continuity at common entities for two cells with different level of refinement with second order edge FEs.

5.2.2. *Constraints for edge FEs.* In order to compute the constraints for edge FEs one could follow a similar procedure as the one above, see Fig. 11 for an illustration. Nevertheless, we propose a different approach for the implementation of restriction operator in edge FEs (see Fig. 11b), which allows us to reuse the restriction operator defined for nodal FEs and avoids the evaluation of edge moments in subcells. Besides, our approach for computing the edge FE constraints is applicable to *any* FE space that relies on a pre-basis of Lagrangian polynomials plus a change of basis (e.g., Raviart-Thomas FEs) without *any* additional implementation effort.

The idea is to build the restriction operator for edge FEs as a composition of operators. We recall the change of basis matrix  $\mathbf{Q} : \mathcal{L}_k(K) \rightarrow \mathcal{N}\mathcal{D}_k(K)$  between Lagrangian and edge FE basis functions (see Sect. 3.4). It leads to the adjoint operator  $\mathbf{Q}^T : \mathcal{N}\mathcal{D}_k(K)' \rightarrow \mathcal{L}_k(K)'$  and its inverse  $\mathbf{Q}^{-T} : \mathcal{L}_k(K)' \rightarrow \mathcal{N}\mathcal{D}_k(K)'$ . As a result, given a subcell  $K_h^s$ , we can define the restriction operator  $\hat{\mathbf{R}}^s \doteq \mathbf{Q}^{-T} \mathbf{R}^s \mathbf{Q}^T : \mathcal{N}\mathcal{D}_k(K_{2h})' \rightarrow \mathcal{N}\mathcal{D}_k(K_h^s)'$  that takes the edge FE DoF values in the coarse cell  $K_{2h}$  and provides the ones of the interpolated function (see (25)) in  $K_h^s$ . Again,  $\hat{\mathbf{R}}^s$  can be computed once at the reference cell. An illustration of the sequence of spaces is shown in Fig. 12.

Given a hanging edge/face  $g \in \mathcal{G}_T^{\text{hg}}$  (vertices do not have associated DoFs for edge FEs), the constraints of its DoFs are computed as follows. We can identify every DoF  $i \in \mathcal{N}\mathcal{D}_k(K_h)'|_g$  with a DoF  $i' \in \mathcal{N}\mathcal{D}_k(K_h^{s(g)})'|_g$  as for conforming meshes (see the equivalence class in Sect. 4.3 and Fig. 11c). As a result, *the DoF value  $i \in \mathcal{N}\mathcal{D}_k(K_h)'|_g$  is constrained by the DoF values of the coarse cell through the row  $i'$  of  $\hat{\mathbf{R}}^{s(g)}$ .*

5.2.3. *Global assembly for non-conforming meshes.* The definition of the constraints is used in the assembly of the linear system as follows. First of all, let us write the problem (7) in algebraic form. The solution  $\mathbf{u} \in \mathcal{N}\mathcal{D}_k(\Omega)$  is expanded by  $\{\phi^a\}_{a=1}^n$ . The elemental matrices are defined as  $\mathcal{M}_{ij} = \int_K \beta \phi^j \cdot \phi^i$  and  $\mathcal{K}_{ij} = \int_K (\alpha \nabla \times \phi^j) \cdot (\nabla \times \phi^i)$ , whereas the elemental right-hand side is the discrete vector  $\mathbf{f}_i = \int_K \mathbf{f}_K \cdot \phi^i$  for  $i, j \in \{1, \dots, n\}$ . The usual assembly for every  $K \in \mathcal{T}_h$  is performed to obtain the global matrix and array, hence the algebraic form reads  $\mathcal{A}\mathbf{u} = \mathbf{f}$ , where  $\mathcal{A} = \mathcal{K} + \mathcal{M}$ . Let us now distinguish between the set of *conforming* DoFs  $\mathbf{u}_c$  and the set of *non-conforming* DoFs  $\mathbf{u}_{nc}$ , whose cardinalities are  $n_c + n_{nc} = n$ . One can write the constraints that  $\mathbf{u}_{nc}$  must satisfy in compact form,  $\mathbf{u}_{nc} = \mathbf{G}\mathbf{u}_c$ , where the entries of  $\mathbf{G}$  (of dimension  $n_{nc} \times n_c$ ) are given by the described procedure for obtaining the *constraining* factors between the *constrained* and the *constraining* part. Then, the solution  $\mathbf{u}_h$ , expressed in terms of both types of DoFs can be written as

$$(26) \quad \mathbf{u}_h = \begin{bmatrix} \mathbf{u}_c \\ \mathbf{u}_{nc} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{G} \end{bmatrix} \mathbf{u}_c = \mathbf{P}\mathbf{u}_c,$$

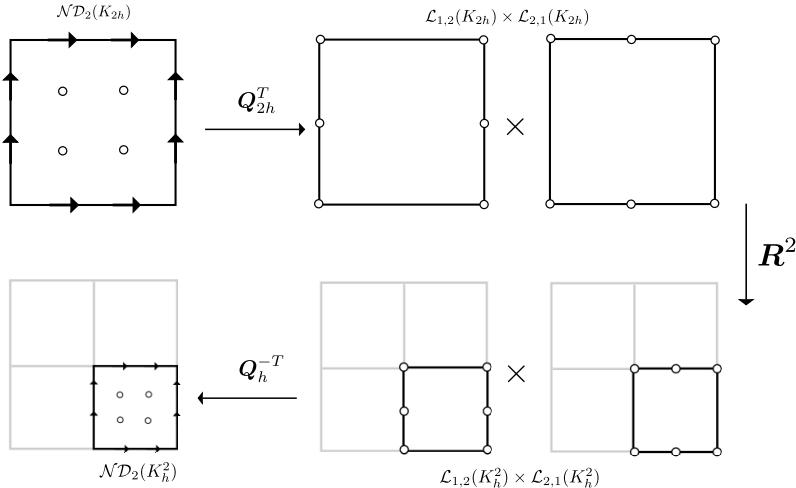


FIGURE 12. Sequence of spaces in order to compute the restriction operator entries for  $K_h^2$  with second order edge FEs. Representation of edge and inner DoFs by arrows and nodes, resp.

and we can write the constrained problem only in terms of the *conforming* DoFs  $\mathbf{u}_c$  as

$$(27) \quad \bar{\mathbf{A}}\mathbf{u}_c = \bar{\mathbf{f}},$$

where  $\bar{\mathbf{A}} = \mathbf{P}^T \mathbf{A} \mathbf{P}$  and  $\bar{\mathbf{f}} = \mathbf{P}^T \mathbf{f}$ . Our implementation directly builds the constrained operator  $\bar{\mathbf{A}}$  by *locally* applying the constraints to eliminate  $\mathbf{u}_{nc}$  DoFs using the constraints given by  $\mathbf{G}$  in the assembly process (see [57] for further details). Once the solution for *conforming* DoFs is obtained, *hanging* DoF values are recovered by (26) as a postprocess.

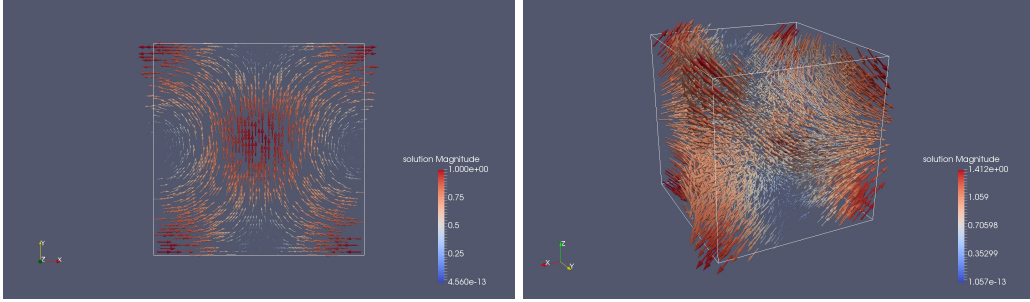
## 6. NUMERICAL EXPERIMENTS

In this section, we test the implementation of edge elements and  $H(\text{curl})$ -conforming spaces in FEMPAR [51], a general purpose, parallel scientific software for the FE simulation of complex multiphysics problems governed by PDEs written in Fortran200X following object-oriented principles. It supports several computing and programming environments, such as, e.g., multi-threading via OpenMP for moderate scale problems and hybrid MPI/OpenMP for HPC clusters. See [50] for a thorough coverage of the software architecture of FEMPAR, which is distributed as open source software under the terms of the GNU GPLv3 license. Regarding the content of this document, FEMPAR supports arbitrary order edge FEs on both hexahedra and tetrahedra, on either structured and unstructured conforming meshes, and also mesh generation by adaptation using hierarchically refined octree-based meshes. In order to test our implementation, we will compare both theoretical and experimental convergence rates. Generally, log-log plots of the computed error in the considered norms ( $L_2$ -norm or  $H(\text{curl})$ -norm) against different values of  $h$  or number of DoFs will be shown. Let us first cite some *a priori* error estimates for edge FEs of the first kind. In the  $H(\text{curl})$ -norm, we find the following optimal estimate, presented in [58]:

**Theorem 6.1.** *If  $\mathcal{T}_h$  is a regular family of triangulations on  $\Omega$  for  $h > 0$ , then there exists a constant  $C > 0$  such that*

$$(28) \quad \|\mathbf{v} - \pi_h^k \mathbf{v}\|_{H(\text{curl})} \leq Ch^{\min\{r,k\}} \|\mathbf{v}\|_{H^r(\text{curl})},$$

for all  $\mathbf{v} \in H^r(\text{curl})$ , where  $r > \frac{1}{2}$  determines the regularity of the function  $\mathbf{v}$ , which is valid for  $H(\text{curl})$ -conforming elements. If the function  $\mathbf{v}$  is smooth enough to have bounded derivatives such that  $s > k$ , then the estimate states that superlinear convergence can be achieved as we increase the polynomial order  $k$ .



(A) Analytical function  $\mathbf{u}^*$  in the 2D case. (B) Analytical function  $\mathbf{u}^*$  in the 3D case.

FIGURE 13. Analytical functions.

For the  $L_2(\Omega)$  approximation, the following convergence rate can be expected [1].

**Theorem 6.2.** *If  $\mathcal{T}_h$  is a regular family of triangulations on  $\Omega$  for  $h > 0$ , then there exists a constant  $C > 0$  such that*

$$(29) \quad \|\mathbf{v} - \pi_h^k \mathbf{v}\|_{L_2(\Omega)} \leq Ch^k \|\mathbf{v}\|_{H^k(\Omega)}.$$

In general, we choose analytical functions that do not belong to the FE space. We show convergence plots with respect to the mesh size for uniform mesh refinement or the number of DoFs for  $h$ -adaptive mesh refinement. In all cases, we will solve the reference problem (5) with homogeneous, scalar parameters  $\alpha = \beta = 1$ . Unless otherwise stated, the results are computed in the unit box domain  $\Omega := [0, 1]^d$ . We utilize the method of *manufactured* solutions, i.e., to plug an analytical function  $\mathbf{u}^*$  in the exact form of the problem and obtain the corresponding source term that verifies the equation. Then, one can solve the problem for the unknown  $\mathbf{u}$  so that the computed solution must converge to the exact solution with mesh refinement.

**6.1. Uniform mesh refinement.** In this section, the experimental rate of convergence will be numerically included in the legend as the value for the slope computed with the two last available data points in each plot. We will make use of the following analytical function and source term for all the 2D cases presented in this section:

$$\mathbf{u}^* = \begin{bmatrix} \cos(\pi x_1) \cos(\pi x_2) \\ \sin(\pi x_1) \sin(\pi x_2) \end{bmatrix}, \quad \mathbf{f} = (2\pi^2 + 1)\mathbf{u}^*,$$

whereas in 3D cases the analytical function and corresponding source term are given by

$$\mathbf{u}^* = \begin{bmatrix} \cos(\pi x_1) \cos(\pi x_2) \\ \sin(\pi x_2) \sin(\pi x_3) \\ \cos(\pi x_1) \cos(\pi x_3) \end{bmatrix}, \quad \mathbf{f} = (\pi^2 + 1)\mathbf{u}^* + \pi^2 \begin{bmatrix} \sin(\pi x_1) \sin(\pi x_3) \\ \sin(\pi x_1) \sin(\pi x_2) \\ \cos(\pi x_2) \cos(\pi x_3) \end{bmatrix}.$$

Dirichlet boundary conditions are strongly imposed over the entire boundary, i.e.,  $\partial\Omega_D := \partial\Omega$ , where we enforce the tangent component of the analytical function  $\mathbf{u}_\tau^*$ .

**6.1.1. Hexahedral meshes.** Structured simulations are performed with a structured mesh on the unit box domain with the same number of elements  $n_K$  in each direction. Figs. 14 and 15 show the convergence rates with the order of the element  $k$  for 2D and 3D cases, resp. In all cases, the expected convergence ratio (see (28) and (29)) is achieved, presented up to  $k = 6$  in 2D and  $k = 4$  in 3D.

**6.1.2. Tetrahedral meshes.** Edge FEs are tested in this section with tetrahedral mesh partitions of the domain  $\Omega$ . Consider a family of tetrahedral meshes  $\{\mathcal{T}_m\}_{m=1}^M$ , obtained by structured, hexahedral meshes plus triangulation of hexahedral cells. Here the element size denotes the usual definition  $h = \max_{K \in \mathcal{T}_m} h_K$ , being  $h_K$  the diameter of the largest

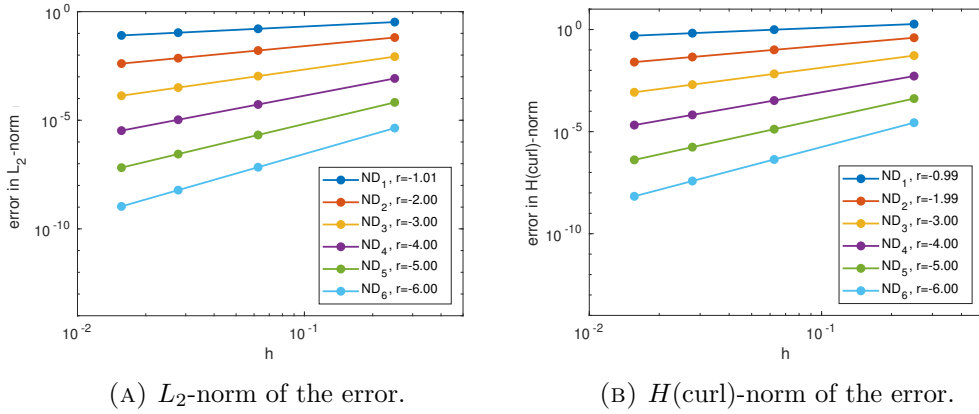


FIGURE 14. Error norms for different orders 2D hexahedral edge FEs.

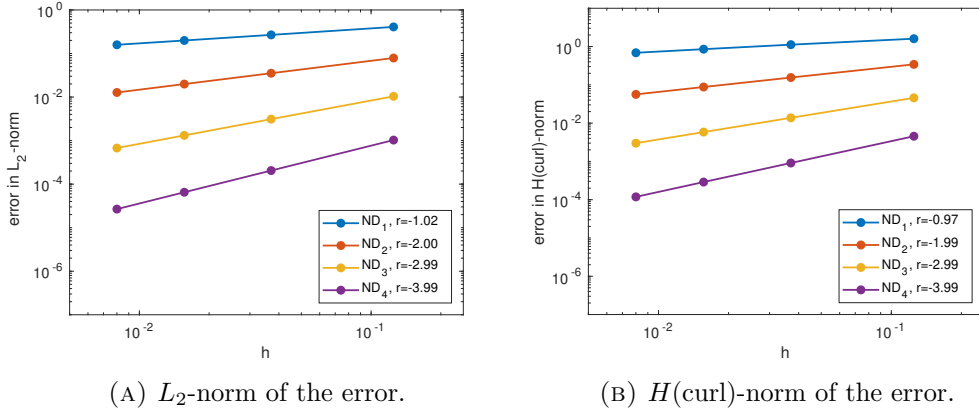


FIGURE 15. Error norms for different orders 3D hexahedral edge FEs.

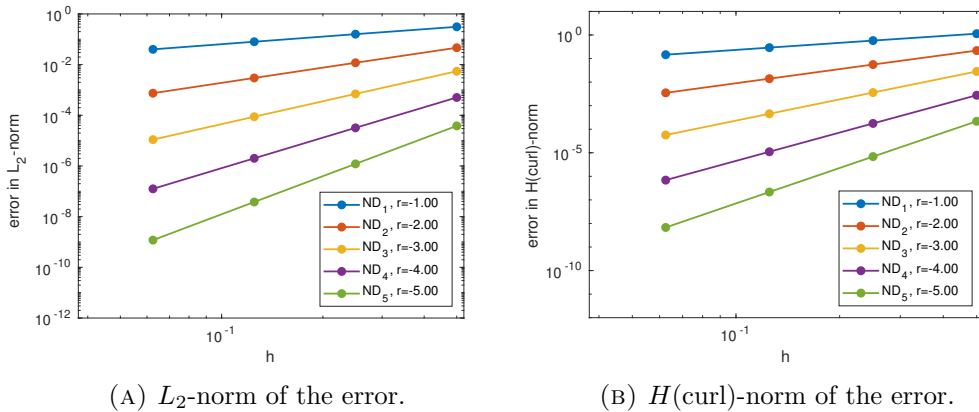


FIGURE 16. Error norms for different orders 2D tetrahedral edge FEs.

circumference or sphere containing  $K$  for 2D and 3D, resp. Convergence results in Figs. 16 and 17 are computed with this family of meshes. In all cases, computed convergence ratios are consistent with the estimates in (28) and (29).

**6.2.  $h$ -adaptive mesh refinement.** In this section, we present results for meshes obtained by adaptive refinement from an initial structured, hexahedral *conforming* mesh.

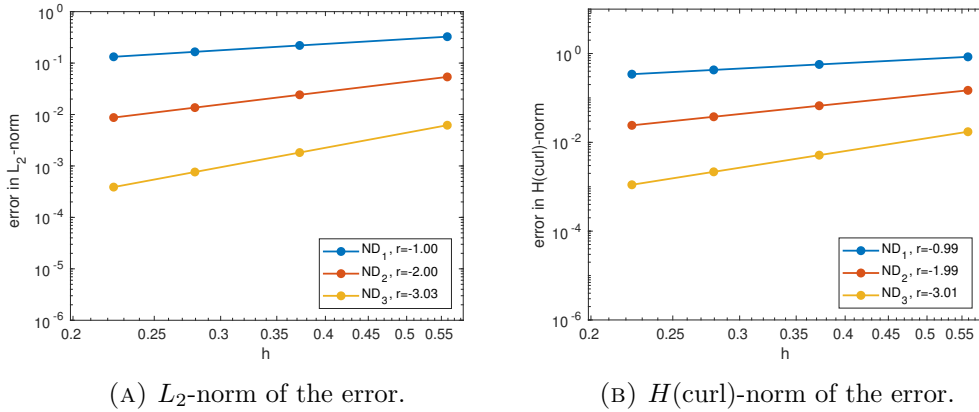


FIGURE 17. Error norms for different orders 3D tetrahedral edge FEs.

The refinement process follows the usual steps: 1) solve the problem on a given mesh, 2) compute an estimation of the local error contribution at every cell using the solution computed at the previous step, 3) mark the cells with more error for refinement, and 4) refine the mesh and restart the process if the stopping criterion is not fulfilled. For comparison purposes, we will also consider a uniform mesh refinement. Then, log-log convergence plots for the ( $L_2$  or  $H(\text{curl})$ -) error against the number of free DoFs involved in the simulation is presented for analytical solutions that contain a singularity in a re-entrant corner. We provide numerical results for problems where the analytical solution is known *a priori*, thus we can make use of the true error. In any case, the implementation of the *a posteriori* error estimators enumerated in Sect. 1 does not pose any extra implementation challenge to what it is already exposed in the paper.

Let us consider a L-shaped domain,  $\Omega = [-1, 1]^2 \setminus ([0, 1] \times [-1, 0])$  with Dirichlet boundary conditions imposed over the entire domain boundary. The source term  $\mathbf{f}$  is such that the solution in polar coordinates  $(r, \theta)$  is

$$(30) \quad \mathbf{u} = \nabla \left( r^{\frac{2n}{3}} \sin \left( \frac{2n}{3} \theta \right) \right).$$

The chosen analytical function  $\mathbf{u}$  has a singular behaviour at the origin of coordinates for  $n = 1$ , which prevents the function to be in  $H^1(\Omega)$ . Larger values of  $n$  lead to smoother solutions. The regularity of the solution is well studied [59], and theoretical convergence rates (28) are bounded by  $\min(2n/3, k)$ .

In Figs. 19 and 20, the theoretical convergence rates are achieved for every order of converge. In the case  $n = 1$ , all FE orders lead to the same convergence rate since  $k > 2/3$ . For a smoother solution, corresponding to  $n = 4$ , solutions converge to the expected order  $\min(k, 8/3)$ . Next, we analyze the error with adaptive refinement. The refinement process is such that, at every iterate, the 5% of cells with higher local contribution to the  $L_2$ -error are marked for refinement. The fraction of cells to be refined is intentionally chosen to be small since we aim to obtain a localized refinement around the singularity. Figs. 18a and 18c show the initial mesh and the final mesh after 18 refinement steps, resp. In Fig. 21, we plot the ( $L_2$ - or  $H(\text{curl})$ -) error against the number of DoFs. We show two plots for each order, namely uniform refinement (solid line) and adaptive refinement (solid line with circles). In all cases, better efficiency is achieved by adaptive meshes, i.e., less error for a given number of DoFs.

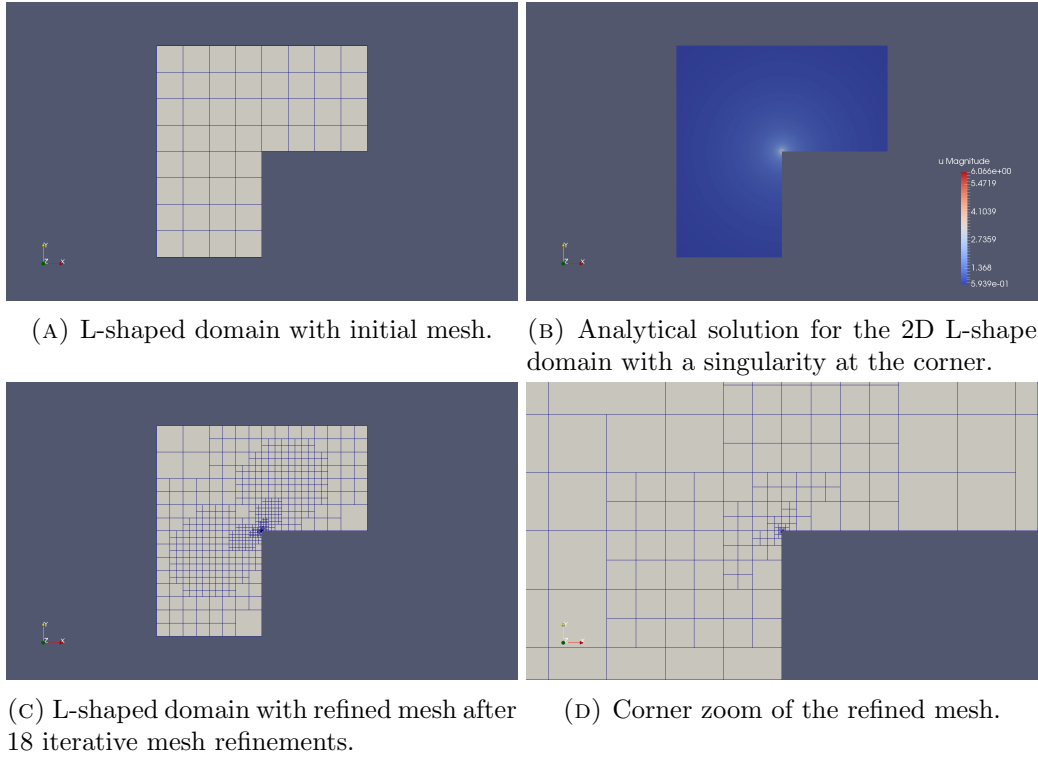


FIGURE 18. fichera 2D problem. At each mesh refinement step the 5% of cells with highest local  $L_2$ -error are refined.

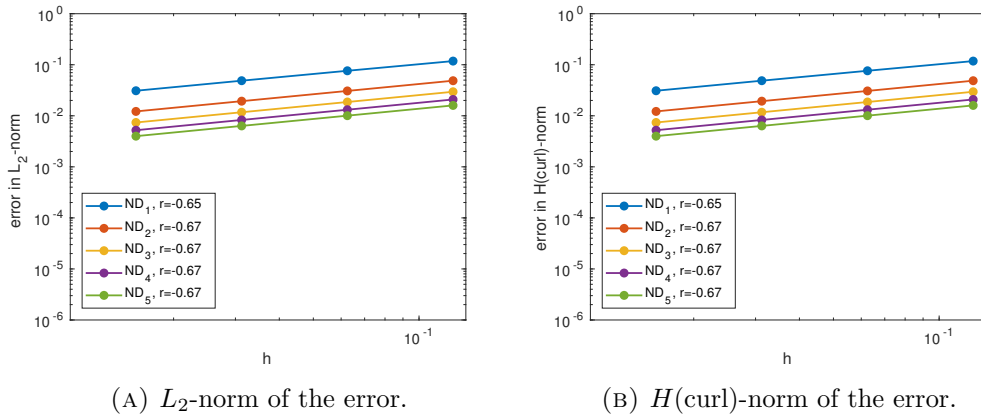


FIGURE 19. Error norms for the fichera 2D problem with uniform refinement for  $n = 1$ .

Let us now consider the Fichera domain  $\Omega = [-1, 1]^3 \setminus [-1, 0]^3$ . The source term  $\mathbf{f}$  is such that the solution is

$$(31) \quad \mathbf{u} = \nabla \left( r^{\frac{2}{3}} \sin \left( \frac{2t}{3} \right) \right) \quad t = \arccos \left( \frac{xyz}{r} \right),$$

where  $r$  is the radius in 3D polar coordinates. The analytical solution has a singular behaviour near the origin and again  $\mathbf{u} \notin H^1(\Omega)$ . We follow the same analysis to determine the efficiency of the  $h$ -adaptive scheme.

Fig. 22 shows the error vs. the number of DoFs of the  $h$ -adaptive refinement process. In Fig. 23, we illustrate the refinement process that takes place from an initial structured



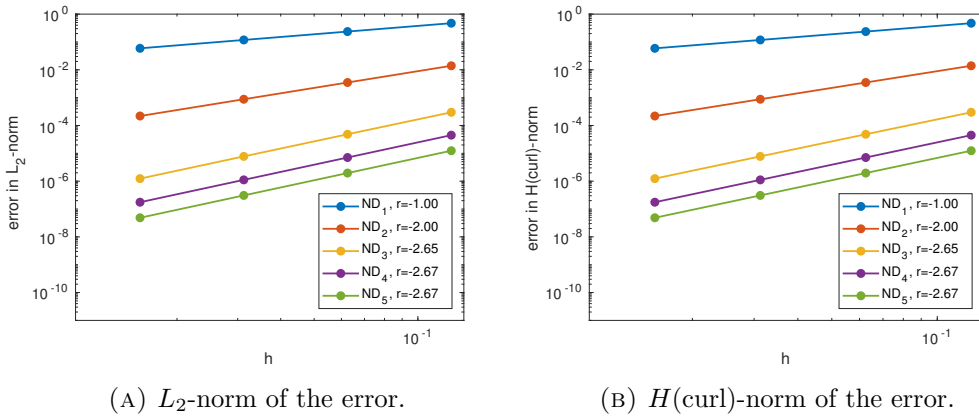


FIGURE 20. Error norms for the fichera 2D problem with uniform refinement for  $n = 4$ .

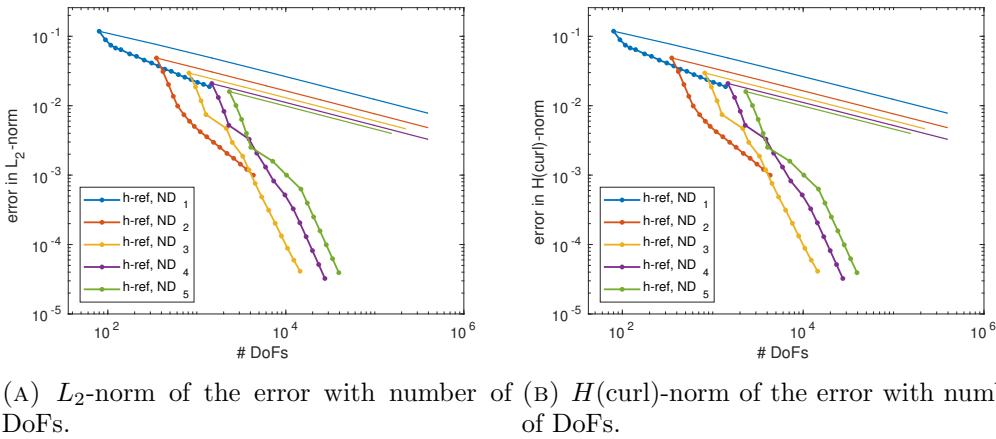
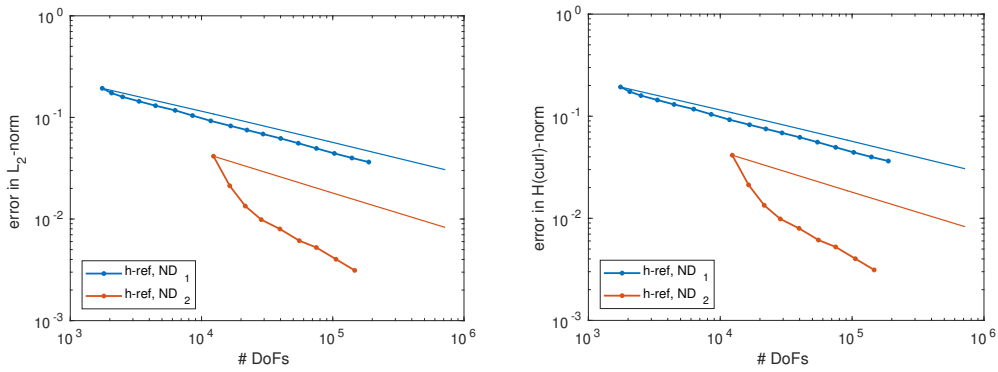


FIGURE 21. Error norms for the fichera 2D with  $n = 1$  and adaptive refinement, which at every iterate marks for refinement the 5% of cells that show the highest local cell  $L_2$ -error. Lines without markers show the error convergence with uniform refinement process for every FE order.

mesh composed of  $8^3$  elements (see Fig. 23a). Clearly, the closer a cell  $K$  is to the corner (see Fig. 23b), the higher the final  $\ell(K)$ . However, Fig. 23b also shows that the refinement is not as localized as in the 2D case, with a greater portion of the domain with higher levels of refinement. This fact has a clear impact on the efficiency achieved by the adaptive refinement for first order edge FEs, where the efficiency gain is mild. On the other hand, the gain for second order FEs is noticeably higher. Finally, Figs. 23c and 23d show the refined mesh after 12 refinement iterates and a zoom at the corner with the singularity.

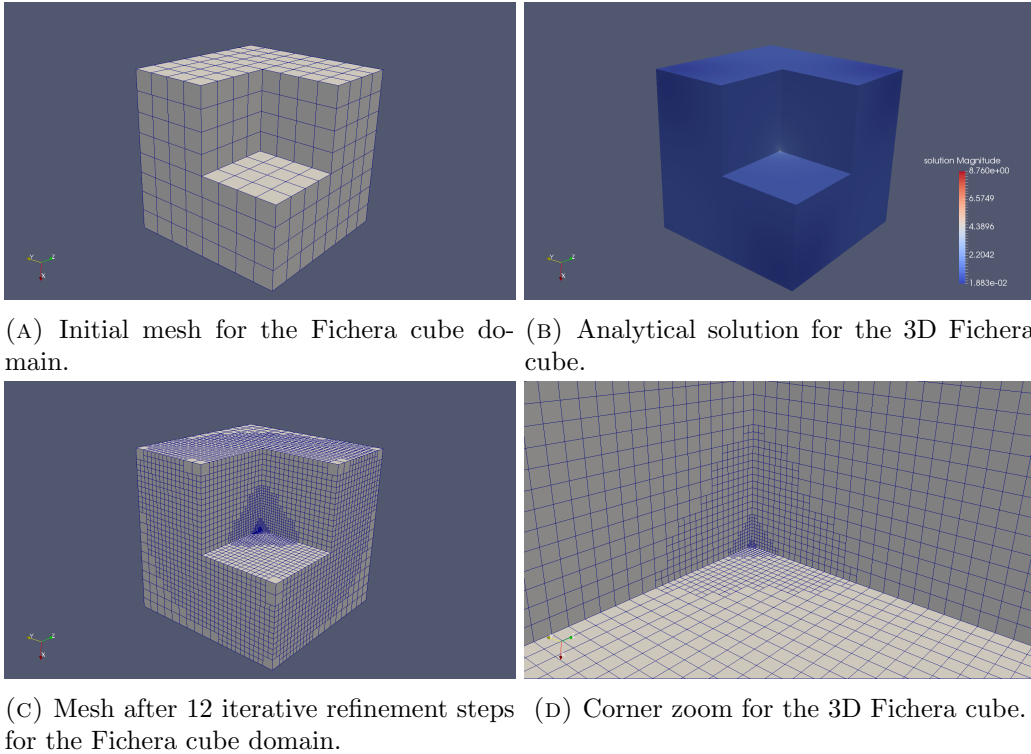
## 7. CONCLUSIONS

In this work, we have covered in detail a general implementation of  $p$ -adaptive and  $h$ -adaptive tetrahedral and hexahedral edge FE methods. We have implemented pre-bases that span the local FE spaces (anisotropic polynomials) which combined with a change of basis automatically provide the shape functions bases. It leads to a general arbitrary order implementation, confronted with hard-coded implementations that preclude high order methods. In order to guarantee the tangent continuity of Piola-mapped elements, special care must be taken with the orientation of the cell geometrical entities. In the



(A)  $L_2$ -norm of the error with number of DoFs. (B)  $H(\text{curl})$ -norm of the error with number of DoFs.

FIGURE 22. Error norms for the Fichera 3D problem with uniform (straight line) and adaptive refinement. 5% of cells that have the highest local  $L_2$ -error marked for refinement at every refinement step.



(A) Initial mesh for the Fichera cube domain. (B) Analytical solution for the 3D Fichera cube. (C) Mesh after 12 iterative refinement steps for the Fichera cube domain. (D) Corner zoom for the 3D Fichera cube.

FIGURE 23. Adaptive meshes for the Fichera 3D problem. Cells with highest 5% Local  $L_2$ -error(K) are refined at each refinement step.

implementation, we require the local numbering of nodes within every element to rely on sorted global indices, i.e., oriented meshes. This manner, we automatically satisfy consistency in every geometrical entity shared by two or more FEs. Finally, we propose an original approach to implement global curl-conforming FE space on hierarchically refined octree-based *non-conforming* meshes. The strategy, which is straightforwardly extensible to any FE based on polynomial spaces, is based on the original Lagrangian constraints and the interplay between the sets of Lagrangian and edge basis functions. To obtain

every constraint, a sequence of spaces can be built so as we can avoid the evaluation of moments in the edge FE space. A detailed set of numerical experiments served to test the implementation, where we show agreement between theoretical and numerical rates of convergence. The proposed approach has been implemented (for first kind edge  $H(\text{curl})$ -conforming FE) in **FEMPAR**, a scientific software for the simulation of problems governed by PDEs.

We note that this framework can be extended to other polynomial-based FEs. Customizable ingredients are the original pre-basis of polynomials, the moments, the geometrical mapping, and the equivalence class of DoFs. However, the change of basis approach to obtain the corresponding shape functions and the enforcement of continuity for non-conforming meshes is identical. In fact, the same machinery has already been used in **FEMPAR** to implement Raviart-Thomas FEs and can straightforwardly be used to implement Brezzi-Douglas-Marini FEs [41], second kind edge FEs [42], or recent divergence-free FEs [43].

We believe that the comprehensive description of all the implementation issues behind edge FE method provided herein will be of high value for other researchers and developers that have to purport similar developments and increase their penetration in the computational mechanics community.

#### REFERENCES

- [1] J.C. Nédélec. Mixed finite elements in  $R^3$ . *Numer. Math.*, 35:315–341, 1980.
- [2] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford Science Publications, 2003.
- [3] G. Mur. Edge elements, their advantages and their disadvantages. *IEEE Transactions on Magnetics*, 30(5):3552–3557, 1994.
- [4] S. Badia and R. Codina. A Nodal-based Finite Element Approximation of the Maxwell Problem Suitable for Singular Solutions. *SIAM Journal on Numerical Analysis*, 50(2):398–417, 2012.
- [5] M. Olm, S. Badia, and A.F. Martín. Simulation of high temperature superconductors and experimental validation. *Computer Physics Communications*, in press, 2018.
- [6] Y. L. Li, S. Sun, Q. I. Dai, and W. C. Chew. Vectorial Solution to Double Curl Equation With Generalized Coulomb Gauge for Magnetostatic Problems. *IEEE Transactions on Magnetics*, 51(8):1–6, 2015.
- [7] I. Perugia and D. Schötzau. The  $hp$ -local discontinuous Galerkin method for low-frequency time-harmonic Maxwell equations. *Mathematics of Computation*, 72:1179–1214, 2003.
- [8] S. Badia and R. Codina. A combined nodal continuous–discontinuous finite element formulation for the Maxwell problem. *Applied Mathematics and Computation*, 218(8):4276 – 4294, 2011.
- [9] R. Vazquez and A. Buffa. Isogeometric analysis for electromagnetic problems. *IEEE Transactions on Magnetics*, 46(8):3305–3308, Aug 2010.
- [10] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences*, 23(01):199–214, 2013.
- [11] L. Beirão da Veiga, F. Brezzi, L.D. Marini, and A. Russo.  $H(\text{div})$  and  $H(\text{curl})$ -conforming VEM. *Numerische Mathematik*, 133:303–332, 2016.
- [12] J. Gopalakrishnan, L.E. García-Castillo, and L.F. Demkowicz. Nédélec spaces in affine coordinates. *Computers & Mathematics with Applications*, 49(7):1285–1294, 2005.
- [13] M. Bonazzoli and F. Rapetti. High-order finite elements in numerical electromagnetism: degrees of freedom and generators in duality. *Numerical Algorithms*, 74(1):111–136, 2017.
- [14] M. Bonazzoli, V. Dolean, F. Hecht, and F. Rapetti. An example of explicit implementation strategy and preconditioning for the high order edge finite elements applied to the time-harmonic Maxwell's equations. *Computers & Mathematics with Applications*, 75(5):1498 – 1514, 2018.
- [15] F. Fuentes, B. Keith, L. Demkowicz, and S. Nagaraj. Orientation embedded high order shape functions for the exact sequence elements of all shapes. *Computers & Mathematics with Applications*, 70(4):353 – 458, 2015.
- [16] L. Demkowicz. *Computing with  $hp$ -adaptive finite elements: Volume I: One and Two dimensional elliptic and Maxwell problems*. Chapman & Hall/CRC Press, 2006.
- [17] L. Demkowicz, J. Kurtz, D. Pardo, M. Paszanski, W. Rachowicz, and A. Zdunek. *Computing with  $hp$ -adaptive finite elements: Volume II: Three Dimensional Elliptic and Maxwell Problems with applications*. Chapman & Hall/CRC Press, 2007.

- [18] A. Amor-Martin, L. E. Garcia-Castillo, and D. Daniel Garcia-Doñoro. Second-order Nédélec curl-conforming prismatic element for computational electromagnetics. *IEEE Transactions on Antennas and Propagation*, 64(10):4384–4395, 2016.
- [19] L. E. Garcia-Castillo, A. J. Ruiz-Genoves, I. Gomez-Revuelto, M. Salazar-Palma, and T. K. Sarkar. Third-order Nédélec curl-conforming finite element. *IEEE Transactions on Magnetics*, 38(5):2370–2372, 2002.
- [20] A. Schneebeli. An  $H(\text{curl};\Omega)$ -conforming FEM: Nédélec elements of first type. Technical report, 2003.
- [21] M. Rognes, R. Kirby, and A. Logg. Efficient Assembly of  $H(\text{div})$  and  $H(\text{curl})$  Conforming Finite Elements. *SIAM J. Sci. Comput.*, 31(6):4130–4151, 2009.
- [22] R. S. Falk, P. Gatto, and P. Monk. Hexahedral  $H(\text{div})$  and  $H(\text{curl})$  finite elements. *ESAIM: M2AN*, 45(1):115–143, 2011.
- [23] S. Badia, F. Verdugo, and A.F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533 – 553, 2018.
- [24] I. Anjam and J. Valdman. Fast MATLAB assembly of FEM matrices in 2D and 3D: Edge elements. *Applied Mathematics and Computation*, 267:252 – 263, 2015.
- [25] L. E. García Castillo and M. Salazar Palma. Second order Nédélec tetrahedral element for computational electromagnetics. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 13(3):261–287.
- [26] R. Agelek, M. Anderson, W. Bangerth, and W. Barth. On orienting edges of unstructured two- and three-dimensional meshes. *ACM Transactions on Mathematical Software*, 44(1):Article No. 5, 2017.
- [27] C. Burstedde, L. Wilcox, and O. Ghattas. **p4est**: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM J. Sci. Comput.*, 33(3):1103–1133, 2011.
- [28] M. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, and G. N. Wells. The FEniCS Project Version 1.5. *Archive of Numerical Software*, 3(100), 2015.
- [29] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II-A general-purpose object-oriented finite element library. *ACM Trans. Math. Softw.*, 33(4), 2007.
- [30] W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin, and D. Wells. The deal.II Library, Version 8.4. *Journal of Numerical Mathematics*, 24, 2016.
- [31] F. Hecht. New development in FreeFem++. *Journal of Numerical Mathematics*, 20(3-4):251–265, 2012.
- [32] MFEM – a free, lightweight, scalable C++ library for finite element methods. <http://mfem.org/>.
- [33] Joachim Schöberl. Netgen an advancing front 2d/3d-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1(1):41–52, Jul 1997.
- [34] Netgen/NGSolve. <http://ngsolve.org/>.
- [35] J.N. Shadid, R.P. Pawlowski, E.C. Cyr, R.S. Tuminaro, L. Chacón, and P.D. Weber. Scalable implicit incompressible resistive MHD with stabilized FE and fully-coupled Newton–Krylov-AMG. *Computer Methods in Applied Mechanics and Engineering*, 304:1–25, 2016.
- [36] R. Ortigosa and A. J. Gil. A new framework for large strain electromechanics based on convex multi-variable strain energies: Finite Element discretisation and computational implementation. *Computer Methods in Applied Mechanics and Engineering*, 302:329–360, 2016.
- [37] G. Wang, S. Wang, N. Duan, Y. Huangfu, H. Zhang, W. Xu, and J. Qiu. Extended Finite-Element Method for Electric Field Analysis of Insulating Plate With Crack. *IEEE Transactions on Magnetics*, 51(3):1–4, 2015.
- [38] M. Bergot and M. Duruflé. High-order optimal edge elements for pyramids, prisms and hexahedra. *Journal of Computational Physics*, 232(1):189–213, 2013.
- [39] M. Bergot and P. Lacoste. Generation of higher-order polynomial basis of Nédélec  $H(\text{curl})$  finite elements for Maxwell’s equations. *Journal of Computational and Applied Mathematics*, 234(6):1937–1944, 2010.
- [40] W. Bangerth and O. Kayser-Herold. Data structures and requirements for hp finite element software. *ACM Trans. Math. Softw.*, 36(1):4:1–4:31, 2009.
- [41] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*. Springer-Verlag, 1991.
- [42] J.C. Nédélec. A new family of mixed finite elements in  $R^3$ . *Numer. Math.*, 50:57–81, 1986.
- [43] M. Neilan and D. Sap. Stokes elements on cubic meshes yielding divergence-free approximations. *Calcolo*, 53(3):1–21, 2015.
- [44] R. Beck, R. Hiptmair, R.H.W. Hoppe, and B. Wohlmuth. Residual based a posteriori error estimators for eddy current computation. *ESAIM: M2AN*, 34(1):159–182, 2000.
- [45] J. Schöberl. A posteriori error estimates for Maxwell equations. *Mathematics of Computation*, 77(262):633–649, 2008.
- [46] S. Nicaise and E. Creusé. A posteriori error estimation for the heterogeneous Maxwell equations on isotropic and anisotropic meshes. *CALCOLO*, 40(4):249–271, 2003.

- [47] R. Beck, R. Hiptmair, and B. Wohlmuth. Hierarchical error estimator for eddy current computation. In *In ENUMATH99: Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications*, pages 110–120. World Scientific, 1999.
- [48] D. Braess and J. Schöberl. Equilibrated residual error estimator for edge elements. *Mathematics of Computation*, 77(262):651–672, 2008.
- [49] S. Nicaise. On Zienkiewicz–Zhu error estimators for Maxwell’s equations. *Comptes Rendus Mathématique*, 340(9):697 – 702, 2005.
- [50] S. Badia, A.F. Martín, and J. Principe. FEMPAR: An object-oriented parallel finite element framework. *Arch. Comput. Methods Eng.*, 25(2):195–271, 2018.
- [51] S. Badia, A.F. Martín, and J. Principe. FEMPAR Web page. <http://www.fempar.org>, 2018.
- [52] D.N. Arnold, D. Boffi, and R.S. Falk. Approximation by quadrilateral finite elements. *Mathematics of computation*, 71(239):909–922, 2002.
- [53] S. Badia, F. Verdugo, and A.F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533 – 553, 2018.
- [54] G. Cohen and S. Pernet. *Definition of Different Types of Finite Elements*, pages 39–93. Springer Netherlands, Dordrecht, 2017.
- [55] T. Tu, D. R. O’Hallaron, and O. Ghattas. Scalable parallel octree meshing for terascale applications. In *Proceedings of the ACM/IEEE SC 2005 Conference, Supercomputing, 2005*, pages 1–15, 2005.
- [56] P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods*. Chapman & Hall/CRC Press, 2003.
- [57] S. Badia, A.F. Martín, E. Neiva, and F. Verdugo. A generic finite element framework on parallel tree-based adaptive meshes. *Submitted for publication*, 2019.
- [58] A. Alonso and A. Valli. An optimal domain decomposition preconditioner for low-frequency time-harmonic maxwell equations. *Math. Comput.*, 68(226):607–631, 1999.
- [59] S. Nicaise. Edge Elements on anisotropic meshes and approximation of the Maxwell Equations. *SIAM Journal of Numerical Analysis*, 39(3):784–816, 2001.