

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

EUGENIO AULISA[†], GIACOMO CAPODAGLIO[†], AND GUOYI KE[†]

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.

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 [3, 20, 6], 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 refinement strategy. With this choice, special nodes called hanging nodes are introduced [14] 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 [40, 2]. More frequently, modified basis functions for the finite element spaces are considered. As pointed out in [19], 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 [21, 26] for works that, on the contrary, assign degrees of freedom to the hanging nodes.

*Submitted to the editors October 2, 2017.

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

[†]Broadway & Boston, Department of Mathematics and Statistics, Texas Tech University, Lubbock TX 79409, USA (eugenio.aulisa@ttu.edu), (giacomo.capodaglio@ttu.edu), (guoyi.ke@ttu.edu).

In constrained approximation, most of the studies available in the literature require the finite element grids to satisfy the so called 1-irregularity condition [7, 29, 31, 30, 34, 33, 16]. Such a condition requires the maximum difference between the refinement level of two adjacent elements to be one [35]. 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 few recent works avoided the 1-regularity requirement [35, 17, 33, 25]. 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 [35, 6], and it does not require the shape functions to be of tensor product type like [17]. 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 and for a wide variety of finite element types, unlike other methods that only apply to 2D problems or to specific element types in 3D. In the deal.II library [5], 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 [6]. 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 [4]. 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 an arbitrary improvement in the convergence properties that is directly proportional to the number of smoothing iterations. Such a feature cannot be achieved when the smoothing is performed only on a subspace of the multigrid spaces, as it is done in many multigrid strategies for local refinement [10, 11, 23].

The paper is structured as follows. In Section 2, the method for the construction of continuous basis functions is laid out. We introduce the inter-space operators for our multigrid algorithm, and describe the implementation of a solver that makes use of the continuous finite element spaces here defined. In Section 3, we compute the spectral radius of the error operator associated with the multigrid algorithm applied to the bilinear form arising from Poisson equation. A comparison is made with a multigrid algorithm that carries out smoothing only on subspaces [11]. This comparison shows better convergence properties for our algorithm for either 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 2D Poisson problem and the 3D 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 inter-space 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 non-negative 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 Ω 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 Ω . 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 Ω , 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 Ω 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 Θ_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 Θ_k .*

We next define new sequences of subsets of Ω . These 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 Ω at the current level k .

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 presented on $k+1$ different subsets of Ω . 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 Ω_k^l and not on the entire domain Ω 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 it can be seen from Figure 1, where linear elements are used, the hanging nodes are marked with black squares.

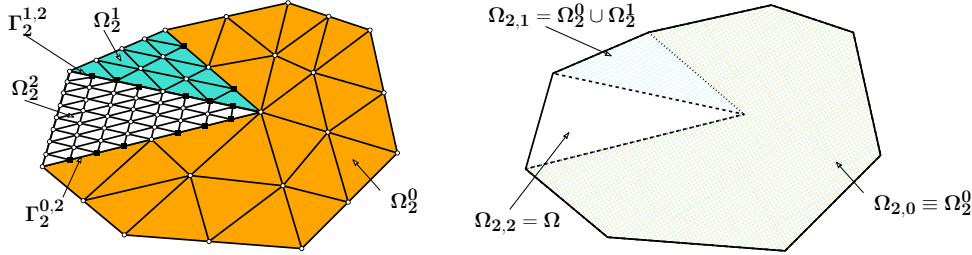


FIGURE 1. An irregular grid with hanging nodes in 2D with $J = 2$. The subdomains involved in the formulation are also highlighted

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

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

where \mathcal{P}_α represents the set of polynomials of degree α . 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 hanging 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 , $X_{k,l} = \bigcup_{m=0}^l X_k^m$ 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 with $X \equiv X_J$ 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 α , a nodal point could be a vertex, an edge mid-point, 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 different X_k^l , since by Definition

2.2, the Ω_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

$$(2) \quad \varphi_{k,i}^l(x_j^l) = \delta_{ij} \text{ for all } x_j^l \in X_k^l.$$

and they are not defined outside Ω_k^l .

- There exist basis functions $\varphi_{k,i}^m$ and $\varphi_{k,i}^n$, defined on different subdomains Ω^m and Ω^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 $\text{supp}(\varphi_{k,i}^n|_{\Gamma_k^{m,n}}) \subset \text{supp}(\varphi_{k,i}^m|_{\Gamma_k^{m,n}})$.
- Let $m < n$, 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 Ω , that in the interior of each subdomain Ω_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

$$(3) \quad \mathcal{E}_k^l \equiv \Omega_{k,l} \setminus \Omega_{k,l-1},$$

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 Ω_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 Ω and

$$(4) \quad \Omega = \bigcup_{l=0}^k \mathcal{E}_k^l.$$

Therefore, $\{\mathcal{E}_k^l\}_{l=0}^k$ represents a disjoint cover of Ω , 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 Ω 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 γ 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 Ω .
- $\varphi_{k,i}$ is continuous if and only if its nodal point is located in the interior of some Ω_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

$$(5) \quad \varphi_{k,i}(x_j) = \delta_{ij} \text{ for all } x_j \in X_k.$$

- $\varphi_{k,i}$ 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 (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 (5), but instead it satisfies the zero property

$$(6) \quad \varphi_{k,i}(x_j) = 0 \text{ for all } x_j \in X_k.$$

- 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

$$(7) \quad \lim_{x \in \mathcal{E}_k^l \rightarrow x_j} \varphi_{k,i}(x) = \delta_{ij}.$$

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

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 $\varphi_{k,j}(x_n) \neq 0$. 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

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

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 $x_j \in X_k$ and assume by contradiction that

$$(8) \quad \varphi_{k,j}(x) = \sum_{i: x_i \in X_k, x_i \neq x_j} c_i \varphi_{k,i}(x), \quad \square$$

where c_i are some real coefficients, not all equal to zero. Recalling Eq. (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

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

With the above lemma we can formally define the space V_k first introduced in (1).

DEFINITION 2.12. *We define V_k to be the set spanned by $\{\varphi_{k,i}\}_{i=1}^{N_k}$, namely $V_k = \text{span}(\{\varphi_{k,i}\}_{i=1}^{N_k})$. 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 $\widehat{\mathbf{V}}_{\mathbf{k}}$. The next step is to build basis functions $\widehat{\varphi}_{k,i}$ on the whole domain Ω 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 α . 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 α , we introduce the sequence $\{x_{i,\beta}^\alpha\}_{\beta=1}^{\beta_{\alpha}^{max}}$: this is the sequence of hanging nodes of degree α 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 α and β . 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*. 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 $\alpha_i^{max} = \max\{\alpha \mid \tilde{\mathcal{H}}_{x_i}^{\alpha, \beta} \neq \emptyset, \text{ with } \beta = 1, \dots, \beta_{\alpha-1}^{max}\}$.

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*

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

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:

$$(9) \quad \hat{\varphi}_k = \begin{bmatrix} \hat{\varphi}_{k, \mathcal{I}_k} \\ \hat{\varphi}_{k, \mathcal{M}_k} \\ \hat{\varphi}_{k, \mathcal{H}_k} \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & \Phi_k \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \varphi_{k, \mathcal{I}_k} \\ \varphi_{k, \mathcal{M}_k} \\ \varphi_{k, \mathcal{H}_k} \end{bmatrix} = \hat{R}_k \varphi_k.$$

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.

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

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.

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, the search extends to neighboring nodes. This situation is embedded in the nature of arbitrary-level hanging nodes, and it cannot be avoided [35, 34]. For such a reason, an exclusive edge-by-edge approach is not possible in 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

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

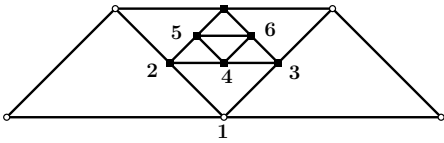


FIGURE 2. Example of a two-dimensional bilinear grid with a two level local refinement.

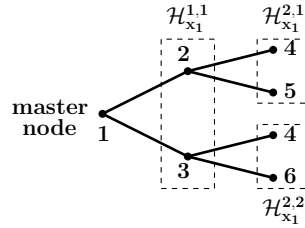


FIGURE 3. Tree diagram that displays the family tree of node x_1 .

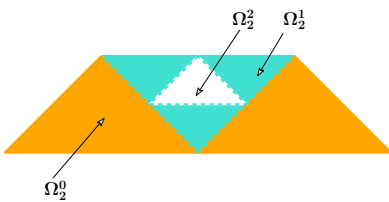


FIGURE 4. Subdomains associated with the different degrees of refinement of the grid in Figure 2.

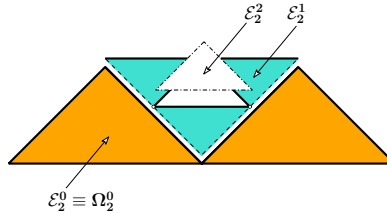


FIGURE 5. Collection $\{E_2^l\}_{l=0}^2$ associated with the grid in Figure 2.

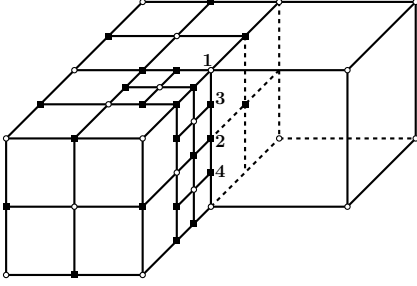


FIGURE 6. Example of three-dimensional bilinear grid on an L-shaped domain with a three level local refinement.

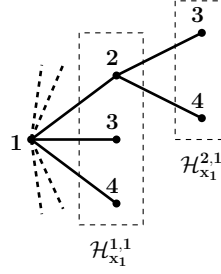


FIGURE 7. Tree diagram that displays the family tree of node x_1 .

$$= \varphi_{k,1} + 0.5 \varphi_{k,2} + 0.5 \varphi_{k,3} + 0.5 \varphi_{k,4} + 0.25 \varphi_{k,5} + 0.25 \varphi_{k,6},$$

$$\hat{\varphi}_{k,2} \equiv 0, \quad \hat{\varphi}_{k,3} \equiv 0, \quad \hat{\varphi}_{k,4} \equiv 0, \quad \hat{\varphi}_{k,5} \equiv 0, \quad \hat{\varphi}_{k,6} \equiv 0.$$

The matrix-vector form corresponding to Eq. (9) is given by:

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

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 $\tilde{\mathcal{H}}_{x_1}^{2,1}$ are different in this case and so it will be clear why we consider the sets $\tilde{\mathcal{H}}_{x_1}^{2,1}$ 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 $\tilde{\mathcal{H}}_{x_1}^{2,1}$ 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.

$$\mathcal{H}_{x_1}^1 = \{x_2, x_3, x_4\}, \quad \mathcal{H}_{x_1}^2 = \{x_3, x_4\},$$

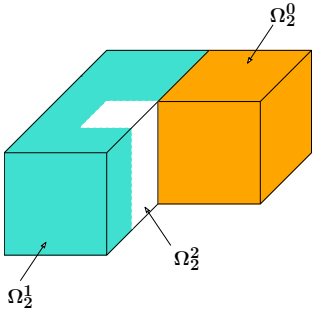


FIGURE 8. Subdomains associated with the different degrees of refinement of the grid in Figure 6.

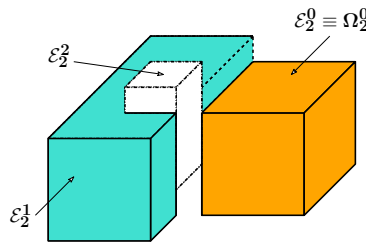


FIGURE 9. Collection $\{\mathcal{E}_2^l\}_{l=0}^2$ associated with the grid in Figure 6.

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

The reason why $\tilde{\mathcal{H}}_{x_1}^{2,1} = \emptyset$ 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

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

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 $\hat{\varphi}_{k,i}$ 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 $\hat{\varphi}_{k,i}$ 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 $\hat{\varphi}_{k,i}$ with the contributions from hanging node functions, $\varphi_{k,j}$, for some j . Therefore, let $x_i \in \mathcal{M}_k$, we will show that $\hat{\varphi}_{k,i}$ 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 η_{max} parent nodes $x_{j_\eta} \in \mathcal{H}_{x_i}^\alpha$ for some α , for which $\varphi_{j_\eta}(x_j) \neq 0$. Considering that x_j can possess multiple degrees as hanging node of x_i , let α_η be the degree of x_j as child of x_{j_η} . Then there will be η_{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 $\hat{\varphi}_{k,i}$ can be rewritten as

$$(10) \quad \hat{\varphi}_{k,i} = \sum_{\eta=1}^{\eta_{max}} \left[\left(\prod_{\zeta=1}^{\alpha_\eta-1} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta,\zeta}) \right) \varphi_{k,j_\eta} + \left(\prod_{\zeta=1}^{\alpha_\eta} \varphi_{k,j_\eta,\zeta-1}(x_{j_\eta,\zeta}) \right) \varphi_{k,j} \right] + f_j,$$

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 $\varphi_{k,j}(x_j) = 0$. Hence, recalling that $x_{j_\eta,\alpha_\eta} \equiv x_j$, and $\varphi_{k,j_\eta,\alpha-1} \equiv \varphi_{k,j_\eta}$, we have

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

Now, let's define the set

$$(11) \quad Z_j = \bigcup_{\gamma} \mathcal{E}_k^\gamma,$$

where each \mathcal{E}_k^γ in the above union is such that x_j belongs to the closure of \mathcal{E}_k^γ .

We observe that \mathcal{E}_k^γ (and so Z_j) is independent of η and that $e_j \subseteq \mathcal{E}_k^\gamma$ for some γ . 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 γ . On every set \mathcal{E}_k^γ , 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 η .

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^γ , for all η , 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^γ , 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^γ .

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 η .

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^γ , for all η . 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 have been removed from the sets $\mathcal{H}_{x_i}^{\alpha,\beta}$ 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^γ .

Hence, let \mathcal{E}_k^γ 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 Eq. (10) this implies that

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

In Case 2, we have for all η ,

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

Again referring to Eq. (10) this gives

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

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

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

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

To complete the proof, we have to show that $\widehat{\varphi}_{k,i}$ 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 edge or face that contains p nodes x_{j_ℓ} , 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:

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

where \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 3D 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 the time.

Let us define the following function

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

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 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 (5).

If $x_i \in \mathcal{M}_k$ then, according to (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 (6). Hence, letting $x_j \notin \mathcal{H}_k$, we have

$$(12) \quad \widehat{\varphi}_{k,\mathcal{M}}(x_j) = \varphi_{k,\mathcal{M}}(x_j) + \Phi_k \varphi_{k,\mathcal{H}}(x_j) = \delta_{ij} \mathbf{1} + \mathbf{0}.$$

This means that

$$(13) \quad \widehat{\varphi}_{k,i}(x_j) = \delta_{ij}, \text{ for all } x_j \in \mathcal{I}_k \cup \mathcal{M}_k.$$

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 (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 $\mathbf{a} = [\mathbf{a}_{\mathcal{I}}, \mathbf{a}_{\mathcal{M}}, \mathbf{a}_{\mathcal{H}}]^\top$ and $\widehat{\mathbf{b}} = [\widehat{\mathbf{b}}_{\mathcal{I}}, \widehat{\mathbf{b}}_{\mathcal{M}}, \widehat{\mathbf{b}}_{\mathcal{H}}]^\top$ be the coefficient representation vectors of a and \widehat{b} , i.e.

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

Then $a = \widehat{b}$ iff $\mathbf{a}_{\mathcal{I}} = \widehat{\mathbf{b}}_{\mathcal{I}}$, $\mathbf{a}_{\mathcal{M}} = \widehat{\mathbf{b}}_{\mathcal{M}}$, and $\mathbf{a}_{\mathcal{H}} = \Phi_k^\top \mathbf{a}_{\mathcal{M}}$.

Proof. The proof of the proposition follows from Eq. (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 $\mathbf{a} = [\mathbf{a}_{\mathcal{I}}, \mathbf{a}_{\mathcal{M}}, \Phi_k^\top \mathbf{a}_{\mathcal{M}}]^\top$, then a is also in \widehat{V}_k and we can choose the same vector \mathbf{a} as representation of a in \widehat{V}_k .

2.5. The inter-space 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

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

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

$$\widehat{v} = \widehat{\varphi}_k^\top \widehat{\mathbf{v}},$$

for some coefficient vector $\widehat{\mathbf{v}} = [\widehat{v}_1, \widehat{v}_2, \dots, \widehat{v}_{N_k}]^\top \in \mathbb{R}^{N_k}$. Then, for the matrix representation of $\widehat{\mathcal{P}}_k$ we have $\widehat{P}_k = \widehat{R}_k^\top$, where \widehat{R}_k is the matrix from Eq. (9).

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

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

Thus, the matrix representation of $\widehat{\mathcal{P}}_k$ is given by $\widehat{P}_k = \widehat{R}_k^\top$. \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 Eq. (9).

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

$$\widehat{\mathcal{P}}_k \widehat{\varphi}_{k,i} = \widehat{\varphi}_{k,i} = \widehat{R}_{k,i} \varphi_k,$$

where $\widehat{R}_{k,i}$ corresponds to the i -th row of the matrix \widehat{R}_k defined in (9). Consequently, for any v in V_k

$$(14) \quad \langle \widehat{\mathcal{R}}_k v, \widehat{\varphi}_{k,i} \rangle = \langle v, \widehat{\mathcal{P}}_k \widehat{\varphi}_{k,i} \rangle = \langle v, \widehat{R}_{k,i} \varphi_k \rangle = \sum_{j=1}^{N_k} \widehat{R}_{k,ij} \langle v, \varphi_{k,j} \rangle.$$

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

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

for all $i = 1, \dots, N_k$. Then Eq. (14) can be rewritten as

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

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

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

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

$$A_{k,ij} = a(\varphi_{k,i}, \varphi_{k,j}), \text{ for all } i, j = 1, \dots, N_k,$$

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

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

Proof. For all i, j

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

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 \widehat{A}_k corresponding to the hanging nodes are set equal to one.

Note that from Definition 2.8 and 2.14 it follows that

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

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 $\widehat{\mathcal{Q}}_k : \widehat{V}_{k-1} \rightarrow \widehat{V}_k$ is

the natural injection between \widehat{V}_{k-1} and \widehat{V}_k . The respective matrix representations are denoted by Q_k and \widehat{Q}_k .

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

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

$$a = Q_k(\widehat{P}_{k-1}(\widehat{v}_{k-1})) = \widehat{v}_{k-1}, \text{ for all } \widehat{v}_{k-1} \in \widehat{V}_{k-1},$$

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

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

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

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

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

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

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

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

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

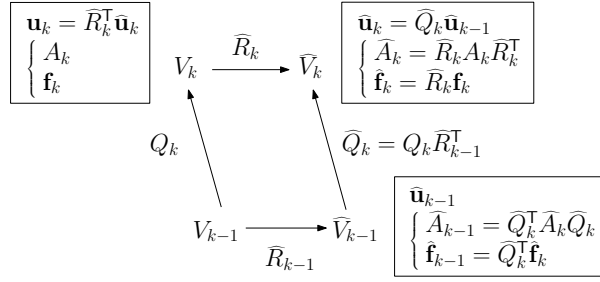
Then, $\widehat{A}_{k-1} = \widehat{Q}_k^T \widehat{A}_k \widehat{Q}_k$, namely, with a slight abuse of terminology, \widehat{A}_{k-1} is the restriction of \widehat{A}_k to the space \widehat{V}_{k-1} .

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

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

Schematics of all the inter-space 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 [4]. The restriction operator \widehat{R}_k from V_k to \widehat{V}_k is pre-computed 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 pre-computed 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 pre-computed as well.

2.6. The numerical algorithm. Once the inter-space operators are available in terms of matrices, the numerical solution of the problem $a(u, v) = \langle f, v \rangle$, is sought

FIGURE 10. *Inter-space operators*

in the continuous space \hat{V}_J . Namely, we seek the solution of the discretized problem $\hat{A}_J \hat{\mathbf{u}}_J = \hat{\mathbf{f}}_J$. We do not work directly in the continuous space \hat{V}_J . We rather work in the discontinuous space V_J and use the restriction operator \hat{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 $\hat{\mathbf{u}}_J^0$ be the initial guess, then

1. assemble the matrix A_J in V_J using the test functions φ_J ;
2. restrict A_J in the \hat{V}_J space

$$\hat{A}_J = \hat{R}_J A_J \hat{R}_J^T;$$

3. prolong the current solution in the V_J space

$$\mathbf{u}_J^i = \hat{R}_J^T \hat{\mathbf{u}}_J^i;$$

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 \hat{V}_J

$$\hat{\mathbf{r}}_J^i = \hat{R}_J \mathbf{r}_J^i;$$

6. Let \hat{D}_J be an “easy to invert” approximation of the matrix \hat{A}_J , then solve

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

7. if $\|\hat{\mathbf{w}}_J^i\| \leq \varepsilon$ 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 element-wise 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 element-wise we use the standard test functions $\varphi_{J,i}^l$, which is a very convenient approach since no new bases have to be really constructed.

The operator \hat{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 (MG) 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 C++ library

PETSc [4] as a black box, providing as inputs the matrices \hat{A}_k for $k = J, \dots, 0$, the matrices \hat{Q}_k for $k = J, \dots, 1$ and the residual vector $\hat{\mathbf{r}}_J$, and letting PETSc compute $\hat{\mathbf{w}}_J^i$. 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 \hat{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 \hat{D}_J is an “easy to invert” approximation of the tangent matrix

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

This corresponds to a Newton-Raphson iterative scheme [24]. 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 (SPD) [22, 8, 39, 9, 38, 11]. Other types of boundary value problems have also been addressed, where A is nonsymmetric or indefinite [27, 32, 37, 12].

In [11], 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 be obtained also when smoothing is performed only on a subspace of the multigrid space at every level. This allows applying the theory proposed in [11] 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 [13] was obtained, in the sense that the error bound δ_J for the multigrid error E_J , depended on the number of multigrid levels J ,

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

where $a(\cdot, \cdot)$ is the energy norm and C is a constant independent of J . It is clear when looking at δ_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 [11] as BPWX algorithm after the authors: Bramble, Pasciak, Wang and Xu.

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.

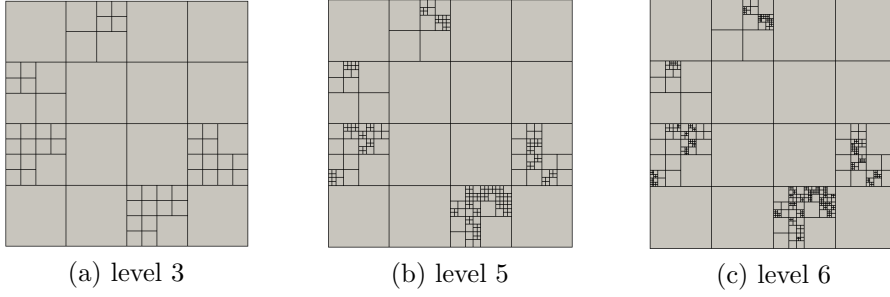


FIGURE 11. Refinement of the quadrilateral element mesh: 50% of the elements from the previous level are randomly selected for refinement.

TABLE 1
Spectral radius considering the meshes depicted in Figure 11 and Figure 12. One pre and post-smoothing iteration is considered.

Methods	Proposed Method		BPWX Method	
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

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 [28].

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

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

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 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

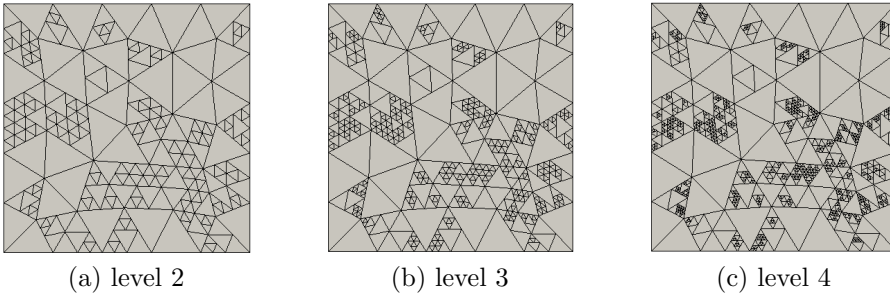


FIGURE 12. Refinement of the triangular element mesh: 50% of the elements from the previous level are randomly selected for refinement.

factor 0.6, preconditioned with ILU. Tests are made considering the bilinear operator $a(u, v) = \langle \nabla u, \nabla v \rangle$. We use biquadratic and triquadratic Lagrangian test functions in 2D and 3D, 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 2×2 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 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 $I - \hat{D}_J^{-1} \hat{A}_J$ for the six cases in Figures 11 and 12, when only one pre and post-smoothing iterations are 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 level c, 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

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 Figure 11 and Figure 12.

Methods	Proposed Method		BPWX Method	
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

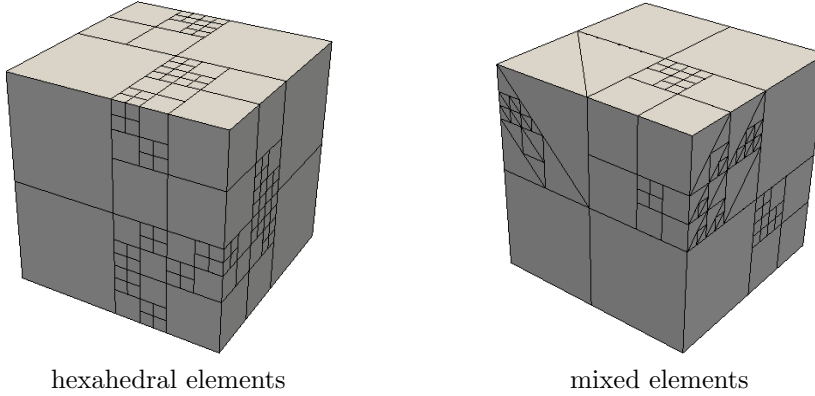


FIGURE 13. Refinement of different types of elements starting with one uniform level and refining up to three nonuniform levels: 50% of the elements from the previous level are randomly selected for refinement.

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.

The results for the spectral radius with one pre and post-smoothing iteration are 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 solver to converge would decrease significantly with the proposed method, while it would stay the same for the BPWX method. Once again the hexahedra perform the best, followed by the wedges, the tetrahedra and the mixed ones.

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

TABLE 5

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

L	Quadrant		Quadrant [23]		Circle		Circle [23]	
	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

4. The Poisson problem. We continue our investigation by considering the weak formulation of the Poisson problem discussed in [23]. Namely find $u_h \in V_h \subset H_0^1(\Omega)$ such that

$$(15) \quad (\nabla u_h, \nabla v_h) = (f, v_h), \text{ for all } v_h \in V_h,$$

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 for irregular triangulations proposed in [23] only applies to 1-irregular grids and has been implemented only on quadrilateral and hexahedral element. Our method has no such limitations.

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 $J = 8$.

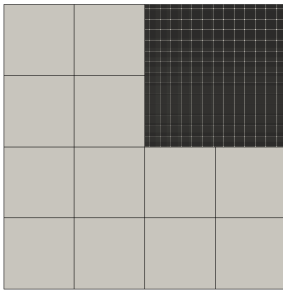


FIGURE 14. Refinement of the first quadrant after 7 refinements.

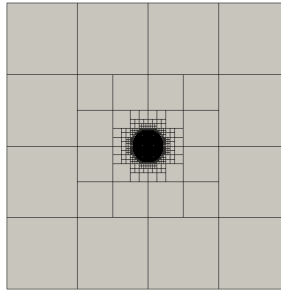


FIGURE 15. Refinement of the circle after 8 refinements.

Continuous piecewise-bilinear approximation is considered for both u_h and v_h . The linear system arising from the weak formulation in Eq.(15) defined on the above irregular triangulations is solved using a Conjugate Gradient (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. In Table 5, we compare our results with the ones in [23]. The multigrid smoother in [23] is a symmetric multiplicative Schwarz smoother [38] 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 [23] and Varga [36]:

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

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} saturate and are very close to the results from Janssen [23]. 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 non-dimensional buoyancy driven flow problem obtained using the Boussinesq approximation [15, 18],

$$(16) \quad \mathbf{u} \cdot \nabla \mathbf{u} - Pr^{1/2} Ra^{-1/2} \Delta \mathbf{u} + \nabla p + \hat{g} T = 0,$$

$$(17) \quad \nabla \cdot \mathbf{u} = 0,$$

$$(18) \quad \mathbf{u} \cdot \nabla T - Pr^{-1/2} Ra^{-1/2} \Delta T = 0,$$

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 gravity direction. The boundary conditions for the system (16) – (18) are $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$, $T = T_D$ on $\partial\Omega_D$, and $\nabla T \cdot \mathbf{n} = 0$ on

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

$\partial\Omega_N$, 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 $\mathbf{V}_h \subset \mathbf{H}_0^1(\Omega)$, $\Pi_h \subset L_0^2(\Omega)$ 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

$$(19) \quad (\mathbf{u}_h \cdot \nabla \mathbf{u}_h, \mathbf{v}_h) + Pr^{1/2} Ra^{-1/2} (\nabla \mathbf{u}_h, \nabla \mathbf{v}_h) - (p_h, \nabla \cdot \mathbf{v}_h) - (T_h, v_{h,g}) = 0,$$

$$(20) \quad (\nabla \cdot \mathbf{u}_h, q_h) = 0,$$

$$(21) \quad (\nabla T_h, \nabla r_h) + Pr^{-1/2} Ra^{-1/2} (\mathbf{u}_h \cdot \nabla T_h, r_h) = 0,$$

for all $(\mathbf{v}_h, q_h, r_h) \in \mathbf{V}_h \times \Pi_h \times X_h$, where $v_{h,g} = -\mathbf{v}_h \cdot \hat{g}$. 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. 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 (19) - (21), for more details see [24]. 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 1.0×10^{-8} . In the GMRES solver, the relative tolerance and absolute tolerance for the scaled preconditioned residual are 1.0×10^{-8} and 1.0×10^{-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. 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 of freedom is $N = 34944, 258044$ and 1981428 at levels 2, 3 and 4, respectively.

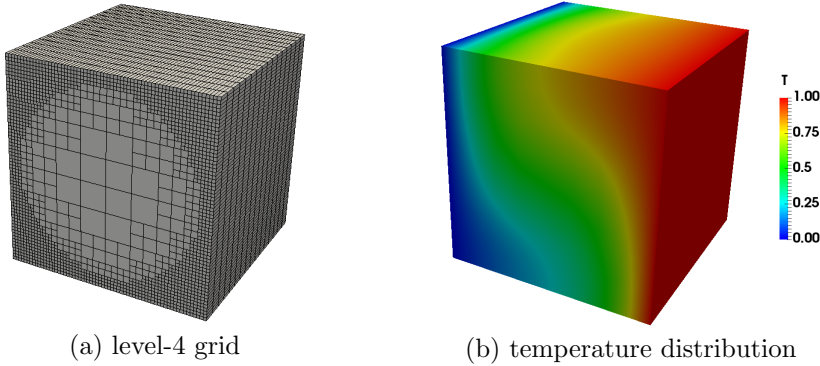


FIGURE 16. Three-dimensional buoyancy driven flow with $Pr = 1$ and $Ra = 10000$.

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	28s	25.4s	34.3s	32.4s	45.7s	41.9s
	n	0.90	0.91	0.89	0.93	0.91	0.93
3	Newton	5	5	6	6	8	8
	GMRES	40.0	28.2	45.2	33.7	41.3	31.9
	Timing	257.6s	158.2s	322.2s	194.8s	410.2s	253.9s
	n	0.93	0.93	0.92	0.92	0.91	0.91
4	Newton	5	5	6	6	8	8
	GMRES	49.4	37.6	56.5	43.5	55	40.8
	Timing	2260.2s	968.6s	2916.1s	1214.5s	3828.0s	1621.6s
	n						

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

$$(22) \quad n = \frac{\ln(\text{Timing}_J / \text{Timing}_{J-1})}{\ln(N_J / N_{J-1})}.$$

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 it 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 it 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 object of current research.

REFERENCES

- [1] Eugenio Aulisa, Simone Bn, and Giorgio Bornia. FEMuS Web page. <https://github.com/eaaulisa/MyFEMuS>, 2017.
- [2] Eugenio Aulisa, Giorgio Bornia, Sara Calandrini, and Giacomo Capodaglio. Convergence estimates for multigrid algorithms with ssc smoothers and applications to overlapping domain decomposition. *submitted: arXiv:1709.09628*, 2017.
- [3] Ivo Babuška and Milo R Dorr. Error estimates for the combined h and p versions of the finite element method. *Numerische Mathematik*, 37(2):257–277, 1981.
- [4] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, and Hong Zhang. PETSc Web page. <http://www.mcs.anl.gov/petsc>, 2017.
- [5] Wolfgang Bangerth, Ralf Hartmann, and Guido Kanschat. deal. iia general-purpose object-oriented finite element library. *ACM Transactions on Mathematical Software (TOMS)*, 33(4):24, 2007.
- [6] Wolfgang Bangerth and O Kayser-Herold. Data structures and requirements for hp finite element software. *ACM Transactions on Mathematical Software (TOMS)*, 36(1):4, 2009.
- [7] Randolph E Bank, Andrew H Sherman, and Alan Weiser. Some refinement algorithms and data structures for regular local mesh refinement. *Scientific Computing, Applications of Mathematics and Computing to the Physical Sciences*, 1:3–17, 1983.
- [8] James H Bramble. *Multigrid methods*, volume 294. CRC Press, 1993.
- [9] James H Bramble and Joseph E Pasciak. New convergence estimates for multigrid algorithms. *Mathematics of computation*, 49(180):311–329, 1987.
- [10] James H Bramble and Joseph E Pasciak. New estimates for multilevel algorithms including the V-cycle. *Mathematics of computation*, 60(202):447–471, 1993.
- [11] James H Bramble, Joseph E Pasciak, Jun Ping Wang, and Jinchao Xu. Convergence estimates for multigrid algorithms without regularity assumptions. *Mathematics of Computation*, 57(195):23–45, 1991.
- [12] James H Bramble, Joseph E Pasciak, and Jinchao Xu. The analysis of multigrid algorithms for nonsymmetric and indefinite elliptic problems. *Mathematics of Computation*, 51(184):389–414, 1988.
- [13] Susanne Brenner and Ridgway Scott. *The mathematical theory of finite element methods*, volume 15. Springer Science & Business Media, 2007.
- [14] Carsten Carstensen and Jun Hu. Hanging nodes in the unifying theory of a posteriori finite element error control. *Journal of Computational Mathematics*, pages 215–236, 2009.
- [15] Mark A Christon, Philip M Gresho, and Steven B Sutton. Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). *International Journal for Numerical Methods in Fluids*, 40(8):953–980, 2002.
- [16] Leszek Demkowicz, J Tinsely Oden, Waldemar Rachowicz, and O Hardy. Toward a universal hp adaptive finite element strategy, part 1. constrained approximation and data structure. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):79–112, 1989.
- [17] Paolo Di Stolfo, Andreas Schröder, Nils Zander, and Stefan Kollmannsberger. An easy treatment of hanging nodes in hp-finite elements. *Finite Elements in Analysis and Design*, 121:101–117, 2016.
- [18] Howard Elman, Milan Mihajlović, and David Silvester. Fast iterative solvers for buoyancy driven flow problems. *Journal of Computational Physics*, 230(10):3900–3914, 2011.
- [19] Thomas-Peter Fries, Andreas Byfut, Alaskar Alizada, Kwok Wah Cheng, and Andreas Schröder. Hanging nodes and xfem. *International Journal for Numerical Methods in Engineering*, 86(4-5):404–430, 2011.
- [20] B Guo and I Babuška. The hp version of the finite element method. *Computational Mechanics*, 1(1):21–41, 1986.
- [21] Ajaya Kumar Gupta. A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering*, 12(1):35–45, 1978.
- [22] Wolfgang Hackbusch. *Multi-grid methods and applications*, volume 4. Springer Science & Business Media, 2013.

- [23] Bärbel Janssen and Guido Kanschat. Adaptive multilevel methods with local smoothing for H^1 - and H^{curl} -conforming high order finite element methods. *SIAM Journal on Scientific Computing*, 33(4):2095–2114, 2011.
- [24] Guoyi Ke, Eugenio Aulisa, Giorgio Bornia, and Victoria Howle. Block triangular preconditioners for linearization schemes of the Rayleigh–Bénard convection problem. *Numerical Linear Algebra with Applications*, 2017.
- [25] Pavel Kus, Pavel Solin, and David Andrs. Arbitrary-level hanging nodes for adaptive hp-fem approximations in 3d. *Journal of Computational and Applied Mathematics*, 270:121–133, 2014.
- [26] DJ Morton, JM Tyler, and JR Dorroh. A new 3d finite element for adaptive h-refinement in 1-irregular meshes. *International journal for numerical methods in engineering*, 38(23):3989–4008, 1995.
- [27] Maxim A Olshanskii and Arnold Reusken. Convergence analysis of a multigrid method for a convection-dominated model problem. *SIAM journal on numerical analysis*, 42(3):1261–1291, 2004.
- [28] Cornelis W Oosterlee and Takumi Washio. An evaluation of parallel multigrid as a solver and a preconditioner for singularly perturbed problems. *SIAM Journal on Scientific Computing*, 19(1):87–110, 1998.
- [29] W Rachowicz, J Tinsley Oden, and L Demkowicz. Toward a universal hp adaptive finite element strategy part 3. design of hp meshes. *Computer Methods in Applied Mechanics and Engineering*, 77(1-2):181–212, 1989.
- [30] Waldemar Rachowicz and L Demkowicz. An hp-adaptive finite element method for electromagnetics part ii: A 3d implementation. *International journal for numerical methods in engineering*, 53(1):147–180, 2002.
- [31] Waldemar Rachowicz and Leszek Demkowicz. An hp-adaptive finite element method for electromagnetics: Part 1: Data structure and constrained approximation. *Computer methods in applied mechanics and engineering*, 187(1):307–335, 2000.
- [32] Arnold Reusken. Convergence analysis of a multigrid method for convection–diffusion equations. *Numerische Mathematik*, 91(2):323–349, 2002.
- [33] Andreas Schröder. Constrained approximation in hp-fem: Unsymmetric subdivisions and multi-level hanging nodes. *Spectral and High Order Methods for Partial Differential Equations (JS Hesthaven and EM Ronquist, eds.)*, (76):317–325, 2011.
- [34] P Šolín and L Demkowicz. Goal-oriented hp-adaptivity for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 193(6):449–468, 2004.
- [35] Pavel Šolín, Jakub Červený, and Ivo Doležel. Arbitrary-level hanging nodes and automatic adaptivity in the hp-fem. *Mathematics and Computers in Simulation*, 77(1):117–132, 2008.
- [36] Richard S Varga. *Matrix iterative analysis*, volume 27. Springer Science & Business Media, 2009.
- [37] Chin-Tien Wu and Howard C Elman. Analysis and comparison of geometric and algebraic multigrid for convection-diffusion equations. *SIAM Journal on Scientific Computing*, 28(6):2208–2228, 2006.
- [38] Jinchao Xu. Iterative methods by space decomposition and subspace correction. *SIAM review*, 34(4):581–613, 1992.
- [39] Harry Yserentant. Old and new convergence proofs for multigrid methods. *Acta numerica*, 2:285–326, 1993.
- [40] Nils Zander, Tino Bog, Stefan Kollmannsberger, Dominik Schillinger, and Ernst Rank. Multi-level hp-adaptivity: high-order mesh adaptivity without the difficulties of constraining hanging nodes. *Computational Mechanics*, 55(3):499–517, 2015.