

# CONSTRUCTION OF H-REFINED CONTINUOUS FINITE ELEMENT SPACES WITH ARBITRARY HANGING NODE CONFIGURATIONS AND APPLICATIONS TO MULTIGRID ALGORITHMS\*

EUGENIO AULISA<sup>†</sup>, GIACOMO CAPODAGLIO<sup>‡</sup>, AND GUOYI KE<sup>§</sup>

**Abstract.** We present a novel approach for the construction of basis functions to be employed in selective or adaptive h-refined finite-element applications with arbitrary-level hanging node configurations. Our analysis is not restricted to 1-irregular meshes, as it is usually done in the literature, allowing our results to be applicable to a broader class of local refinement strategies. The proposed method does not require the solution of any linear system to obtain the constraints necessary to enforce continuity of the basis functions and it can be easily implemented. A mathematical analysis is carried out to prove that the proposed basis functions are continuous and linearly independent. Finite-element spaces are then defined as the spanning sets of such functions, and the implementation of a multigrid algorithm built on these spaces is discussed. A spectral analysis of the multigrid algorithm highlights superior convergence properties of the proposed method over existing strategies based on a local smoothing procedure. Finally, linear and nonlinear numerical examples are tested to show the robustness and versatility of the multigrid algorithm.

**Key words.** selective h-refinement, arbitrary-level hanging nodes, multigrid methods, finite element method

**AMS subject classifications.** 65N55, 65N30, 65N22, 65F08

**DOI.** 10.1137/18M1175409

**1. Introduction.** In many scientific applications, the problem of obtaining an accurate solution in a timely manner is of major relevance. In a finite-element setting, this could be done in a variety of ways. For instance, local refinement of the finite-element grid allows introducing new degrees of freedom only in the areas of the domain where a more accurate solution is sought. In this way, the addition of extra degrees of freedom on the entire domain can be avoided, with the result that the computational time can be significantly decreased. The method of locally refining the mesh to obtain a more accurate solution on prescribed parts of the domain is usually referred to as h-refinement. Alternatively, a dual idea consists of increasing the degree of the polynomial functions that approximate the solution only on the elements that require greater accuracy. This approach is usually referred to as p-refinement. With the intention of combining the advantages of h-refinement and p-refinement, mixed methods have also been introduced [4, 21, 7], known as hp-refinement strategies.

In this work, the analysis is set in the framework of h-refinement for Lagrangian finite elements and local midpoint refinement is adopted as the refinement strategy.

---

\*Submitted to the journal's Methods and Algorithms for Scientific Computing section March 14, 2018; accepted for publication (in revised form) December 3, 2018; published electronically February 7, 2019.

<http://www.siam.org/journals/sisc/41-1/M117540.html>

**Funding:** This work was supported by the National Science Foundation grant DMS-1412796.

<sup>†</sup>Department of Mathematics and Statistics, Broadway & Boston, Texas Tech University, Lubbock TX 79409 ([eugenio.aulisa@ttu.edu](mailto:eugenio.aulisa@ttu.edu)).

<sup>‡</sup>Department of Scientific Computing, 400 Dirac Science Library, Florida State University, Tallahassee, FL 32306-4120 ([gcapodaglio@fsu.edu](mailto:gcapodaglio@fsu.edu)).

<sup>§</sup>Department of Mathematics and Physical Sciences, Louisiana State University of Alexandria, 8100 Hwy 71 S, Alexandria, LA 71302-9119 ([gke@lsu.edu](mailto:gke@lsu.edu)).

With this choice, special nodes called hanging nodes are introduced [15] and the finite-element solution ceases to be continuous, unless extra care is exerted to prevent this. For instance, continuity could be enforced using a multilevel approach [42, 2, 3]. More frequently, modified basis functions for the finite-element spaces are considered. As pointed out in [20], basis functions may be altered in two different ways, depending on whether or not degrees of freedom are associated with the hanging nodes. Our analysis fits in the framework of *constrained approximation*, where no degrees of freedom are associated with the hanging nodes. We refer to [22, 28] for works that, on the contrary, assign degrees of freedom to the hanging nodes.

In constrained approximation, most of the studies available in the literature require the finite-element grids to satisfy the so called 1-irregularity condition [8, 33, 31, 32, 37, 35, 17]. Such a condition requires the maximum difference between the refinement level of two adjacent elements to be one [36]. In this work, we present a novel approach for the construction of continuous basis functions defined on grids with arbitrary-level hanging node configurations. Indeed, the 1-irregularity condition is not enforced.

Only a few recent works have avoided the 1-regularity requirement [36, 18, 35, 27]. The strategy presented in this paper is simpler and more general than the ones suggested in the existing works. The method here proposed does not require the solution of a linear system for the determination of the constraint coefficients like [36, 7], and it does not require the shape functions to be of tensor product type like [18]. Rather, it can be easily implemented once the tree structure of the grid has been determined. Moreover, our approach works for applications in two and three dimensions for Lagrangian finite-element types of any order and shape, such as hexahedra, wedges, tetrahedra, and mixed triangulations, unlike other methods that only apply to two-dimensional problems or to specific element shapes in three dimensions. In the deal.II library [6], the case of 1-irregular meshes has been implemented in the framework of hp-refinement. There, the constraints are enforced in an edge-by-edge fashion and can be determined by solving a linear system that enforces continuity [7]. However, when dealing with arbitrary-level hanging nodes an edge-by-edge approach is not sufficient and the constraints extend to nodes belonging to neighboring elements. This is intrinsic in the nature of arbitrary-level hanging node configurations and cannot be avoided. To complete our results, we present a theoretical analysis that shows the new functions are continuous and linearly independent, and therefore form a basis for their spanning set.

In our approach, all the burden of dealing with the hanging nodes is embedded in building a projection operator between the original discontinuous space and the continuous one. This approach allows the implementation of the proposed method on existing finite-element codes with very few modifications, and it has been implemented in FEMuS [1], an open-source finite-element C++ library built on top of PETSc [5]. All our theoretical investigation is carried out in a multilevel setting. The construction of nested continuous finite-element spaces arising from the proposed continuous basis functions is also discussed. These finite-element spaces and corresponding projection operators are well suited for multigrid methods. The continuity of our finite-element spaces allows the multigrid level smoothing to be performed on all degrees of freedom. This global smoothing guarantees that the convergence of the method does not saturate on each level with increasing number of pre- and/or post-smoothing iterations. Such a feature cannot be achieved when the smoothing is performed only on a subspace of the multigrid spaces, as is done in many multigrid strategies for local refinement [11, 12, 24].

The paper is structured as follows. In section 2, the method for the construction of continuous basis functions is laid out. We introduce the interspace operators for our multigrid algorithm, and describe the implementation of a solver that makes use of the continuous finite-element spaces defined here. In section 3, we compute the spectral radius of the error operator associated with the multigrid algorithm applied to the bilinear form arising from the Poisson equation. A comparison is made with a multigrid algorithm that carries out smoothing only on subspaces [12]. This comparison shows better convergence properties for our algorithm for either a fixed or increasing number of smoothing iterations. In sections 4 and 5, linear and nonlinear numerical experiments are shown, where our multigrid algorithm is used as a preconditioner for other linear solvers. In particular, the two-dimensional Poisson problem and the three-dimensional buoyancy driven flow are used as numerical tests to showcase the suitability and broad range of applicability of our multigrid to multiphysics problems.

**2. Formulation.** In this section, the modified basis functions that define the finite-element spaces involved in the formulation of the problem are described. We introduce the interspace operators used in the multigrid context, and summarize the steps of a solver based on the proposed theory.

**2.1. Preliminaries.** Let  $J$  be a nonnegative integer and let  $k = 0, \dots, J$ . Here,  $J$  represents the maximum degree of local refinement, while  $k$  refers to a generic refined level. Let  $\Omega$  be a closed and bounded subset of  $\mathbb{R}^n$  for  $n = 1, 2, 3$  and let  $\mathcal{T}_0$  be a regular triangulation of size  $h_0$  on  $\Omega$ . The term triangulation on  $\Omega$  is intended as a finite collection of pairwise disjoint closed sets  $\{T_i\}$  such that  $\cup T_i = \Omega$ . A regular triangulation has no hanging nodes. With a slight abuse of terminology, we use the word triangulation even for grids that are not composed of triangular elements. As a matter of fact, many element types other than triangles are considered in the numerical simulations. Let  $\{\Theta_k\}_{k=0}^J$  be a collection of closed overlapping subsets of  $\Omega$ , such that

$$\Theta_J \subseteq \Theta_{J-1} \subseteq \dots \subseteq \Theta_0 \equiv \Omega.$$

For any given  $k = 1, \dots, J$ , we can define recursively an irregular triangulation  $\mathcal{T}_k$  on  $\Omega$  starting from the triangulation  $\mathcal{T}_{k-1}$  in the following way.

**DEFINITION 2.1.** *Let  $\mathcal{T}_0$  be a quasi-uniform triangulation built on  $\Theta_0$ . Then, for  $k = 1, \dots, J$ ,  $\mathcal{T}_k$  is obtained by midpoint refinement of only the elements of  $\mathcal{T}_{k-1}$  that lie on  $\Theta_k$ .*

We next define new sequences of subsets of  $\Omega$ . An example is reported in Figure 1. These sequences are introduced to set our formulation in a multilevel framework, in order to easily apply our analysis to multigrid methods.

**DEFINITION 2.2.** *For all  $l = 0, \dots, k$  with  $k \leq J$ , we define the sets*

$$\Omega_k^l \equiv \begin{cases} \overline{\Theta_l \setminus \Theta_{l+1}} & \text{if } l < k, \\ \Theta_k & \text{if } l = k, \end{cases}$$

and  $\Gamma_k^{l,n} \equiv \Omega_k^l \cap \Omega_k^n$  to be the interface between any two of such sets.

To summarize,  $J$  denotes the maximum degree of refinement,  $k$  refers to the current refinement level considered, while  $l$  identifies the different subdomains of  $\Omega$  at the current level  $k$ .

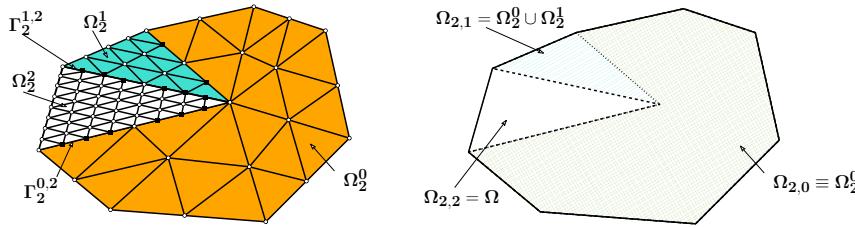


FIG. 1. An irregular grid with hanging nodes in two dimensions with  $J = 2$ . The subdomains involved in the formulation are also highlighted.

DEFINITION 2.3. For all  $l = 0, \dots, k$  with  $k \leq J$  define

$$\Omega_{k,0} \equiv \Omega_k^0, \quad \Omega_{k,l} \equiv \bigcup_{m=0}^l \Omega_k^m, \quad \Omega_k \equiv \Omega_{k,k}.$$

*Remark 1.* Note that  $\Omega_k = \Omega$  and, by definition,  $\{\Omega_{k,l}\}_{l=0}^k$  is a sequence of nested sets. This sequence will be fundamental in the analysis that follows to define the discontinuous basis functions used for the construction of the continuous ones.

By construction of  $\mathcal{T}_k$ , we have that, for a given  $k$  level,  $k+1$  regular triangulations are present on  $k+1$  different subsets of  $\Omega$ . We formalize this in the following definition.

DEFINITION 2.4. For all  $l = 0, \dots, k$  with  $k \leq J$  we define  $\mathcal{T}_k^l$  to be the regular triangulation composed of all the elements  $T \in \mathcal{T}_k$  such that  $T \cap \text{int}(\Omega_k^l) \neq \emptyset$ .

From this definition we have that  $\mathcal{T}_k^l$  is a triangulation defined only on  $\Omega_k^l$  and not on the entire domain  $\Omega$  as  $\mathcal{T}_k$ . Triangulations can be defined on  $\Omega_{k,l} \subseteq \Omega$  in the following way.

DEFINITION 2.5. For all  $l = 0, \dots, k$  with  $k \leq J$ , define

$$\mathcal{T}_{k,0} \equiv \mathcal{T}_k^0, \quad \mathcal{T}_{k,l} \equiv \bigcup_{m=0}^l \mathcal{T}_k^m.$$

Then  $\mathcal{T}_{k,l}$  is an irregular triangulation on  $\Omega_{k,l}$  for all  $l = 1, \dots, k$ .

Note that  $\mathcal{T}_{k,k} = \mathcal{T}_k$ . Moreover, the above definition is consistent with Definition 2.3.

Due to the local midpoint refinement procedure, hanging nodes are present on  $\mathcal{T}_k$  for  $k = 1, \dots, J$ . These are special nodes of some element  $T_i \in \mathcal{T}_k$  that lie on edges (or faces) of another element  $T_j \in \mathcal{T}_k$  without being nodes for  $T_j$ . For example, as can be seen from Figure 1, where linear elements are used, the hanging nodes are marked with black squares.

Using the triangulations we defined above, for all  $k = 0, \dots, J$ , we can define finite-element spaces in the following way:

$$(2.1) \quad \begin{aligned} V_k^l &= \{v \in H^1(\Omega_k^l) : v|_T \in \mathcal{P}_\alpha \ \forall T \in \mathcal{T}_k^l\}, \quad l = 0, \dots, k, \\ V_k &= \{v : v|_{\text{int}(T)} \in \mathcal{P}_\alpha, \ \forall T \in \mathcal{T}_k\}, \\ \widehat{V}_k &= \{\widehat{v} \in H^1(\Omega) \cap C^0(\Omega) : \widehat{v}|_T \in \mathcal{P}_\alpha, \ \forall T \in \mathcal{T}_k\}, \end{aligned}$$

where  $\mathcal{P}_\alpha$  represents the set of polynomials of degree  $\alpha$ . Note that the spaces  $V_k^l$  are continuous by construction since the functions are defined on regular grids. The space  $V_k$  is built on an irregular triangulation and contains continuous and discontinuous functions. For the space  $\widehat{V}_k$ , we enforce continuity by removing from  $V_k$  the hang-

ing nodal functions, and enriching specific basis functions with contributions from the removed ones. An explicit characterization of such finite-element spaces will be discussed in the next sections.

**2.2. Construction of the uniform finite-element spaces  $V_k^l$ .** Let  $X_k^l$  be the set of nodal points of the triangulation  $\mathcal{T}_k^l$ ,  $X_{k,l} = \cup_{m=0}^l X_k^m$  be the set of nodal points of  $\mathcal{T}_{k,l}$  and  $X_k = X_{k,k}$  be the set of nodal points of  $\mathcal{T}_k$ . Denote by  $X \equiv X_J$  the set of nodal points of the triangulation  $\mathcal{T}_J$ . Also, let  $\text{card}(X_k) = N_k$  and  $\text{card}(X) = N$ . Depending on  $\alpha$ , a nodal point could be a vertex, an edge midpoint, a face center, an element center, and more.

*Remark 2.* There exist points  $x_i \in X_k$  for which  $x_i \in \bigcap_{l \in A_{i,k}} X_k^l$  for some set  $A_{i,k} \subseteq \{0, \dots, k\}$ .

This means that there are nodal points that belong to multiple  $X_k^l$ , since by Definition 2.2, the  $\Omega_k^l$  are allowed to overlap on portions of their boundaries.

**DEFINITION 2.6.** We define  $\varphi_{k,i}^l$  to be the standard Lagrangian nodal basis of  $V_k^l$  associated with  $x_i^l \in X_k^l$ .

Three considerations about the functions  $\varphi_{k,i}^l$  are listed:

- The functions  $\varphi_{k,i}^l$  satisfy the delta property

$$(2.2) \quad \varphi_{k,i}^l(x_j^l) = \delta_{ij} \quad \text{for all } x_j^l \in X_k^l$$

and they are not defined outside  $\Omega_k^l$ .

- There exist basis functions  $\varphi_{k,i}^m$  and  $\varphi_{k,i}^n$ , defined on different subdomains  $\Omega_k^m$  and  $\Omega_k^n$  associated with the same nodal point,  $x_i^m = x_i^n = x_i \in \Gamma_k^{m,n}$  (the *master* node). The two functions take different values on the shared interface  $\Gamma_k^{m,n}$  and if  $m < n$ , then  $\text{supp}(\varphi_{k,i}^n|_{\Gamma_k^{m,n}}) \subset \text{supp}(\varphi_{k,i}^m|_{\Gamma_k^{m,n}})$ .
- Letting  $m < n$ , there exist basis functions  $\varphi_{k,i}^n$  defined on the finer triangulation associated with nodal points  $x_i \in \Gamma_k^{m,n}$  (the *hanging nodes*) for which no corresponding functions  $\varphi_{k,i}^m$  defined on the coarse triangulation exist.

*Remark 3.* If for any given  $m$  and  $n$  the point  $x_i \in X_k$  belongs to  $\Gamma_k^{m,n}$ , then it is either a master or a hanging node. Nodes in  $X_k$  that are neither master nor hanging will be referred to as interior nodes.

**2.3. Construction of the discontinuous finite-element space  $V_k$ .** The next goal is to build basis functions  $\varphi_{k,i}$  on the whole domain  $\Omega$ , that in the interior of each subdomain  $\Omega_k^l$  take the corresponding value of  $\varphi_{k,i}^l$ , and on a shared interface  $\Gamma_k^{m,n}$  take the value from the coarser triangulation. Assuming  $\varphi_{k,i}^m$  is the function associated with the coarser triangulation, such a value is given by  $\varphi_{k,i}^m$  if  $x_i$  is a master node, and it is zero if  $x_i$  is a hanging node.

**DEFINITION 2.7.** For  $k \leq J$  and  $l = 1, \dots, k$  we define the sets

$$(2.3) \quad \mathcal{E}_k^l \equiv \Omega_{k,l} \setminus \Omega_{k,l-1}$$

and  $\mathcal{E}_k^0 \equiv \Omega_{k,0}$ .

Note that  $\mathcal{E}_k^l \neq \Omega_k^l$  in general, since the sets  $\{\Omega_k^l\}_{l=0}^k$  in Definition 2.2 are not disjoint. Any point  $x \in \mathcal{E}_k^l$  is either in the interior of  $\Omega_k^l$  or on the shared interface  $\Gamma_k^{l,n}$  with  $n > l$ , but it is not on the shared interface  $\Gamma_k^{l,m}$  with  $m < l$ . It follows from

Definition 2.7 that  $\{\mathcal{E}_k^l\}_{l=0}^k$  is a pairwise disjoint collection of subsets of  $\Omega$  and

$$(2.4) \quad \Omega = \bigcup_{l=0}^k \mathcal{E}_k^l.$$

Therefore,  $\{\mathcal{E}_k^l\}_{l=0}^k$  represents a disjoint cover of  $\Omega$  and so, if  $x \in \Omega$ , then there exists a unique  $\gamma \in \{0, 1, \dots, k\}$  such that  $x \in \mathcal{E}_k^\gamma$ .

**DEFINITION 2.8.** For all  $k = 0, \dots, J$  and  $x_i \in X_k$ , the functions  $\varphi_{k,i}$  are defined for all  $x$  in  $\Omega$  in the following way:

$$\varphi_{k,i}(x) = \begin{cases} \varphi_{k,i}^\gamma(x) & \text{if } x_i \in X_k^\gamma, \\ 0 & \text{otherwise,} \end{cases}$$

where  $\gamma$  is the unique integer for which  $x \in \mathcal{E}_k^\gamma$ .

Let us highlight the main features of the functions  $\varphi_{k,i}$ :

- The functions  $\varphi_{k,i}$  can be either continuous or discontinuous and are defined on all  $\Omega$ .
- $\varphi_{k,i}$  is continuous if and only if its nodal point is located in the interior of some  $\Omega_k^l$  (where there is a regular triangulation  $\mathcal{T}_k^l$ ) or on the exterior boundary if surrounded by a regular triangulation. When  $\varphi_{k,i}$  is continuous, it can be considered the zero extension of  $\varphi_{k,i}^l$  to the whole domain, and it satisfies the delta property

$$(2.5) \quad \varphi_{k,i}(x_j) = \delta_{ij} \text{ for all } x_j \in X_k.$$

- $\varphi_{k,i}$  is discontinuous if and only if its nodal point is either a master or a hanging node.
- A function  $\varphi_{k,i}$  associated with a master node does not satisfy the delta property (2.5) for some  $x_j$ , with  $x_j$  being a hanging node.
- A function  $\varphi_{k,i}$  associated with a hanging node does not satisfy the delta property (2.5), but instead it satisfies the zero property

$$(2.6) \quad \varphi_{k,i}(x_j) = 0 \text{ for all } x_j \in X_k.$$

- Let  $x_j \in X_k$ , hence, there exists at least one  $X_k^l$  for which  $x_j \in X_k^l$ . For all  $x_i \in X_k$ , we have

$$(2.7) \quad \lim_{x \in \mathcal{E}_k^l \rightarrow x_j} \varphi_{k,i}(x) = \delta_{ij}.$$

In fact, by Definition 2.8, on  $\mathcal{E}_k^l$  the function  $\varphi_{k,i}$  is either zero or equal to the function  $\varphi_{k,i}^l$  which is continuous and satisfies the delta property (2.2).

We now have to introduce a more complex idea than the concept of master and hanging nodes.

**DEFINITION 2.9.** A node  $x_i \in X_k$  is a parent of a node  $x_j \neq x_i$ ,  $x_j \in X_k$ , if

$$\varphi_{k,i}(x_j) \neq 0.$$

The node  $x_j$  is called a child of  $x_i$ .

The reader is advised to become familiarized with Figures 2, 3, 4, and 5 or Figures 6, 7, 8, and 9. It might be beneficial to keep them handy as new notations and definitions are introduced next. Parent nodes are master nodes while child nodes are hanging nodes. If a parent node  $x_i$  has a child  $x_j$  which itself has a child  $x_n$ , then  $x_i$  is a *grandparent* of  $x_n$  and  $x_n$  is a *grandchild* of  $x_i$ . Note that being both a child and a parent,  $x_j$  is both a master and a hanging node. Hence  $\varphi_{k,j}(x_j) = 0$ , but

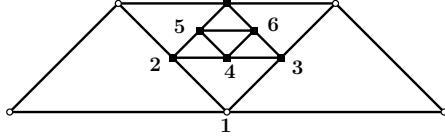


FIG. 2. Example of a two-dimensional bilinear grid with a two level local refinement.

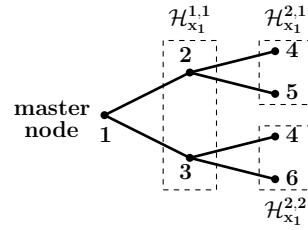


FIG. 3. Tree diagram that displays the family tree of node  $x_1$ .

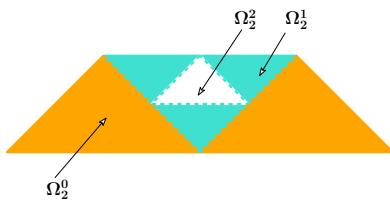


FIG. 4. Subdomains associated with the different degrees of refinement of the grid in Figure 2.

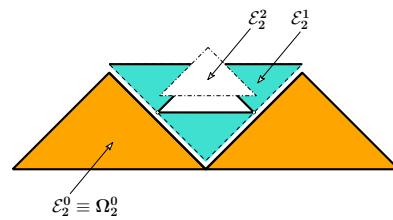


FIG. 5. Collection  $\{E_2^l\}_{l=0}^2$  associated with the grid in Figure 2.

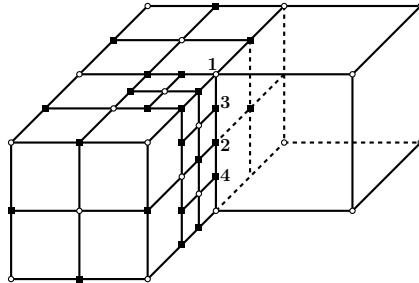


FIG. 6. Example of three-dimensional bilinear grid on an L-shaped domain with a three level local refinement.

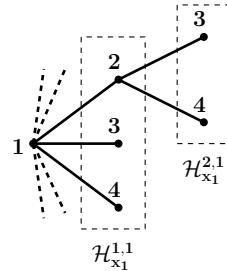


FIG. 7. Tree diagram that displays the family tree of node  $x_1$ .

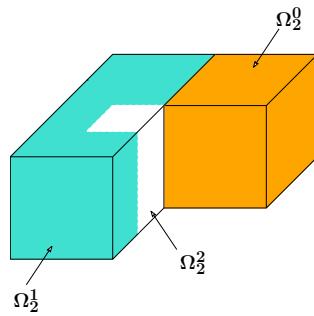


FIG. 8. Subdomains associated with the different degrees of refinement of the grid in Figure 6.

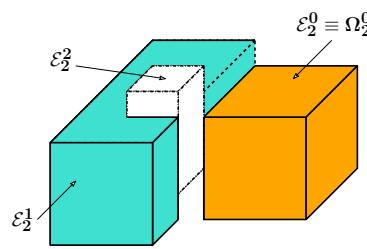


FIG. 9. Collection  $\{E_2^l\}_{l=0}^2$  associated with the grid in Figure 6.

$\varphi_{k,j}(x_n) \neq 0$ . Nodes that are both master and hanging ultimately are considered hanging nodes, as the following definition states.

DEFINITION 2.10. Let  $k \in \{0, 1, \dots, J\}$  be fixed and recall that  $\Omega = \bigcup_{l=0}^k \Omega_k^l$ . We define the following sets:

- Interior :  $\mathcal{I}_k = \{x_i \in X_k \mid x_i \in \text{int}(\Omega_k^l) \text{ for some } l\};$
- Hanging :  $\mathcal{H}_k = \{x_i \in X_k \mid \varphi_{k,j}(x_i) \neq 0 \text{ for at least one } x_j \neq x_i\};$
- Master :  $\mathcal{M}_k = \{x_i \in X_k \mid \varphi_{k,i}(x_j) \neq 0 \text{ for at least one } x_j \in \mathcal{H}_k, \text{ and } x_i \notin \mathcal{H}_k\}.$

The set  $\mathcal{I}_k$  is the set of interior nodes at the level  $k$ . We refer to  $\mathcal{M}_k$  as the set of master nodes at the level  $k$ , and it is the set of all nodes that are parents without ever being children. We call  $\mathcal{H}_k$  the set of hanging nodes at level  $k$ , and it is the set of all nodes that are children at least once. It includes all the parents that are also children. Note that the sets in Definition 2.10 are disjoint and that  $\mathcal{I}_k \cup \mathcal{M}_k \cup \mathcal{H}_k = X_k$ ; recall Remark 3.

LEMMA 2.11. The set  $\{\varphi_{k,i} : x_i \in X_k\}$  is linearly independent for all  $k = 0, \dots, J$ .

*Proof.* Let  $\varphi_{k,j}$  be associated with  $x_j \in X_k$ . Proceeding with proof by contradiction, assume that

$$(2.8) \quad \varphi_{k,j}(x) = \sum_{i:x_i \in X_k, x_i \neq x_j} c_i \varphi_{k,i}(x),$$

where  $c_i$  are some real coefficients, not all equal to zero. Recalling (2.7), there exists at least one  $X_k^l$  for which  $x_j \in X_k^l$ , and for all  $x_i \in X_k$ ,

$$\lim_{x \in \mathcal{E}_k^l \rightarrow x_j} \varphi_{k,i}(x) = \delta_{ij}.$$

Hence, the following contradiction is obtained:

$$1 = \lim_{x \in \mathcal{E}_k^l \rightarrow x_j} \varphi_{k,j}(x) = \lim_{x \in \mathcal{E}_k^l \rightarrow x_j} \sum_{i:x_i \in X_k, x_i \neq x_j} c_i \varphi_{k,i}(x) = 0. \quad \square$$

With the above lemma we can formally define the space  $V_k$  first introduced in (2.1).

DEFINITION 2.12. We define  $V_k$  to be the set spanned by  $\{\varphi_{k,i}\}_{i=1}^{N_k}$ . Because of Lemma 2.11, it follows that  $\{\varphi_{k,i}\}_{i=1}^{N_k}$  also forms a basis for  $V_k$ , where  $V_k$  is a vector space with the standard addition and scalar multiplication for real valued functions.

**2.4. Construction of the continuous finite-element space  $\widehat{V}_k$ .** The next step is to build basis functions  $\widehat{\varphi}_{k,i}$  on the whole domain  $\Omega$  that are also continuous. Let  $x_i \in \mathcal{M}_k$  and define the set  $\mathcal{H}_{x_i}^1$  as the set of all hanging nodes associated with node  $x_i$ . Nodes in  $\mathcal{H}_{x_i}^1$  are children of  $x_i$ . With the same idea, we can define the following sets:

$$\begin{aligned} \mathcal{H}_{x_i}^2 &= \{x \mid x \in \mathcal{H}_y^1 \text{ for some } y \in \mathcal{H}_{x_i}^1\}, \\ \mathcal{H}_{x_i}^3 &= \{x \mid x \in \mathcal{H}_y^1 \text{ for some } y \in \mathcal{H}_{x_i}^2\}, \\ &\vdots \\ \mathcal{H}_{x_i}^\alpha &= \{x \mid x \in \mathcal{H}_y^1 \text{ for some } y \in \mathcal{H}_{x_i}^{\alpha-1}\}. \end{aligned}$$

The set  $\mathcal{H}_{x_i}^2$  is the set of hanging nodes associated with nodes in  $\mathcal{H}_{x_i}^1$ , therefore, any node in  $\mathcal{H}_{x_i}^2$  is a grandchild of  $x_i$ . The set  $\mathcal{H}_{x_i}^3$  is the set of hanging nodes associated with nodes in  $\mathcal{H}_{x_i}^2$  so it contains the *great-grandchildren* of  $x_i$  and so on. If  $x \in \mathcal{H}_{x_i}^\alpha$  we say that  $x$  is a hanging node associated with  $x_i$  of *degree* at least  $\alpha$ . Note that a given node can be a hanging node for more than just one master (a given node can be the child of more than one parent). Therefore, for given  $\alpha$ , we introduce the sequence  $\{x_{i,\beta}^\alpha\}_{\beta=1}^{\beta_{\alpha}^{\max}}$ , where  $\beta_{\alpha}^{\max}$  denotes the number of hanging nodes of degree  $\alpha$  associated with the master node  $x_i$ , but considering *repetition*. Observe that  $\{x_{i,\beta}^\alpha\}_{\beta=1}^{\beta_{\alpha}^{\max}} \subseteq \mathcal{H}_{x_i}^\alpha$ . Moreover, we also point out that for  $\alpha = 1$  the elements of the sequence  $\{x_{i,\beta}^\alpha\}_{\beta=1}^{\beta_{\alpha}^{\max}}$  do not repeat and if  $\alpha = 0$  there is only  $x_{i,1}^0 \equiv x_i$ . We define the sets

$$\begin{aligned}\mathcal{H}_{x_i}^{1,1} &\equiv \mathcal{H}_{x_i}^1, \\ \mathcal{H}_{x_i}^{\alpha,\beta} &= \{x \in \mathcal{H}_{x_i}^\alpha \mid x \in \mathcal{H}_{x_{i,\beta}^{\alpha-1}}^1\} \text{ for } \alpha \geq 2 \text{ and } \beta = 1, \dots, \beta_{\alpha-1}^{\max}.\end{aligned}$$

**DEFINITION 2.13.** *If  $x_j \in \mathcal{H}_{x_i}^{\alpha,\beta}$  then there exist  $\alpha + 1$  nodes  $x_{j_0}, x_{j_1}, x_{j_2}, \dots, x_{j_\alpha}$  (with  $x_{j_0} \equiv x_i$  and  $x_{j_\alpha} \equiv x_j$ ) that we refer to as the *ancestors* of  $x_j$ , such that for all  $m = 0, \dots, \alpha$ , we have  $x_{j_m} = x_{i,\beta_{j_m}}^m$  for some  $\beta_{j_m} \in \{1, 2, \dots, \beta_m^{\max}\}$ . Note that  $\beta_{j_0} = 1$  always.*

The *ancestors* of  $x_j$  depend on  $\alpha$  and  $\beta$ . However, not to make the notation too heavy, such letters do not appear in the *ancestor* sequence. Moreover, we remark that the only  $m \in \{0, \dots, \alpha\}$  for which  $x_{j_m} \equiv x_j$  is  $m = \alpha$ .

For  $\alpha \geq 2$ , and any  $x_j \in \mathcal{H}_{x_i}^{\alpha,\beta}$ , define the set

$$U_{x_i,x_j}^{\alpha,\beta} = \bigcup_{m=1}^{\alpha-1} \mathcal{H}_{x_i}^{\alpha-m,\beta_{j_{\alpha-m-1}}}.$$

The above set represents the set of all *uncles* (parents included) and *great-uncles* (grandparents included) of  $x_j \in \mathcal{H}_{x_i}^{\alpha,\beta}$ . Moreover, for  $\alpha \geq 2$ , we also define

$$S_{x_i,x_j}^{\alpha,\beta} = \{x_{j_\gamma} \in \{x_{j_m}\}_{m=2}^\alpha \mid x_{j_\gamma} \in U_{x_i,x_{j_\gamma}}^{\gamma,\beta_{j_{\gamma-1}}}\},$$

where  $x_{j_\gamma} \in \mathcal{H}_{x_i}^{\gamma,\beta_{j_{\gamma-1}}}$ . This is the set of all nodes in the ancestor sequence  $\{x_{j_m}\}_{m=1}^\alpha$  of  $x_j$  that are uncles or great-uncles of *themselves*. An example is visible in Figures 6 and 7. Finally, we introduce the set of all nodes in  $\mathcal{H}_{x_i}^{\alpha,\beta}$  for which no ancestor is uncle or great-uncle of *itself*, namely,

$$\begin{aligned}\tilde{\mathcal{H}}_{x_i}^{1,1} &= \mathcal{H}_{x_i}^{1,1}, \\ \tilde{\mathcal{H}}_{x_i}^{\alpha,\beta} &= \{x_j \in \mathcal{H}_{x_i}^{\alpha,\beta} \mid S_{x_i,x_j}^{\alpha,\beta} = \emptyset\} \text{ for } \alpha \geq 2 \text{ and } \beta = 1, \dots, \beta_{\alpha-1}^{\max}.\end{aligned}$$

Let  $\beta_0^{\max} \equiv 1$  and  $\alpha_i^{\max} = \max\{\alpha \mid \tilde{\mathcal{H}}_{x_i}^{\alpha,\beta} \neq \emptyset, \text{ with } \beta = 1, \dots, \beta_{\alpha-1}^{\max}\}$ . An example of  $\alpha_i^{\max} > 1$  is reported in Figures 2 and 3.

**DEFINITION 2.14.** For all  $x_i \in X_k$ , we define functions  $\hat{\varphi}_{k,i}$  using the previously introduced functions  $\{\varphi_{k,i}\}_{i=1}^{N_k}$  in the following way:

$$\hat{\varphi}_{k,i} = \begin{cases} \varphi_{k,i} & \text{if } i \in \mathcal{I}_k, \\ \varphi_{k,i} + \sum_{\alpha=1}^{\alpha_i^{\max}} \sum_{\beta=1}^{\beta_{\alpha-1}^{\max}} \left( \sum_{j:x_j \in \tilde{\mathcal{H}}_{x_i}^{\alpha,\beta}} \left( \prod_{m=1}^{\alpha} \varphi_{k,jm-1}(x_{jm}) \right) \varphi_{k,j} \right) & \text{if } i \in \mathcal{M}_k, \\ 0 & \text{if } i \in \mathcal{H}_k. \end{cases}$$

We reorder the basis functions just defined so that the interior nodes come first, the master nodes second and the hanging nodes last. In matrix-vector form, the vector  $\hat{\varphi}_k$  of the reordered  $\hat{\varphi}_{k,i}$  functions can be written as

$$(2.9) \quad \hat{\varphi}_k = \begin{bmatrix} \hat{\varphi}_{k,\mathcal{I}_k} \\ \hat{\varphi}_{k,\mathcal{M}_k} \\ \hat{\varphi}_{k,\mathcal{H}_k} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & \Phi_k \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{k,\mathcal{I}_k} \\ \varphi_{k,\mathcal{M}_k} \\ \varphi_{k,\mathcal{H}_k} \end{bmatrix} = \hat{R}_k \varphi_k.$$

**Remark 4.** The hanging nodes are considered in the definition of the functions  $\hat{\varphi}_{k,i}$  to maintain the same dimension for vectors and matrices used in either  $V_k$  or  $\hat{V}_k$ . In this way, an identity mapping between the degrees of freedom of  $V_k$  and  $\hat{V}_k$  is preserved. This is convenient for the implementation of the algorithm, especially in a parallel environment and, as it will be shown later, for extending the formulation to a multilevel setting.

Let us now consider two simple examples that do not fall in the case of 1-irregular meshes. These are illustrative to better understand the algorithm described in this section. Their generalization to complex arbitrary-level hanging node meshes as the ones used in the next sections is straightforward.

We start with a two-dimensional example to visualize the construction of a function associated with a node in  $\mathcal{M}_k$ , as described in Definition 2.14. The triangulation considered in the example is depicted in Figure 2. This can be part of a larger triangulation but for simplicity we just focus on the portion in the figure. We consider bilinear triangular elements with a maximum of two level local refinement (so  $J = 2$ ) and focus on the master node  $x_1 \in \mathcal{M}_2$ .

Let us start describing the sets and sequences involved in Definition 2.14 for this specific example:

$$\begin{aligned} \mathcal{H}_{x_1}^1 &= \{x_2, x_3\}, \quad \mathcal{H}_{x_1}^2 = \{x_4, x_5, x_6\}, \\ \{x_{1,\beta}^1\}_{\beta=1}^2 &= \{x_2, x_3\}, \quad \{x_{1,\beta}^2\}_{\beta=1}^4 = \{x_4, x_5, x_4, x_6\}, \\ \tilde{\mathcal{H}}_{x_1}^{1,1} &= \mathcal{H}_{x_1}^{1,1} = \{x_2, x_3\}, \quad \tilde{\mathcal{H}}_{x_1}^{2,1} = \mathcal{H}_{x_1}^{2,1} = \{x_4, x_5\}, \quad \tilde{\mathcal{H}}_{x_1}^{2,2} = \mathcal{H}_{x_1}^{2,2} = \{x_4, x_6\}. \end{aligned}$$

Note that, in this case, there are no hanging nodes that are uncles or great-uncles of themselves. In Figure 3, the family tree associated with node  $x_1$  is reported. The subdomains related to the different degrees of refinement and the collection  $\{\mathcal{E}_2^l\}_{l=0}^2$  are shown in Figures 4 and 5.

**Remark 5.** In the construction of the tree structure of the mesh, the child of a master node is always located on the same edge (or face) of the parent. Only when considering grandchildren, great-grandchildren, and so on, does the search extends to neighboring nodes and edges/faces. This situation is embedded in the nature of

arbitrary-level hanging nodes, and it cannot be avoided [36, 37]. For such a reason, an exclusive edge-by-edge approach is not possible in the case of arbitrary-level hanging nodes. The search for the children, however, remains localized on the support of the coarse master node basis, so the proposed approach is still local.

Using Definition 2.14 we obtain

$$\begin{aligned}\widehat{\varphi}_{k,1} &= \varphi_{k,1} + \varphi_{k,1}(x_2)\varphi_{k,2} + \varphi_{k,1}(x_3)\varphi_{k,3} + \varphi_{k,1}(x_2)\varphi_{k,2}(x_4)\varphi_{k,4} \\ &\quad + \varphi_{k,1}(x_2)\varphi_{k,2}(x_5)\varphi_{k,5} + \varphi_{k,1}(x_3)\varphi_{k,3}(x_4)\varphi_{k,4} + \varphi_{k,1}(x_3)\varphi_{k,3}(x_6)\varphi_{k,6} \\ &= \varphi_{k,1} + 0.5\varphi_{k,2} + 0.5\varphi_{k,3} + 0.5 \cdot 0.5\varphi_{k,4} + 0.5 \cdot 0.5\varphi_{k,5} \\ &\quad + 0.5 \cdot 0.5\varphi_{k,4} + 0.5 \cdot 0.5\varphi_{k,6} \\ &= \varphi_{k,1} + 0.5\varphi_{k,2} + 0.5\varphi_{k,3} + 0.25\varphi_{k,4} + 0.25\varphi_{k,5} + 0.25\varphi_{k,4} + 0.25\varphi_{k,6} \\ &= \varphi_{k,1} + 0.5\varphi_{k,2} + 0.5\varphi_{k,3} + 0.5\varphi_{k,4} + 0.25\varphi_{k,5} + 0.25\varphi_{k,6}, \\ \widehat{\varphi}_{k,2} &\equiv 0, \quad \widehat{\varphi}_{k,3} \equiv 0, \quad \widehat{\varphi}_{k,4} \equiv 0, \quad \widehat{\varphi}_{k,5} \equiv 0, \quad \widehat{\varphi}_{k,6} \equiv 0.\end{aligned}$$

The matrix-vector form corresponding to (2.9) is given by

$$\begin{bmatrix} \widehat{\varphi}_{k,1} \\ \widehat{\varphi}_{k,2} \\ \widehat{\varphi}_{k,3} \\ \widehat{\varphi}_{k,4} \\ \widehat{\varphi}_{k,5} \\ \widehat{\varphi}_{k,6} \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.5 & 0.25 & 0.25 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{k,1} \\ \varphi_{k,2} \\ \varphi_{k,3} \\ \varphi_{k,4} \\ \varphi_{k,5} \\ \varphi_{k,6} \end{bmatrix}.$$

We continue with a three-dimensional example of an L-shaped domain, visible in Figure 6. Once again the maximum degree of local refinement is two, so  $J = 2$ , and we only focus on the node  $x_1 \in \mathcal{M}_2$ . We present this specific example because the sets  $\mathcal{H}_{x_1}^{2,1}$  and  $\widetilde{\mathcal{H}}_{x_1}^{2,1}$  are different in this case and so it will be clear why we consider the sets  $\widetilde{\mathcal{H}}_{x_1}^{2,1}$  instead of  $\mathcal{H}_{x_1}^{2,1}$  in Definition 2.14. In Figures 8 and 9 the subdomains associated with the different degrees of refinement are shown together with the collection  $\{\mathcal{E}_2^l\}_{l=0}^2$ . Since the goal of this example is to illustrate a situation where the sets  $\mathcal{H}_{x_1}^{2,1}$  and  $\widetilde{\mathcal{H}}_{x_1}^{2,1}$  are different, we only consider the contributions given by the numbered nodes in Figure 6.

Let us describe the sets and sequences involved in Definition 2.14 for this three-dimensional example:

$$\begin{aligned}\mathcal{H}_{x_1}^1 &= \{x_2, x_3, x_4\}, \quad \mathcal{H}_{x_1}^2 = \{x_3, x_4\}, \\ \{x_{1,\beta}^1\}_{\beta=1}^3 &= \{x_2, x_3, x_4\}, \quad \{x_{1,\beta}^2\}_{\beta=1}^2 = \{x_3, x_4\}, \\ \widetilde{\mathcal{H}}_{x_1}^{1,1} &= \mathcal{H}_{x_1}^{1,1} = \{x_2, x_3, x_4\}, \quad \widetilde{\mathcal{H}}_{x_1}^{2,1} = \emptyset, \quad \mathcal{H}_{x_1}^{2,1} = \{x_3, x_4\}.\end{aligned}$$

The reason why  $\widetilde{\mathcal{H}}_{x_1}^{2,1} = \emptyset$  is because the nodes  $x_3$  and  $x_4$  are uncles of themselves, so they are removed from  $\mathcal{H}_{x_1}^{2,1}$ . However, since this set is composed only of such nodes, it becomes the empty set. The family tree of node  $x_1$  is visible in Figure 7. Using Definition 2.14 we obtain

$$\begin{aligned}\widehat{\varphi}_{k,1} &= \varphi_{k,1} + \varphi_{k,1}(x_2)\varphi_{k,2} + \varphi_{k,1}(x_3)\varphi_{k,3} + \varphi_{k,1}(x_4)\varphi_{k,4} + f_{k,1} \\ &= \varphi_{k,1} + 0.5\varphi_{k,2} + 0.75\varphi_{k,3} + 0.25\varphi_{k,4} + f_{k,1}, \\ \widehat{\varphi}_{k,2} &\equiv 0, \quad \widehat{\varphi}_{k,3} \equiv 0, \quad \widehat{\varphi}_{k,4} \equiv 0,\end{aligned}$$

where  $f_{k,i}$  is a function that gives the contributions of the other hanging nodes associated with  $x_1$  that have not been made explicit in this example.

*Remark 6.* In these examples we only addressed bilinear and trilinear elements for simplicity. The same method applies to all Lagrangian elements of any polynomial degree satisfying the delta properties. In the numerical example section we used bi/trilinear, quadratic, bi/triquadratic elements, and combinations of them.

**THEOREM 2.15.** *The functions  $\widehat{\varphi}_{k,i}$  are continuous for all  $x_i \in X_k$ .*

*Proof.* For any node  $x_i \in \mathcal{I}_k$ , the continuity is inherited from  $\varphi_{k,i}$ , while if  $x_i \in \mathcal{H}_k$ , then  $\widehat{\varphi}_{k,i}$  is identically 0, thus it is continuous. For  $x_i \in \mathcal{M}_k$ , the proof is more complex. Vaguely speaking we want to show that if  $\varphi_{k,i}$  is a discontinuous function associated with a master node  $x_i$ , it is possible to build a continuous function  $\widehat{\varphi}_{k,i}$  with the contributions from hanging node functions,  $\varphi_{k,j}$ , for some  $j$ . Therefore, letting  $x_i \in \mathcal{M}_k$ , we will show that  $\widehat{\varphi}_{k,i}$  is continuous at all hanging nodes  $x_j$  associated with  $x_i$ . Note that as we have already discussed,  $x_j$  could belong to more than just one  $\tilde{\mathcal{H}}_{x_i}^{\alpha,\beta}$ , meaning that  $x_j$  could be a child of more than just one parent. Let  $e_j$  be the edge (or face) containing  $x_j$  that belongs to the coarsest triangulation that has  $x_j$  among its nodal points. On  $e_j$  then, there will be  $\eta_{\max}$  parent nodes  $x_{j_\eta} \in \mathcal{H}_{x_i}^\alpha$  for some  $\alpha$  for which  $\varphi_{j_\eta}(x_j) \neq 0$ . Considering that  $x_j$  can posses multiple degrees as hanging nodes of  $x_i$ , let  $\alpha_\eta$  be the degree of  $x_j$  as a child of  $x_{j_\eta}$ . Then there will be  $\eta_{\max}$  ancestor sequences  $\{x_{j_\eta,\zeta}\}_{\zeta=0}^{\alpha_\eta}$  with  $x_{j_\eta,\alpha_\eta} \equiv x_j$ ,  $x_{j_\eta,\alpha_\eta-1} \equiv x_{j_\eta}$ , and  $x_{j_\eta,0} \equiv x_i$ . If we denote by  $\varphi_{k,j_\eta,\zeta}$  the functions whose nodal point is  $x_{j_\eta,\zeta}$ , then  $\widehat{\varphi}_{k,i}$  can be rewritten as

(2.10)

$$\widehat{\varphi}_{k,i} = \sum_{\eta=1}^{\eta_{\max}} \left[ \left( \prod_{\zeta=1}^{\alpha_\eta-1} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta,\zeta}) \right) \varphi_{k,j_\eta} + \left( \prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta,\zeta}) \right) \varphi_{k,j_\eta} \right] + f_j,$$

where  $f_j$  is a function for which  $f_j(x_j) = 0$  and  $\lim_{x \rightarrow x_j} f_j(x) = 0$ . Since  $x_j$  is a hanging node, we have that  $\varphi_{k,j_\eta}(x_j) = 0$ . Hence, recalling that  $x_{j_\eta,\alpha_\eta} \equiv x_j$ , and  $\varphi_{k,j_\eta,\alpha_\eta-1} \equiv \varphi_{k,j_\eta}$ , we have

$$\widehat{\varphi}_{k,i}(x_j) = \sum_{\eta=1}^{\eta_{\max}} \left( \prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta,\zeta}) \right).$$

Now, let's define the set

$$(2.11) \quad Z_j = \bigcup_{\gamma} \mathcal{E}_k^\gamma,$$

where each  $\mathcal{E}_k^\gamma$  in the above union is such that  $x_j$  belongs to the closure of  $\mathcal{E}_k^\gamma$ . We observe that  $\mathcal{E}_k^\gamma$  (and so  $Z_j$ ) is independent of  $\eta$  and that  $e_j \subseteq \mathcal{E}_k^\gamma$  for some  $\gamma$ . Moreover,  $\lim_{x \in \Omega \rightarrow x_j} \widehat{\varphi}_{k,i}(x) = \lim_{x \in Z_j \rightarrow x_j} \widehat{\varphi}_{k,i}(x)$ , therefore, to show existence of the limit  $\lim_{x \in \Omega \rightarrow x_j} \widehat{\varphi}_{k,i}(x)$  we have to show that the limits  $\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \widehat{\varphi}_{k,i}(x)$  all exist and are equal for all  $\gamma$ . On every set  $\mathcal{E}_k^\gamma$ , by definition, exactly one of the following two cases can happen:

1.  $x_j \in X_k^\gamma$ ,  $x_j \notin \mathcal{E}_k^\gamma$ , and  $\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}(x) = 0$  for all  $\eta$ .

The above limit is obtained for the following reason: If  $x_{j_\eta} \in X_k^\gamma$ , then  $\varphi_{k,j_\eta} \equiv \varphi_{k,j_\eta}^\gamma$  on  $\mathcal{E}_k^\gamma$  for all  $\eta$ , and it is true that  $\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}^\gamma(x) = 0$ .

If  $x_{j_\eta} \notin X_k^\gamma$ , then  $\varphi_{k,j_\eta} \equiv 0$  on  $\mathcal{E}_k^\gamma$ , so the same result is verified. Note that since  $x_j \in X_k^\gamma$ , from Definition 2.8 we have that  $\varphi_{k,j} \equiv \varphi_{k,j}^\gamma$  on  $\mathcal{E}_k^\gamma$ .

2.  $x_j \notin X_k^\gamma$ ,  $x_j \in \mathcal{E}_k^\gamma$  and  $\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}(x) = \varphi_{k,j_\eta}(x_j)$  for all  $\eta$ .

In this case,  $x_{j_\eta} \in X_k^\gamma$  always, and we have  $\varphi_{k,j_\eta} \equiv \varphi_{k,j_\eta}^\gamma$  on  $\mathcal{E}_k^\gamma$  for all  $\eta$ . Nodes  $x_{j_\eta} \notin X_k^\gamma$  on  $e_j$  cannot exist. If they did, then  $x_j$  would be uncle or great-uncle of itself but such nodes do not belong to the sets  $\tilde{\mathcal{H}}_{x_i}^{\alpha,\beta}$ , so this cannot happen. Moreover, since  $x_j \notin X_k^\gamma$  and  $x_j \in \mathcal{E}_k^\gamma$ , Definition 2.8 gives  $\varphi_{k,j} \equiv 0$  on  $\mathcal{E}_k^\gamma$ .

Hence, let  $\mathcal{E}_k^\gamma$  be given and assume we are in case 1. Then

$$\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}(x) = 0 \text{ for all } \eta, \quad \lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j}(x) = \lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j}^\gamma(x) = \delta_{jj} = 1.$$

Referring to (2.10) this implies that

$$\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \hat{\varphi}_{k,i}(x) = \sum_{\eta=1}^{\eta_{\max}} \left( \prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta}, \zeta) \right).$$

In case 2, we have for all  $\eta$ ,

$$\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}(x) = \lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j_\eta}^\gamma(x) = \varphi_{k,j_\eta}^\gamma(x_j) = \varphi_{k,j_\eta}(x_j), \quad \lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \varphi_{k,j}(x) = 0.$$

Again referring to (2.10) this gives

$$\lim_{x \in \mathcal{E}_k^\gamma \rightarrow x_j} \hat{\varphi}_{k,i}(x) = \sum_{\eta=1}^{\eta_{\max}} \left( \prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta}, \zeta) \right).$$

This proves that  $\lim_{x \in \Omega \rightarrow x_j} \hat{\varphi}_{k,i}(x)$  exists and

$$\hat{\varphi}_{k,i}(x_j) = \sum_{\eta=1}^{\eta_{\max}} \left( \prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta}, \zeta) \right) = \lim_{x \in \Omega \rightarrow x_j} \hat{\varphi}_{k,i}(x).$$

Hence, the continuity of  $\hat{\varphi}_{k,i}$  at any hanging node  $x_j$  of  $x_i$  is proved.

To complete the proof, we have to show that  $\hat{\varphi}_{k,i}$  is continuous on all edges or faces in the triangulation  $\mathcal{T}_k$  that contain at least one hanging node. Continuity at any other point in the domain is inherited from the continuity of  $\varphi_{k,i}$ . Let  $e_j$  be such an edge or face that contains  $p$  nodes  $x_{j_\ell}$  with  $\ell = 1, \dots, p$ , where at least one of them is a hanging node. Then there exist at least two elements  $\Omega_c \in \mathcal{T}_k^c$  and  $\Omega_f \in \mathcal{T}_k^f$  such that

- $e_j = \Omega_c \cap \Omega_f$ ,
- $\Omega_c \subseteq \mathcal{E}_k^c$  and  $\Omega_f \subseteq \overline{\mathcal{E}_k^f}$ ,

where  $\overline{\mathcal{E}_k^f}$  denotes the closure of  $\mathcal{E}_k^f$ . We remark that the superscripts  $c$  and  $f$  stand for coarse and fine, respectively. If there exist more than two elements that share the same  $e_j$  (this is possible if  $e_j$  is an edge in a three-dimensional triangulation) then we consider them in pairs, where  $f$  is fixed and always refers to the element on the finest triangulation while  $c$  spans the remaining elements, one at a time.

Let us define the following function:

$$h(x) := \begin{cases} \hat{\varphi}_{k,i}(x) & \text{if } x \in \text{int}(\Omega_f), \\ \lim_{t \in \text{int}(\Omega_f) \rightarrow x} \hat{\varphi}_{k,i}(t) & \text{if } x \in e_j. \end{cases}$$

Then  $h$  is a polynomial function on its domain. Let  $h|_{e_j}$  be the trace of  $h$  on  $e_j$ . Its value on  $e_j$  is uniquely determined by the values of  $h$  at the  $p$  interface nodes. Namely, the space of the traces, referred to as  $V_k^f|_{e_j} \subseteq H^{1/2}(e_j)$ , has dimension  $p$  and

$$h|_{e_j}(x) = \sum_{\ell=1}^p h(x_{j_\ell}) \varphi_{k,j_\ell}^f|_{e_j}(x).$$

Let  $g(x) := \widehat{\varphi}_{k,i}(x)$  for all  $x \in \Omega_c$ . The function  $g$  is also a polynomial function on its domain. Let  $g|_{e_j}$  be the trace of  $g$  on  $e_j$  and  $V_k^c|_{e_j}$  the corresponding space of the traces. Since  $g|_{e_j} \in V_k^c|_{e_j} \subseteq V_k^f|_{e_j}$ , the function  $g|_{e_j}$  can also be represented with the bases of  $V_k^f|_{e_j}$ , namely,

$$g|_{e_j}(x) = \sum_{\ell=1}^p g(x_{j_\ell}) \varphi_{k,j_\ell}^f|_{e_j}(x).$$

Recalling that by continuity of  $\widehat{\varphi}_{k,i}$  at the  $p$  nodes, on  $e_j$  we also have

$$h(x_{j_\ell}) = \widehat{\varphi}_{k,i}(x_{j_\ell}) = g(x_{j_\ell}) \text{ for all } \ell = 1, \dots, p,$$

and, consequently,  $h(x) = g(x)$  for all  $x \in e_j$ . Therefore, for all  $x \in e_j$

$$\lim_{t \in \text{int}(\Omega_f) \rightarrow x} \widehat{\varphi}_{k,i}(t) = h(x) = g(x) = \widehat{\varphi}_{k,i}(x) = \lim_{t \in \text{int}(\Omega_c) \rightarrow x} \widehat{\varphi}_{k,i}(t).$$

Thus,  $\widehat{\varphi}_{k,i}$  is continuous at every face or edge  $e_j$  that contains at least one hanging node. This completes the proof.  $\square$

**LEMMA 2.16.** *The set  $\{\widehat{\varphi}_{k,i} : x_i \in \mathcal{I}_k \cup \mathcal{M}_k\}$  is linearly independent.*

*Proof.* If  $x_i \in \mathcal{I}_k$ , then  $\widehat{\varphi}_{k,i} = \varphi_{k,i}$  and the statement follows from (2.5). If  $x_i \in \mathcal{M}_k$  then, according to (2.9), each  $\widehat{\varphi}_{k,i}$  is a linear combination of one master node function  $\varphi_{k,i}$ , which satisfies the delta property for all  $x_j \notin \mathcal{H}_k$ , and several hanging node functions  $\varphi_{k,j}$ , which satisfy the zero property (2.6). Hence, letting  $x_j \notin \mathcal{H}_k$ , we have

$$(2.12) \quad \widehat{\varphi}_{k,\mathcal{M}_k}(x_j) = \varphi_{k,\mathcal{M}_k}(x_j) + \Phi_k \varphi_{k,\mathcal{H}_k}(x_j) = \delta_{ij} \mathbf{1} + \mathbf{0}.$$

This means that

$$(2.13) \quad \widehat{\varphi}_{k,i}(x_j) = \delta_{ij} \text{ for all } x_j \in \mathcal{I}_k \cup \mathcal{M}_k.$$

Since the  $\widehat{\varphi}_{k,i}$  have the delta property, the set  $\{\widehat{\varphi}_{k,i} : x_i \in \mathcal{I}_k \cup \mathcal{M}_k\}$  is linearly independent.  $\square$

**DEFINITION 2.17.** *Let  $\widehat{N}_k$  be the total number of interior and master nodes. We define the set  $\widehat{V}_k$  initially introduced in (2.1) to be the set spanned by  $\{\widehat{\varphi}_{k,i}\}_{i=1}^{\widehat{N}_k}$ , namely,  $\widehat{V}_k \equiv \text{span}(\{\widehat{\varphi}_{k,i}\}_{i=1}^{\widehat{N}_k})$ . Because of Lemma 2.16, it follows that  $\{\widehat{\varphi}_{k,i}\}_{i=1}^{\widehat{N}_k}$  also forms a basis for  $\widehat{V}_k$ , where  $\widehat{V}_k$  is a vector space with the standard addition and scalar multiplication for real valued functions.*

*Remark 7.* To the unique representation of an element  $\widehat{v} \in \widehat{V}_k$ ,

$$\widehat{v} = \sum_{i:x_i \in \mathcal{I}_k \cup \mathcal{M}_k} \widehat{v}_i \widehat{\varphi}_{k,i},$$

we artificially add the hanging node functions, which are identically zero, i.e.,

$$\widehat{v} = \sum_{i:x_i \in \mathcal{I}_k \cup \mathcal{M}_k} \widehat{v}_i \widehat{\varphi}_{k,i} + \sum_{i:x_i \in \mathcal{H}_k} \widehat{v}_i \widehat{\varphi}_{k,i}.$$

The new representation of  $\widehat{v}$  is not unique anymore, since the coefficients  $\widehat{v}_i$  associated with the zero hanging node functions are arbitrary. This choice may seem odd, but we again emphasize that it is motivated by the fact that in the numerical implementation of the algorithm we want to preserve the same dimensions between the arrays and matrices associated with the spaces  $V_k$  and  $\widehat{V}_k$ .

**PROPOSITION 2.18.** *The space  $\widehat{V}_k$  is a subspace of  $V_k$ .*

*Proof.* The proof is immediate, since each function  $\widehat{\varphi}_{k,i}$  is constructed as a linear combination of functions  $\varphi_{k,i}$ , which belong to a basis for  $V_k$ .  $\square$

**PROPOSITION 2.19.** *Let  $a \in V_k$  and  $\widehat{b} \in \widehat{V}_k$ . Let  $\mathbf{a} = [\mathbf{a}_{\mathcal{I}}, \mathbf{a}_{\mathcal{M}}, \mathbf{a}_{\mathcal{H}}]^T$  and  $\widehat{\mathbf{b}} = [\widehat{\mathbf{b}}_{\mathcal{I}}, \widehat{\mathbf{b}}_{\mathcal{M}}, \widehat{\mathbf{b}}_{\mathcal{H}}]^T$  be the coefficient representation vectors of  $a$  and  $\widehat{b}$ , i.e.,*

$$a = \varphi_k^T \mathbf{a} \quad \text{and} \quad \widehat{b} = \widehat{\varphi}_k^T \widehat{\mathbf{b}}.$$

*Then  $a = \widehat{b}$  if and only if  $\mathbf{a}_{\mathcal{I}} = \widehat{\mathbf{b}}_{\mathcal{I}}$ ,  $\mathbf{a}_{\mathcal{M}} = \widehat{\mathbf{b}}_{\mathcal{M}}$ , and  $\mathbf{a}_{\mathcal{H}} = \Phi_k^T \mathbf{a}_{\mathcal{M}}$ .*

*Proof.* The proof of the proposition follows from (2.9).  $\square$

*Remark 8.* Note that the above proposition implies that the equality of  $a \in V_k$  and  $\widehat{b} \in \widehat{V}_k$  is independent of the value of  $\widehat{\mathbf{b}}_{\mathcal{H}}$ . Moreover if  $a \in V_k$  is such that  $\mathbf{a} = [\mathbf{a}_{\mathcal{I}}, \mathbf{a}_{\mathcal{M}}, \Phi_k^T \mathbf{a}_{\mathcal{M}}]^T$ , then  $a$  is also in  $\widehat{V}_k$  and we can choose the same vector  $\mathbf{a}$  as a representation of  $a$  in  $\widehat{V}_k$ .

## 2.5. The interspace operators.

**DEFINITION 2.20.** *The prolongation operator  $\widehat{\mathcal{P}}_k : \widehat{V}_k \rightarrow V_k$  is the natural injection and its action on  $\widehat{v}$  is given by*

$$\widehat{\mathcal{P}}_k \widehat{v} = \varphi_k^T (\widehat{P}_k \widehat{\mathbf{v}}),$$

where  $\widehat{P}_k$  is the matrix representation of  $\widehat{\mathcal{P}}_k$ .

**PROPOSITION 2.21.** *Let  $\widehat{v}$  be given as a linear combination of the basis of  $\widehat{V}_k$  as in Remark 7. In vector notation*

$$\widehat{v} = \widehat{\varphi}_k^T \widehat{\mathbf{v}}$$

*for some coefficient vector  $\widehat{\mathbf{v}} = [\widehat{v}_1, \widehat{v}_2, \dots, \widehat{v}_{N_k}]^T \in \mathbb{R}^{N_k}$ . Then, for the matrix representation of  $\widehat{\mathcal{P}}_k$  we have  $\widehat{P}_k = \widehat{R}_k^T$ , where  $\widehat{R}_k$  is the matrix from (2.9).*

*Proof.* Since  $\widehat{v} \in \widehat{V}_k \subseteq V_k$ , its prolongation into  $V_k$  is the natural injection, namely,

$$\varphi_k^T (\widehat{P}_k \widehat{\mathbf{v}}) = \widehat{\mathcal{P}}_k \widehat{v} = \widehat{v} = \widehat{\varphi}_k^T \widehat{\mathbf{v}} = (\widehat{R}_k \varphi_k)^T \widehat{\mathbf{v}} = \varphi_k^T (\widehat{R}_k^T \widehat{\mathbf{v}}) \text{ for all } \widehat{v} \in \widehat{V}_k.$$

Thus, the matrix representation of  $\widehat{\mathcal{P}}_k$  is given by  $\widehat{P}_k = \widehat{R}_k^T$ .  $\square$

**DEFINITION 2.22.** *Let  $\langle \cdot, \cdot \rangle$  denote the  $L^2(\Omega)$  inner product. The restriction operator  $\widehat{\mathcal{R}}_k : V_k \rightarrow \widehat{V}_k$  is defined as the adjoint of  $\widehat{\mathcal{P}}_k$  with respect to the  $L^2(\Omega)$  inner product. In other words*

$$\langle \widehat{\mathcal{R}}_k v, \widehat{u} \rangle = \langle v, \widehat{\mathcal{P}}_k \widehat{u} \rangle \text{ for all } v \in V_k, \widehat{u} \in \widehat{V}_k.$$

*Clearly, its matrix representation is the matrix  $\widehat{R}_k$  from (2.9).*

Note that if we choose  $\hat{u} = \hat{\varphi}_{k,i}$  for some  $i$ , then

$$\hat{\mathcal{P}}_k \hat{\varphi}_{k,i} = \hat{\varphi}_{k,i} = \hat{R}_{k,i} \boldsymbol{\varphi}_k,$$

where  $\hat{R}_{k,i}$  corresponds to the  $i$ th row of the matrix  $\hat{R}_k$  defined in (2.9). Consequently, for any  $v$  in  $V_k$

$$(2.14) \quad \langle \hat{\mathcal{R}}_k v, \hat{\varphi}_{k,i} \rangle = \langle v, \hat{\mathcal{P}}_k \hat{\varphi}_{k,i} \rangle = \langle v, \hat{R}_{k,i} \boldsymbol{\varphi}_k \rangle = \sum_{j=1}^{N_k} \hat{R}_{k,ij} \langle v, \varphi_{k,j} \rangle.$$

Let  $g \in H^{-1}(\Omega)$  be given and define in an entrywise fashion the vectors  $\hat{\mathbf{f}}$  and  $\mathbf{f}$ ,

$$\hat{\mathbf{f}}_i(g) = \langle g, \hat{\varphi}_{k,i} \rangle \text{ and } \mathbf{f}_i(g) = \langle g, \varphi_{k,i} \rangle,$$

for all  $i = 1, \dots, N_k$ . Then (2.14) can be rewritten as

$$\hat{\mathbf{f}}_i(\hat{\mathcal{R}}_k v) = \sum_{j=1}^{N_k} \hat{R}_{k,ij} \mathbf{f}_j(v) = \hat{R}_{k,i} \mathbf{f}(v)$$

for all  $i = 1, \dots, N_k$ . Then in matrix-vector notation, for any  $v$  in  $V_k$

$$\hat{\mathbf{f}}(\hat{\mathcal{R}}_k v) = \hat{R}_k \mathbf{f}(v).$$

**LEMMA 2.23.** *For any bilinear form  $a(\cdot, \cdot)$ , define the matrices  $A_k$  and  $\hat{A}_k$  by*

$$\begin{aligned} A_{k,ij} &= a(\varphi_{k,i}, \varphi_{k,j}) \text{ for all } i, j = 1, \dots, N_k, \\ \hat{A}_{k,ij} &= a(\hat{\varphi}_{k,i}, \hat{\varphi}_{k,j}) \text{ for all } i, j = 1, \dots, N_k. \end{aligned}$$

*Then  $\hat{A}_k = \hat{R}_k A_k \hat{R}_k^\top = \hat{R}_k A_k \hat{P}_k$ . Namely, with a slight abuse of terminology,  $\hat{A}_k$  is the restriction of  $A_k$  to the space  $\hat{V}_k$ .*

*Proof.* For all  $i, j$

$$\begin{aligned} \hat{A}_{k,ij} &= a(\hat{\varphi}_{k,i}, \hat{\varphi}_{k,j}) = a(\hat{R}_{k,i} \boldsymbol{\varphi}_k, \hat{R}_{k,j} \boldsymbol{\varphi}_k) = \sum_{l=1}^{N_k} \sum_{m=1}^{N_k} \hat{R}_{k,il} a(\varphi_{k,l}, \varphi_{k,m}) \hat{R}_{k,jm} \\ &= \sum_{l=1}^{N_k} \sum_{m=1}^{N_k} \hat{R}_{k,il} A_{k,lm} \hat{R}_{k,jm} = \sum_{l=1}^{N_k} \sum_{m=1}^{N_k} \hat{R}_{k,il} A_{k,lm} \hat{R}_{k,mj}^\top = \hat{R}_{k,i} A_k \hat{R}_{k,j}^\top, \end{aligned}$$

thus the result follows.  $\square$

The rows and the columns corresponding to the hanging nodes are not removed but zeroed. In order to get a nonsingular matrix, the diagonal entries of  $\hat{A}_k$  corresponding to the hanging nodes are set equal to one.

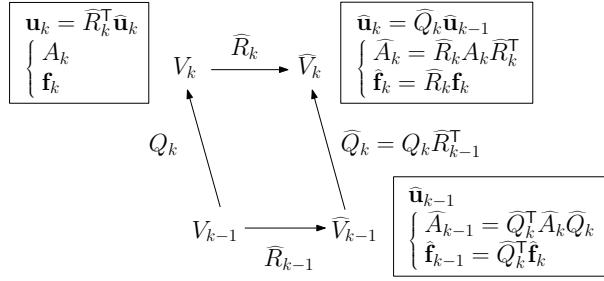
Note that from Definitions 2.8 and 2.14 it follows that

$$V_1 \subseteq V_2 \subseteq \dots \subseteq V_k \quad \text{and} \quad \hat{V}_1 \subseteq \hat{V}_2 \subseteq \dots \subseteq \hat{V}_k.$$

For this reason, prolongation operators can be defined.

**DEFINITION 2.24.** *The prolongation operator  $\mathcal{Q}_k : V_{k-1} \rightarrow V_k$  is the natural injection between  $V_{k-1}$  and  $V_k$ , while the prolongation operator  $\hat{\mathcal{Q}}_k : \hat{V}_{k-1} \rightarrow \hat{V}_k$  is the natural injection between  $\hat{V}_{k-1}$  and  $\hat{V}_k$ . The respective matrix representations are denoted by  $Q_k$  and  $\hat{Q}_k$ .*

Since  $\hat{V}_{k-1} \subseteq V_{k-1} \subseteq V_k$ , it immediately follows that the prolongation operator  $\mathcal{Q}_k \hat{\mathcal{P}}_{k-1} : \hat{V}_{k-1} \rightarrow V_k$  with matrix representation given by  $Q_k \hat{P}_{k-1}$  is the natural injection between  $\hat{V}_{k-1}$  and  $V_k$ .

FIG. 10. *Interspace operators.*

LEMMA 2.25. *The natural injection from  $\hat{V}_{k-1}$  to  $V_k$  can be obtained as a composition of the two prolongations  $\hat{\mathcal{P}}_{k-1} : \hat{V}_{k-1} \rightarrow V_{k-1}$  and  $\mathcal{Q}_k : V_{k-1} \rightarrow V_k$ :*

$$a = \mathcal{Q}_k(\hat{\mathcal{P}}_{k-1}(\hat{v}_{k-1})) = \hat{v}_{k-1} \text{ for all } \hat{v}_{k-1} \in \hat{V}_{k-1},$$

or in vector-matrix notation  $\mathbf{a} = Q_k \hat{P}_{k-1} \hat{\mathbf{v}}_{k-1}$ , where  $\mathbf{a}$  is the vector representation of  $a$  in  $V_k$ .

PROPOSITION 2.26. *The matrix  $Q_k \hat{P}_{k-1}$  can also be used as matrix representation of the prolongation  $\hat{\mathcal{Q}}_k : \hat{V}_{k-1} \rightarrow \hat{V}_k$ .*

*Proof.* Let  $\hat{\mathbf{v}}_{k-1}$  be the vector representation of any  $\hat{v}_{k-1} \in \hat{V}_{k-1}$ . We want to show that if  $\hat{\mathbf{b}} \equiv Q_k \hat{P}_{k-1} \hat{\mathbf{v}}_{k-1}$ , then the following equality holds,

$$\hat{\mathbf{b}} \equiv \hat{\varphi}_k^T \hat{\mathbf{b}} = \hat{v}_{k-1}.$$

From Lemma 2.25, the vector  $\mathbf{a} = Q_k \hat{P}_{k-1} \hat{\mathbf{v}}_{k-1}$  is the vector representation of  $a \in V_k$  with  $a = \hat{v}_{k-1}$ . Since  $a = \hat{v}_{k-1} \in \hat{V}_{k-1} \subseteq \hat{V}_k \subseteq V_k$ ,  $a$  is also an element of  $\hat{V}_k$ . According to Remark 8 we can choose  $\mathbf{a}$  as a vector representation of  $a$  in  $\hat{V}_k$ .

Then, we have two elements  $\hat{\mathbf{b}}$  and  $a$  in  $\hat{V}_k$  with the same vector representation,  $\hat{\mathbf{b}} = \mathbf{a}$ , thus they are equal. This means  $\hat{\mathbf{b}} = a = \hat{v}_{k-1}$ .  $\square$

Similarly, as we did before for the nested spaces  $V_k$  and  $\hat{V}_k$ , we can show that the matrix representation of the restriction operator between  $\hat{V}_k$  and  $\hat{V}_{k-1}$  is  $\hat{Q}_k^T$ , the transpose of the matrix representation of the prolongation operator.

LEMMA 2.27. *For any bilinear form  $a(\cdot, \cdot)$ , let  $\hat{A}_k$  be the matrix defined in Lemma 2.23 and let  $\hat{A}_{k-1}$  be*

$$\hat{A}_{k-1,ij} = a(\hat{\varphi}_{k-1,i}, \hat{\varphi}_{k-1,j}) \text{ for all } i, j = 1, \dots, N_{k-1}.$$

*Then,  $\hat{A}_{k-1} = \hat{Q}_k^T \hat{A}_k \hat{Q}_k$ , namely, with a slight abuse of terminology,  $\hat{A}_{k-1}$  is the restriction of  $\hat{A}_k$  to the space  $\hat{V}_{k-1}$ .*

*Proof.* The proof is similar to the one in Lemma 2.23.  $\square$

Once again, the diagonal entries of  $\hat{A}_{k-1}$  corresponding to the hanging nodes are set equal to one.

Schematics of all the interspace operators defined above, and how to use them, are given in Figure 10. All the prolongation and restriction operators are global and are implemented in the FEMuS library [1] using PETSc sparse parallel matrices [5].

The restriction operator  $\widehat{R}_k$  from  $V_k$  to  $\widehat{V}_k$  is precomputed when the irregular mesh is generated since it depends only on the mesh and the finite-element family. Similarly, the prolongation operator  $Q_k$  from  $V_{k-1}$  to  $V_k$  is precomputed once the meshes at the two consecutive refinement levels are available, since it also depends only on the mesh and the finite-element family. The prolongation operator  $\widehat{Q}_k$  is given by the product between the two matrices and is precomputed as well.

**2.6. The numerical algorithm.** Once the interspace operators are available in terms of matrices, the numerical solution of the problem  $a(u, v) = \langle f, v \rangle$  is sought in the continuous space  $\widehat{V}_J$ . Namely, we seek the solution of the discretized problem  $\widehat{A}_J \widehat{\mathbf{u}}_J = \widehat{\mathbf{f}}_J$ . We do not work directly in the continuous space  $\widehat{V}_J$ . We rather work in the discontinuous space  $V_J$  and use the restriction operator  $\widehat{R}_J$  to move objects between spaces. Since an iterative solver is used the above system is rewritten in its residual form.

For  $i = 0$ , let  $\widehat{\mathbf{u}}_J^0$  be the initial guess, then

1. assemble the matrix  $A_J$  in  $V_J$  using the test functions  $\varphi_J$ ;
2. restrict  $A_J$  in the  $\widehat{V}_J$  space:  $\widehat{A}_J = \widehat{R}_J A_J \widehat{R}_J^\top$ ;
3. prolong the current solution in the  $V_J$  space:  $\mathbf{u}_J^i = \widehat{R}_J^\top \widehat{\mathbf{u}}_J^i$ ;
4. assemble the residual in  $V_J$ :  $\mathbf{r}_J^i = \mathbf{f}_J - A_J \mathbf{u}_J^i$ ;
5. restrict the residual to  $\widehat{V}_J$ :  $\widehat{\mathbf{r}}_J^i = \widehat{R}_J \mathbf{r}_J^i$ ;
6. let  $\widehat{D}_J$  be an *easy to invert* approximation of the matrix  $\widehat{A}_J$ , then solve

$$\widehat{D}_J \widehat{\mathbf{w}}_J^i = \widehat{\mathbf{r}}_J^i, \text{ and set } \widehat{\mathbf{u}}_J^{i+1} = \widehat{\mathbf{u}}_J^i + \widehat{\mathbf{w}}_J^i;$$

7. if  $\|\widehat{\mathbf{w}}_J^i\| \leq \varepsilon$  exit, otherwise set  $i = i + 1$  and go back to step 3.

A few remarks on the above algorithm.

We construct the matrix  $A_J$  and the residual  $\mathbf{r}_J$  in  $V_J$ , since the assembling is done elementwise following the standard finite-element approach for unstructured grids. Moreover, let  $x_i, x_j$  in  $X_J^l$  be two nodal points belonging to a generic element  $E$  of the triangulation  $\mathcal{T}_J^l$ . The two pairs of test functions  $\varphi_{J,i}, \varphi_{J,j}$  and  $\varphi_{J,i}^l, \varphi_{J,j}^l$  differ at most on the boundary of the element  $E$ . Then on  $E$ ,

$$a(\varphi_{J,i}, \varphi_{J,j})|_E = a(\varphi_{J,i}^l, \varphi_{J,j}^l)|_E \quad \text{and} \quad \langle f, \varphi_{J,i} \rangle|_E = \langle f, \varphi_{J,i}^l \rangle|_E$$

for all  $f \in H^{-1}(\Omega)$ . Thus to construct  $A_J$  and  $\mathbf{r}_J$  elementwise we use the standard test functions  $\varphi_{J,i}^l$ , which is a very convenient approach since no new bases have to be constructed.

The operator  $\widehat{D}_J$  is abstract, and its inversion can be interpreted as the action of a linear algebra solver which in turn can also be preconditioned. In our applications, we used geometric multigrid either as a solver (in section 3) or as a preconditioner for the conjugate gradient (CG) method and the generalized minimal residual (GMRES) (in sections 4 and 5). For this purpose we used the PETSc toolkit [5] as a black box, providing as inputs the matrices  $\widehat{A}_k$  for  $k = J, \dots, 0$ , the matrices  $\widehat{Q}_k$  for  $k = J, \dots, 1$ , and the residual vector  $\widehat{\mathbf{r}}_J^i$ , and letting PETSc compute  $\widehat{\mathbf{w}}_J^i$ . This algebraic approach is convenient and elegant since all the burden of dealing with the hanging nodes at different level triangulations  $\mathcal{T}_k$  has been incorporated in the prolongation operator  $\widehat{Q}_k$  and no ad hoc multigrid solver or preconditioner needs to be implemented.

The above algorithm can be easily extended to nonlinear problems. In this case the residual  $\mathbf{r}_J^i$  at step 4 is a more complex function of  $\mathbf{u}_J^i$  and  $\widehat{D}_J$  is an easy to invert

approximation of the tangent matrix

$$\hat{J}_J(\hat{\mathbf{u}}_J^i) = \hat{R}_J \left( -\frac{\partial \mathbf{r}_J^i}{\partial \mathbf{u}_J^i} \right) \hat{R}_J^T.$$

This corresponds to a Newton–Raphson iterative scheme [26]. Such a scheme is used in the last example where a Navier–Stokes flow is considered.

**3. Eigenvalue analysis of the multigrid method.** Historically, the theory of multigrid methods has been developed for linear equations of the form  $Au = f$ , where the operator  $A$  is symmetric and positive definite [23, 9, 41, 10, 40, 12]. Other types of boundary value problems have also been addressed, where  $A$  is nonsymmetric or indefinite [29, 34, 39, 13].

In [12], the authors obtained convergence estimates for a multigrid algorithm without making regularity assumptions on the solution. This was a breakthrough in the theoretical analysis of multigrid, since previous convergence estimates used to rely on both a smoothing property, and an approximation property, where the latter is typically obtained assuming a certain degree of regularity. Moreover, in the same paper, it was shown that convergence of the multigrid algorithm could also be obtained when smoothing is performed only on a subspace of the multigrid space at every level. This allows applying the theory proposed in [12] to local midpoint refinement applications, where the smoothing is performed on subspaces that do not have degrees of freedom associated with the interface nodes. This includes both hanging and master nodes. A weaker convergence estimate with respect to the classical multigrid convergence [14] was obtained, in the sense that the error bound  $\delta_J$  for the multigrid error  $E_J$  depended on the number of multigrid levels  $J$ ,

$$a(E_J v, v) \leq \delta_J a(v, v) \text{ with } \delta_J = 1 - (C J)^{-1},$$

where  $a(\cdot, \cdot)$  is the energy norm and  $C$  is a constant independent of  $J$ . It is clear when looking at  $\delta_J$  that, from a theoretical point of view, increasing the pre- and/or post-smoothing iterations at each level does not guarantee an improvement of the convergence bound. From a practical point of view as well, since smoothing is done only on subspaces of the actual multigrid spaces, an increase in the number of smoothing steps can only improve the convergence bound up to some saturation value. In this work, using the continuous nodal basis functions in Definition 2.14, we are allowed to perform the smoothing procedure on the entire multigrid space. This results in an improved convergence when the number of smoothing steps is increased, as it will be shown from the numerical results. For the rest of the paper, we refer to the local smoothing approach outlined in [12] as the BPWX algorithm after the authors: Bramble, Pasciak, Wang, and Xu. The BPWX approach performs smoothing on subspaces that do not contain degrees of freedom associated with interface nodes.

In this section, we compare the performance of our algorithm with the BPWX, and show that since our multigrid performs a global smoothing on all nodes at a given multigrid level, the drawbacks of the BPWX method are eliminated.

We do so by evaluating the spectral radius of  $E_J = I - \hat{D}_J^{-1} \hat{A}_J$ , where the matrix  $\hat{D}_J^{-1}$  corresponds to one V-cycle of either the BPWX multigrid or the one proposed in this work. For this purpose, we use the method developed in [30].

Consider one iteration for the algorithm described in section 2.6. The solution at the first iteration step is given by

$$\hat{\mathbf{u}}_J^1 = \hat{\mathbf{u}}_J^0 + \hat{\mathbf{w}}_J^1 = (I - \hat{D}_J^{-1} \hat{A}_J) \hat{\mathbf{u}}_J^0 + \hat{D}_J^{-1} \hat{\mathbf{f}}_J.$$

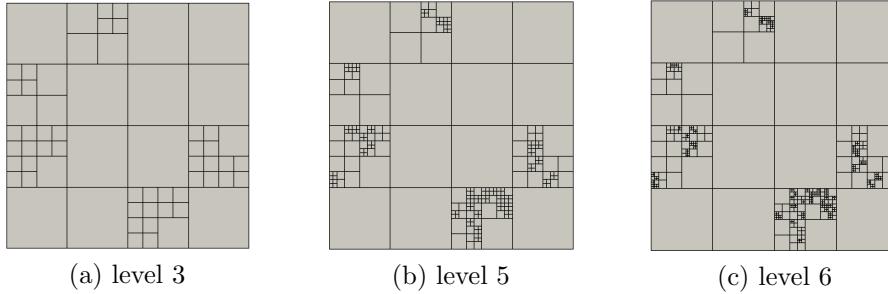


FIG. 11. *Refinement of the quadrilateral element mesh: 50% of the elements from the previous level are randomly selected for refinement.*

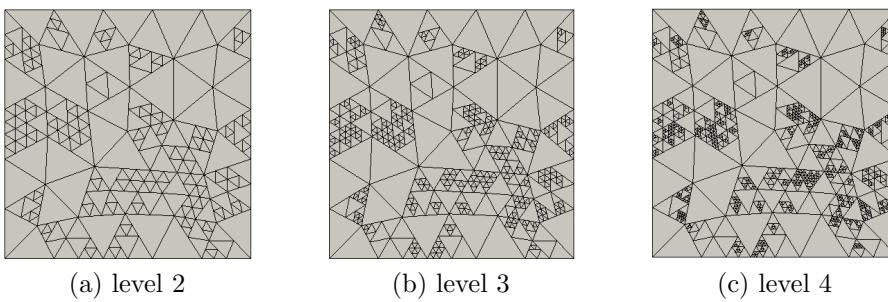


FIG. 12. *Refinement of the triangular element mesh: 50% of the elements from the previous level are randomly selected for refinement.*

Let  $\hat{\mathbf{u}}_J^0$  be zero, impose zero Dirichlet boundary conditions for  $\hat{\mathbf{u}}_J^1$ , and set  $\hat{\mathbf{f}}_J = \mathbf{e}_l$  with entries one at the  $l$ th row and zero elsewhere. Then, the system becomes

$$\hat{\mathbf{u}}_{J,l}^1 = \hat{D}_J^{-1} \mathbf{e}_l \quad \text{for } l = 1, \dots, N_J.$$

The inverse of the multigrid matrix is obtained using the solution  $\hat{\mathbf{u}}_{J,l}^1$  as the  $l$ th column of  $\hat{D}_J^{-1}$ . The spectrum of  $\hat{D}_J^{-1} \hat{A}_J$  follows.

We remark that in the following examples only one iteration of the multigrid V-cycle is carried out. At each level, we employ a Richardson smoother with damping factor 0.6, preconditioned with ILU. At the coarse level, a direct method is employed. Tests are made considering the bilinear operator  $a(u, v) = \langle \nabla u, \nabla v \rangle$ . We use bi-quadratic and tri-quadratic Lagrangian test functions in two and three dimensions, respectively, and study the performances of the two methods with respect to different element types and different numbers of pre- and post-smoothing iterations.

Convergence of the multigrid solver is obtained only if the spectral radius of  $I - \hat{D}_J^{-1} \hat{A}_J$  is less than one. Moreover, the smaller such a quantity is, the better  $\hat{D}_J^{-1}$  approximates the inverse of  $A_J$  and the faster the solver converges to the solution.

We first consider two types of elements for the two-dimensional unit square geometry: quadrilaterals and triangles. The criterion for the local selective refinement is that 50% of the elements from the previous level are randomly selected for refinement. For each element type we study the three irregular triangulations depicted in Figures 11 and 12. In the quadrilateral case, the coarse mesh consists of a  $2 \times 2$  regular triangulation, that is uniformly refined once and then selectively refined 2, 4, and 5 times, for a total of 323, 1201, and 2265 degrees of freedom, respectively. See cases (a), (b), and (c) in Figure 11. In the triangular case, the coarse mesh has 62

TABLE 1

*Spectral radius considering the meshes depicted in Figures 11 and 12. One pre- and post-smoothing iteration is considered.*

Methods	Proposed method	BPWX method	Quad.	Tri.
L	Quad.	Tri.	Quad.	Tri.
(a)	1.75e-2	5.35e-2	1.37e-1	3.03e-1
(b)	2.25e-2	5.79e-2	1.97e-1	3.04e-1
(c)	1.74e-2	7.87e-2	2.04e-1	3.04e-1

TABLE 2

*Spectral radius for several numbers of pre- and post-smoothing iterations. The chosen refined configuration corresponds to case (c) for both mesh types in Figures 11 and 12.*

Methods	Proposed method	BPWX method	Quad.	Tri.
Smoothing	Quad.	Tri.	Quad.	Tri.
1	1.74e-2	7.87e-2	2.04e-1	3.04e-1
2	5.80e-2	2.10e-2	2.02e-1	2.62e-1
4	2.10e-2	3.90e-3	2.02e-1	2.50e-1
8	9.16e-4	1.00e-3	2.02e-1	2.47e-1

elements that are selectively refined 2, 3, and 4 times, for a total of 1145, 2349, and 4603 degrees of freedom, respectively. See cases (a), (b), and (c) in Figure 12.

Both our method and the BPWX method are tested. Table 1 shows the spectral radius of  $I - \widehat{D}_J^{-1} \widehat{A}_J$  for the six cases in Figures 11 and 12, when only one pre- and one post-smoothing iteration is used. As the number of levels increases, the spectral radius has slight variations but does not change dramatically for both quadrilateral and triangular elements. More importantly, at each level, with the same element type, the spectral radius of the new method is 3 to 11 times smaller than the one in the BPWX method. Roughly speaking this indicates that our solver would converge 3 to 11 times faster than the BPWX.

Next we investigate the effect of the number of pre- and post-smoothing iterations on the spectral radius at the finest level for each method; see Table 2. Here we consider 1, 2, 4, and 8 symmetric pre- and post-smoothing iterations. With the proposed method, under the same element type, the spectral radius reduces 3 to 5 times as we double the number of iterations, while in the BPWX method it quickly saturates. Thus we expect that, with the proposed method, the number of linear iterations required by the solver to converge would decrease significantly, while they would stay the same for the BPWX method.

We conclude that for two-dimensional examples, our method always has an advantage over the BPWX in terms of convergence rate of the solver. Moreover, this advantage increases dramatically as the number of smoothing iterations increases.

We then investigate a three-dimensional example in a unit box geometry considering triangulations made of four types of elements: hexahedra, wedges, tetrahedra, and a combination of the three. The criterion for the local selective refinement remains the same as the two-dimensional case. The coarse grid consists of 8, 16, 6, and 20 elements for the hexahedra, wedges, tetrahedra, and the combination of the three, respectively. Each coarse grid is selectively refined 1, 2, and 3 times. The total number of degrees of freedom is 401, 1389, and 5087 for the hexahedral case, 506, 1817, 6538 for the wedge-shaped case, 146, 456, 1684 for the tetrahedral case, and 656, 2306, 8560 for the mixed case. Figure 13 shows the three-dimensional irregular triangulations for the hexahedral and mixed cases at level 3.

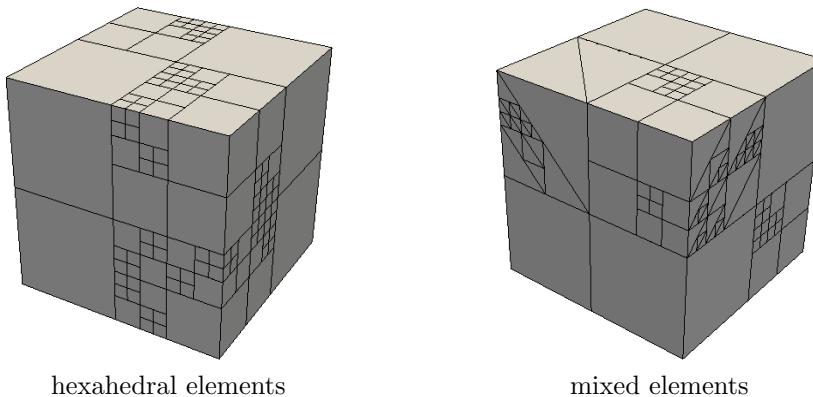


FIG. 13. Refinement of different types of elements starting with one uniform level and refining up to three nonuniform levels: 50% of the elements from the previous level are randomly selected for refinement.

TABLE 3

Spectral radius for different refined meshes. One pre- and one post-smoothing iteration is considered.

L	Proposed Method				BPWX Method			
	Hex.	Wedge	Tet.	Mixed	Hex.	Wedge	Tet.	Mixed
1	7.50e-3	3.95e-2	2.31e-2	3.58e-2	9.34e-2	2.08e-1	6.96e-2	1.59e-1
2	1.61e-2	5.71e-2	5.80e-2	1.01e-1	2.55e-1	2.07e-1	1.19e-1	2.29e-1
3	1.76e-2	6.25e-2	6.00e-2	1.12e-1	2.63e-1	2.98e-1	1.37e-1	3.17e-1

TABLE 4

Spectral radius for several numbers of pre- and post-smoothing iterations. The chosen refined configuration corresponds to level 3 for all mesh types.

S	Proposed method				BPWX method			
	Hex.	Wedge	Tet.	Mixed	Hex.	Wedge	Tet.	Mixed
1	1.76e-2	6.25e-2	6.00e-2	1.12e-1	2.63e-1	2.98e-1	1.37e-1	3.17e-1
2	7.20e-3	2.23e-2	1.53e-2	3.45e-2	2.63e-1	2.65e-1	1.29e-1	3.16e-1
4	3.00e-3	1.05e-2	6.20e-3	2.08e-2	2.63e-1	2.61e-1	1.29e-1	3.16e-1
8	6.64e-4	3.40e-3	1.80e-3	1.06e-2	2.63e-1	2.61e-1	1.29e-1	3.16e-1

The results for the spectral radius with one pre- and one post-smoothing iteration is listed in Table 3 for both the proposed method and the BPWX method. Notice that the spectral radius under the same type of elements does not change significantly from level 2 to 3. In some cases a larger variation occurs from level 1 to 2, that we attribute to the presence of relatively few degrees of freedom at level 1. At each level, considering the same element type, the spectral radius of the new method is 2 to 15 times smaller than the one obtained with the BPWX method. Roughly speaking this indicates that our solver would converge 2 to 15 times faster than the BPWX one. In particular the hexahedral case performs the best, while the mixed is 2–4 times better.

Next we investigate the effect of the number of pre- and post-smoothing iterations on the spectral radius at level 3; see Table 4. Again we consider 1, 2, 4, and 8 symmetric pre- and post-smoothing iterations. Under the same element type in our method the spectral radius reduces 2 to 5 times doubling the number of iterations, while in the BPWX method it is invariant. Thus, we expect that the number of linear iterations required by the solver to converge would decrease significantly with the proposed method, while it would stay the same for the BPWX method. Once

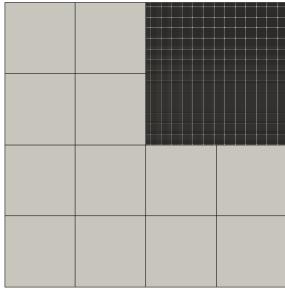


FIG. 14. Refinement of the first quadrant after 7 refinements.

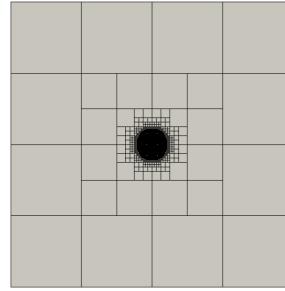


FIG. 15. Refinement of the circle after 8 refinements.

again the hexahedra perform the best, followed by the wedges, the tetrahedra, and the mixed ones.

**4. The Poisson problem.** We continue our investigation by considering the weak formulation of the Poisson problem discussed in [24]. Namely, find  $u_h \in V_h \subset H_0^1(\Omega)$  such that

$$(4.1) \quad (\nabla u_h, \nabla v_h) = (f, v_h), \text{ for all } v_h \in V_h$$

with  $\Omega = [-1, 1] \times [-1, 1]$  and  $f \equiv 1$ . The space  $V_h$  denotes an appropriate finite-element space. Note that the multigrid method proposed in [24] applies to 1-irregular grids, but the more general case of  $n$ -irregular triangulations is not addressed. Moreover, it has been implemented only for quadrilateral and hexahedral elements. Our method has no 1-irregularity limitations and can be easily applied to a variety of Lagrangian element types, other than quadrilateral and hexahedral.

We start with a coarse grid consisting of 4 elements, which are uniformly refined once. Two selective refinement strategies are used: Refine only the elements in the first quadrant ( $x \geq 0, y \geq 0$ ), and refine all the elements for which the centers are located within the circle of radius  $\frac{\pi}{4k}$  ( $k = 1, \dots, J - 1$ ) centered at the origin. Figure 14 shows the irregular triangulation of the first quadrant for  $J = 7$ . Figure 15 shows the irregular triangulation of the circle for  $J = 8$ .

Continuous piecewise-bilinear approximation is considered for both  $u_h$  and  $v_h$ . The linear system arising from the weak formulation in (4.1) defined on the above irregular triangulations is solved using a CG solver, preconditioned with the proposed V-cycle multigrid. At each level, we employ a Richardson smoother with one iteration for pre- and post-smoothing and damping factor 0.8. The smoother is further preconditioned with ILU. At the coarse level, a direct solver is employed. In Table 5, we compare our results with the ones in [24]. The multigrid smoother in [24] is a symmetric multiplicative Schwarz smoother [40] and the smoothing procedure is performed only locally. We present the number of CG steps  $n_{10}$  needed to reduce the norm of the residual  $r$  by a factor of  $10^{10}$ , and the average logarithmic convergence rate according to Janssen and Kanschat [24] and Varga [38]:

$$\bar{r} = \frac{1}{n} \log_{10} \frac{|r_0|}{|r_n|},$$

where  $|r_n|$  is the Euclidean norm of the residual vector  $r_n$  at the  $n$ th CG step. We observe that, as the local refinement level increases, the values of both  $n_{10}$  and  $\bar{r}$

TABLE 5

*Number of iterations and average logarithmic convergence rate for the Poisson problem.*

L	Quadrant		Quadrant [24]		Circle		Circle [24]	
	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$
3	6	1.72	4	2.56	6	1.94	5	2.29
4	7	1.55	6	1.69	6	1.67	6	1.83
5	7	1.47	7	1.61	7	1.62	6	1.76
6	7	1.46	7	1.60	7	1.57	7	1.44
7	8	1.41	7	1.60	7	1.53	7	1.61
8	8	1.39	7	1.60	7	1.50	7	1.57
9	8	1.38	7	1.60	7	1.50	7	1.59
10	8	1.37	7	1.60	7	1.52	7	1.59

TABLE 6

*Comparison of different degrees of finite-element families for the Poisson problem.*

L	Quadrant						Circle					
	Bilinear		Quadratic		Biquadratic		Bilinear		Quadratic		Biquadratic	
	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$	$n_{10}$	$\bar{r}$
3	6	1.72	7	1.52	7	1.57	6	1.94	7	1.56	7	1.61
4	7	1.55	7	1.51	7	1.46	6	1.67	7	1.51	7	1.51
5	7	1.47	7	1.48	8	1.41	7	1.62	7	1.50	7	1.48
6	7	1.46	7	1.46	8	1.39	7	1.57	7	1.50	7	1.49
7	8	1.41	7	1.44	8	1.39	7	1.53	7	1.52	7	1.56
8	8	1.39	7	1.43	8	1.39	7	1.50	7	1.55	7	1.57
9	8	1.38	8	1.44	8	1.38	7	1.50	7	1.57	7	1.57
10	8	1.37	8	1.42	8	1.37	7	1.52	7	1.60	7	1.61

converge to fix values and are very close to the results from Janssen and Kanschat [24]. Thanks to our global smoothing approach, better convergence rates may be obtained simply by increasing the number of pre- and post-smoothing iterations. No significant degeneration is observed, showing that our method is suitable for more general applications with respect to the 1-irregular grid constraint.

Table 6 shows more results for various finite-element families: bilinear, quadratic, and biquadratic. For both refinements of the circle and of the first quadrant, the number of iterations and average logarithmic convergence rate are almost independent of the refinement level and of the degree of the finite-element family.

**5. The buoyancy driven flow.** We conclude with the nondimensional buoyancy driven flow problem obtained using the Boussinesq approximation [16, 19],

$$(5.1) \quad \mathbf{u} \cdot \nabla \mathbf{u} - Pr^{1/2} Ra^{-1/2} \Delta \mathbf{u} + \nabla p + \hat{g}T = 0,$$

$$(5.2) \quad \nabla \cdot \mathbf{u} = 0,$$

$$(5.3) \quad \mathbf{u} \cdot \nabla T - Pr^{-1/2} Ra^{-1/2} \Delta T = 0$$

for  $\Omega = [-0.5, 0.5]^3 \subset \mathbb{R}^3$ . The symbols  $\mathbf{u}$ ,  $p$ , and  $T$  denote the velocity, pressure, and temperature fields, respectively. The Prandtl number  $Pr$  describes the ratio between momentum diffusivity and thermal diffusivity, while the Rayleigh number  $Ra$  describes the relation between buoyancy and viscosity within a fluid. The quantity  $\hat{g}$  is the downward unit vector along the negative  $y$  direction, denoted as  $(0, -1, 0)$ . The boundary conditions for the system (5.1)–(5.3) are  $\mathbf{u} = \mathbf{0}$  on  $\partial\Omega$ ,  $T = T_D$  on  $\partial\Omega_D$ , and  $\nabla T \cdot \mathbf{n} = 0$  on  $\partial\Omega_N$ , where  $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$  is the whole boundary. The subsets  $\partial\Omega_D$  and  $\partial\Omega_N$  are portions of  $\partial\Omega$ , where Dirichlet and Neumann boundary conditions are specified for the variable  $T$ , respectively. Given the finite-element spaces  $\mathbf{V}_h \subset \mathbf{H}_0^1(\Omega)$ ,

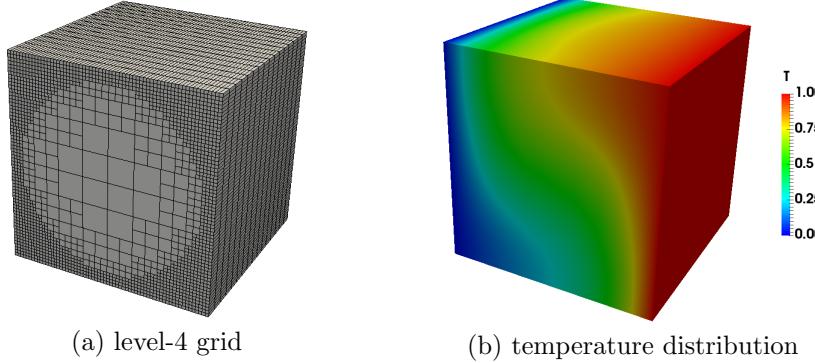


FIG. 16. Three-dimensional buoyancy driven flow with  $Pr = 1$  and  $Ra = 10000$ .

$\Pi_h \subset L_0^2(\Omega)$ , and  $X_h \subset H^1(\Omega)$ , the weak solution  $(\mathbf{u}_h, p_h, T_h) \in \mathbf{V}_h \times \Pi_h \times X_h$  of the buoyancy driven flow problem satisfies

$$(5.4) \quad (\mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) + Pr^{1/2} Ra^{-1/2} (\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) - (p_h, \nabla \cdot \mathbf{v}_h) - (T_h, v_{h,g}) = 0,$$

$$(5.5) \quad (\nabla \cdot \mathbf{u}_h, q_h) = 0,$$

$$(5.6) \quad (\nabla T_h, \nabla r_h) + Pr^{-1/2} Ra^{-1/2} (\mathbf{u}_h \cdot \nabla T_h, r_h) = 0$$

for all  $(\mathbf{v}_h, q_h, r_h) \in \mathbf{V}_h \times \Pi_h \times X_h$ , where  $v_{h,g} = -\mathbf{v}_h \cdot \hat{\mathbf{g}}$ . For the temperature variable,  $T_D$  equals 1 on the right face,  $T_D$  equals 0 on the left face, and Neumann zero boundary conditions are taken on the remaining boundaries; see Figure 16. We consider a continuous triquadratic Lagrangian discretization for  $\mathbf{u}_h$  and  $\mathbf{v}_h$ , a discontinuous piecewise-linear discretization for  $p_h$  and  $q_h \in \Pi_h$ , and a continuous trilinear Lagrangian discretization for  $T_h$  and  $r_h$ . Three different values for the Rayleigh number  $Ra$  are considered, namely,  $Ra = 1000, 5000$ , and  $10000$ , while the Prandtl is constant and equal to 1.

The regular coarse triangulation consists of a  $4 \times 4 \times 1$  hexahedral mesh, uniformly refined once and selectively refined 1, 2, and 3 times. The local refinement procedure follows the strategy that all elements outside the cylinder of radius  $0.5(1 - 0.5^k)$  ( $k = 1, 2, \dots, J - 1$ ) and centered on the z-axis are refined. See Figure 16(a), where the z-axis is parallel to the depth direction.

We use a Newton scheme to linearize the nonlinear equation system (5.4)–(5.6); for more details see [26]. At each Newton step, a GMRES solver preconditioned with our multigrid is applied to solve the linearized system. In the Newton scheme, the stopping tolerance for the  $L_2$ -norm of the distance between two successive solutions is  $1.0 \times 10^{-8}$ . In the GMRES solver, the relative tolerance and absolute tolerance for the scaled preconditioned residual are  $1.0 \times 10^{-8}$  and  $1.0 \times 10^{-15}$ , respectively. For the multigrid V-cycle, once again we use a Richardson smoother with one pre- and post-smoothing iteration and damping factor 0.6. The level solver is further preconditioned with ILU. At the coarse level, a direct solver is employed. This configuration for the solver has been applied to the coupled  $\mathbf{u}, p, T$  system in [25]. Details about the implementation can also be found in [25]. Figure 16(b) shows the temperature distribution at level 4 for the irregular triangulation with  $Pr = 1$  and  $Ra = 10000$ .

The numerical results for the number of Newton iterations, the average number of GMRES iterations per Newton iteration, and the total computational time are listed in Table 7. When uniform refinement is carried out, the total number of degrees

TABLE 7

*Numerical results for both uniform and nonuniform refinement for the buoyancy driven flow.*

L	Ra	1000		5000		10000	
		Uniform	Nonuniform	Uniform	Nonuniform	Uniform	Nonuniform
2	Newton	5	5	6	6	8	8
	GMRES	23.8	24	30.0	30.3	30.4	30.3
	timing	28 s	25.4 s	34.3 s	32.4 s	45.7 s	41.9 s
3	n	0.90	0.91	0.89	0.93	0.91	0.93
	Newton	5	5	6	6	8	8
	GMRES	40.0	28.2	45.2	33.7	41.3	31.9
4	timing	257.6 s	158.2 s	322.2 s	194.8 s	410.2 s	253.9 s
	n	0.93	0.93	0.92	0.92	0.91	0.91
	Newton	5	5	6	6	8	8
4	GMRES	49.4	37.6	56.5	43.5	55	40.8
	timing	2260.2 s	968.6 s	2916.1 s	1214.5 s	3828.0 s	1621.6 s

of freedom is  $N = 34944$ ,  $258044$ , and  $1981428$  at levels 2, 3, and 4, respectively. With nonuniform refinement, we have  $N = 30270$ ,  $161942$ , and  $878726$  at levels 2, 3, and 4, respectively. At each level, the number of Newton iterations for the cases of uniform and nonuniform refinement is the same. The average number of GMRES steps increases for both cases as the level increases. However, for the case of non-uniformly refined meshes, it increases by a smaller factor. Table 7 also shows that the computational time is slightly better than  $O(N)$  as the mesh is refined for both the uniform and nonuniform refinement cases. To estimate the complexity of the algorithm, we used the formula  $\text{Timing}_J = CN_J^n$  for some constant  $C$  independent of  $J$ . Then, considering 2 successive levels  $J - 1$  and  $J$ ,

$$(5.7) \quad n = \frac{\ln(\text{Timing}_J / \text{Timing}_{J-1})}{\ln(N_J / N_{J-1})}.$$

For  $n = 1$  the complexity is linear, for  $n < 1$  it is better than linear, and for  $n > 1$  it is worse than linear. In our table,  $n$  is always less than 1, being consistent with the standard multigrid theory for elliptic problems.

The numerical evidence we reported shows great robustness of our multigrid algorithm.

**6. Conclusions.** We have presented continuous basis functions for the construction of finite-element spaces built on irregular grids arising from a local midpoint refinement procedure. We do not require the grids to be 1-irregular as is done in most existing works in the literature. This makes our results suitable for arbitrary-level hanging node configurations. Our method works with any finite-element geometry, such as quadrilaterals, triangles, tetrahedra, hexahedra, wedges, and mixed grids, and can be applied to all Lagrangian shape functions satisfying the delta property. Our results are comparable to existing results obtained with the deal.II library for the case of 1-irregular grids. In addition, the use of continuous basis functions allows us to perform the smoothing procedure on the entire multigrid space rather than just on a subspace, as is usually done for local refinement strategies. This results in better convergence properties that improve with the number of smoothing iterations.

In conclusion, the multigrid method presented in this work is broader and more versatile compared to existing strategies for h-refinement. It is robust both as a solver and as a preconditioner, and can be applied to linear and nonlinear problems defined on grids with arbitrary-level hanging node configurations.

Future work will consist of extending the proposed method to the case of hp-refinement. Although only 1-irregular meshes are implemented in the deal.II library, the case of hp-refinement is completely addressed in deal.II. The extension of the proposed method to Hermite,  $H(\text{div})$ , and  $H(\text{curl})$  finite elements is also an object of current research.

## REFERENCES

- [1] E. AULISA, S. BNÁ, AND G. BORNIA, *MyFEMuS*, <https://github.com/eaulisa/MyFEMuS> (2017).
- [2] E. AULISA, G. BORNIA, S. CALANDRINI, AND G. CAPODAGLIO, *Convergence estimates for multigrid algorithms with SSC smoothers and applications to overlapping domain decomposition*, Appl. Numer. Math., 131 (2018), pp. 16–38.
- [3] E. AULISA, S. CALANDRINI, AND G. CAPODAGLIO, *An improved multigrid algorithm for n-irregular meshes with subspace correction smoother*, Comput. Math. Appl., 76 (2018), pp. 620–632.
- [4] I. BABUŠKA AND M. R. DORR, *Error estimates for the combined h and p versions of the finite element method*, Numer. Math., 37 (1981), pp. 257–277.
- [5] S. BALAY, S. ABHYANKAR, M. F. ADAMS, J. BROWN, P. BRUNE, K. BUSCHELMAN, L. DALCIN, V. EIJKHOUT, W. D. GROPP, D. KAUSHIK, M. G. KNEPLEY, L. C. MCINNES, K. RUPP, B. F. SMITH, S. ZAMPINI, AND H. ZHANG, *PETSc*, <http://www.mcs.anl.gov/petsc> (2017).
- [6] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II - a general purpose object oriented finite element library*, ACM Trans. Math. Software, 33 (2007), 24.
- [7] W. BANGERTH AND O. KAYSER-HEROLD, *Data structures and requirements for hp finite element software*, ACM Trans. Math. Software, 36 (2009), 4.
- [8] R. E. BANK, A. H. SHERMAN, AND A. WEISER, *Some refinement algorithms and data structures for regular local mesh refinement*, Sci. Comput. Appl. Math. Comput. Phys. Sci., 1 (1983), pp. 3–17.
- [9] J. H. BRAMBLE, *Multigrid Methods*, Pitman Res. Notes Math. 294, CRC Press, Boca Raton, FL, 1993.
- [10] J. H. BRAMBLE AND J. E. PASCIAK, *New convergence estimates for multigrid algorithms*, Math. Comp., 49 (1987), pp. 311–329.
- [11] J. H. BRAMBLE AND J. E. PASCIAK, *New estimates for multilevel algorithms including the V-cycle*, Math. Comp., 60 (1993), pp. 447–471.
- [12] J. H. BRAMBLE, J. E. PASCIAK, J. WANG, AND J. XU, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.
- [13] J. H. BRAMBLE, J. E. PASCIAK, AND J. XU, *The analysis of multigrid algorithms for nonsymmetric and indefinite elliptic problems*, Math. Comp., 51 (1988), pp. 389–414.
- [14] S. BRENNER AND R. SCOTT, *The Mathematical Theory of Finite Element Methods*, Texts Appl. Math. 15, Springer, New York, 1994.
- [15] C. CARSTENSEN AND J. HU, *Hanging nodes in the unifying theory of a posteriori finite element error control*, J. Comput. Math., 27 (2009), pp. 215–236.
- [16] M. A. CHRISTON, P. M. GRESHO, AND S. B. SUTTON, *Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution)*, Internat. J. Numer. Methods Fluids, 40 (2002), pp. 953–980.
- [17] L. DEMKOWICZ, J. T. ODEN, W. RACHOWICZ, AND O. HARDY, *Toward a universal hp adaptive finite element strategy, part 1. Constrained approximation and data structure*, Comput. Methods Appl. Mech. Engrg., 77 (1989), pp. 79–112.
- [18] P. DI STOLFO, A. SCHRÖDER, N. ZANDER, AND S. KOLLMANNSSBERGER, *An easy treatment of hanging nodes in hp-finite elements*, Finite Elem. Anal. Des., 121 (2016), pp. 101–117.
- [19] H. ELMAN, M. MIHAJLOVIĆ, AND D. SILVESTER, *Fast iterative solvers for buoyancy driven flow problems*, J. Comput. Phys., 230 (2011), pp. 3900–3914.
- [20] T.-P. FRIES, A. BYFUT, A. ALIZADA, K. W. CHENG, AND A. SCHRÖDER, *Hanging nodes and XFEM*, Internat. J. Numer. Methods Engrg., 86 (2011), pp. 404–430.
- [21] B. GUO AND I. BABUŠKA, *The hp version of the finite element method*, Comput. Mech., 1 (1986), pp. 21–41.
- [22] A. K. GUPTA, *A finite element for transition from a fine to a coarse grid*, Internat. J. Numer. Methods Engrg., 12 (1978), pp. 35–45.
- [23] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer Ser. Comput. Math. 4, Springer, Berlin, 1985.

- [24] B. JANSSEN AND G. KANSCHAT, *Adaptive multilevel methods with local smoothing for  $H^1$ - and  $H^{\text{curl}}$ -conforming high order finite element methods*, SIAM J. Sci. Comput., 33 (2011), pp. 2095–2114.
- [25] G. KE AND E. AULISA, *New preconditioning techniques for the steady and unsteady buoyancy driven flow problems*, J. Comput. Phys., 371 (2018), pp. 244–260.
- [26] G. KE, E. AULISA, G. BORNIA, AND V. HOWLE, *Block triangular preconditioners for linearization schemes of the Rayleigh–Bénard convection problem*, Numer. Linear Algebra Appl., 24 (2017), e2096.
- [27] P. KUS, P. ŠOLÍN, AND D. ANDRS, *Arbitrary-level hanging nodes for adaptive hp-FEM approximations in 3D*, J. Comput. Appl. Math., 270 (2014), pp. 121–133.
- [28] D. MORTON, J. TYLER, AND J. DORROH, *A new 3D finite element for adaptive h-refinement in 1-irregular meshes*, Internat. J. Numer. Methods Engrg., 38 (1995), pp. 3989–4008.
- [29] M. A. OLSHANSKII AND A. REUSKEN, *Convergence analysis of a multigrid method for a convection-dominated model problem*, SIAM J. Numer. Anal., 42 (2004), pp. 1261–1291.
- [30] C. W. OOSTERLEE AND T. WASHIO, *An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems*, SIAM J. Sci. Comput., 19 (1998), pp. 87–110.
- [31] W. RACHOWICZ AND L. DEMKOWICZ, *An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation*, Comput. Methods Appl. Mech. Engrg., 187 (2000), pp. 307–335.
- [32] W. RACHOWICZ AND L. DEMKOWICZ, *An hp-adaptive finite element method for electromagnetics-part II: A 3D implementation*, Internat. J. Numer. Methods Engrg., 53 (2002), pp. 147–180.
- [33] W. RACHOWICZ, J. T. ODEN, AND L. DEMKOWICZ, *Toward a universal hp adaptive finite element strategy part 3. design of hp meshes*, Comput. Methods Appl. Mech. Engrg., 77 (1989), pp. 181–212.
- [34] A. REUSKEN, *Convergence analysis of a multigrid method for convection-diffusion equations*, Numer. Math., 91 (2002), pp. 323–349.
- [35] A. SCHRÖDER, *Constrained approximation in hp-FEM: Unsymmetric subdivisions and multi-level hanging nodes*, in Spectral and High Order Methods for Partial Differential Equations, J. S. Hesthaven and E. M. Ronquist, eds., Springer, Berlin, 2011, pp. 317–325.
- [36] P. ŠOLÍN, J. ČERVENÝ, AND I. DOLEŽEL, *Arbitrary-level hanging nodes and automatic adaptivity in the hp-FEM*, Math. Comput. Simul., 77 (2008), pp. 117–132.
- [37] P. ŠOLÍN AND L. DEMKOWICZ, *Goal-oriented hp-adaptivity for elliptic problems*, Comp. Methods Appl. Mech. Engrg., 193 (2004), pp. 449–468.
- [38] R. S. VARGA, *Matrix Iterative Analysis*, Springer Ser. Comput. Math. 27, Springer, Berlin, 2000.
- [39] C.-T. WU AND H. C. ELMAN, *Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations*, SIAM J. Sci. Comput., 28 (2006), pp. 2208–2228.
- [40] J. XU, *Iterative methods by space decomposition and subspace correction*, SIAM Rev., 34 (1992), pp. 581–613.
- [41] H. YSERENTANT, *Old and new convergence proofs for multigrid methods*, Acta Numer., 2 (1993), pp. 285–326.
- [42] N. ZANDER, T. BOG, S. KOLLMANNSSBERGER, D. SCHILLINGER, AND E. RANK, *Multi-level hp-adaptivity: High-order mesh adaptivity without the difficulties of constraining hanging nodes*, Comput. Mech., 55 (2015), pp. 499–517.