



# The 3D Nonconforming Virtual Element Method with enhanced spaces: Numerical investigation and implementation guide

Franco Dassi<sup>1</sup> · Manuel Trezzi<sup>1</sup>

Received: 14 July 2025 / Accepted: 24 November 2025  
© The Author(s) 2025

## Abstract

In this paper, we present the implementation details of the three-dimensional Nonconforming Virtual Element Method, building upon the theoretical framework introduced in (*ESAIM: Math. Model. Numer. Anal.*, 50(3), 879–904, (2016), (*IMA J. Numer. Anal.*, 37, (2015)). In this formulation, the local spaces are suitably enhanced to allow the construction of the  $L^2$  projection of virtual functions onto polynomials of degree  $k$ . This higher-order projection provides a more accurate representation of the load term and, more importantly, enables the definition of the reaction term. We perform a numerical convergence study of the  $L^2$ -norm and  $H^1$ -seminorm errors with respect to both mesh size and polynomial degree in the three-dimensional setting. Numerical results are then compared with those obtained using the conforming VEM, showing the consistency and computational performance of the proposed 3D implementation.

**Keywords** Nonconforming virtual element method · Diffusion reaction problems · Polyhedral meshes · Implementation details

**Mathematics Subject Classification** 65N30 · 35J15

---

Franco Dassi and Manuel Trezzi contributed equally to this work.

---

✉ Manuel Trezzi  
manuel.trezzi@unimib.it

Franco Dassi  
franco.dassi@unimib.it

<sup>1</sup> Dipartimento di Matematica e Applicazioni, Università degli Studi di Milano Bicocca, Via Cozzi 55, Milan 20125, Italy

## 1 Introduction

The Virtual Element Method (VEM) was introduced in 2013 by L. Beirão da Veiga et al. [3] as a generalization of the Finite Element Method. It allows for very general polygonal and polyhedral meshes, including non-convex shapes, and can handle regular solutions.

The key feature of the VEM lies in the definition of the basis functions. Unlike the FEM, where basis functions are *only* polynomials of a prescribed degree, the VEM defines basis functions as the solution of a specific Partial Differential Equation (PDE) that includes polynomials but also other functions. These basis functions are never computed explicitly, and in practice, there is no need to know them point-wise during the matrix and right hand side assembling processes. Furthermore, since polynomials are part of the VE space, the accuracy of the method and the compatibility with the FEM are guaranteed.

This approach brings some advantages for both mesh discretization and the definition of functional spaces. On one side, having the possibility to deal with generally shaped polygons/polyhedrons provides more flexibility in mesh generation. Furthermore, if we consider mesh adaptation processes, the VEM allows the presence of hanging nodes, which facilitate the refinement process. On the other side, the VE spaces can be built in such a way that the discrete functions inherit some useful properties. For instance, in [4], the authors developed a family of VE spaces for Stokes problems exactly “divergence-free”. FEM approaches obtain such a property in a relaxed/weak sense, or using special discretizations, for instance the Scott-Vogelius type elements. Similar consideration can be made for [5] where a VEM approach was developed to solve elasticity equations based on the Hellinger–Reissner variational principle. The FEM imposes the symmetry of the stress tensor variable in a weak sense, and this constraint increases the dimension of the linear system. The VEM discretization proposed in [5] overcomes this issue by imposing the symmetry of the tensor field within the discrete functional spaces, resulting in a cheaper linear system, as no additional equations are needed to enforce the symmetry.

One of the main advantages of the nonconforming VEM over its conforming counterpart lies in its definition and implementation in the 3D case. The conforming method requires the prior definition of a VE space on each polygonal face of a polyhedron, followed by the definition of the VE space in the interior. In the nonconforming case, there is no need to define a VE space on the faces, making the overall definition more natural. When moving from 2D to 3D, the degrees of freedom (DoFs) undergo a dimensional shift (edges become faces, and faces become volumes), but their definition remains essentially the same, preserving the structure of the 2D setting.

B. Ayuso de Dios et al. introduce the nonconforming version of the VEM for second order elliptic problems [1]. They provide the construction of both two and three dimensional spaces and they make the error analysis, but they limit themselves to test these nonconforming virtual elements *only* to two dimensions. Although the work [2] by Cangiani et al. offers a valuable guide for the definition of local matrices in both two and three dimensions. To the best of our knowledge, the only available numerical results for the 3D nonconforming VEM are those presented in [6]. Here, the authors focus on the Poisson problem and develop an analysis that accommodates polyhedra

with small faces. Then, the nonconforming VEM has been applied to several problems [2, 7–11]. However, all these papers provide the theory for both 2D and 3D setting, but they provide numerical experiments *only* in 2D.

The aim of this paper is twofold: first, to numerically validate the theory established in [1, 2] in the three-dimensional setting; second, to provide further implementation details for the practical realization of the nonconforming VEM in 3D.

The paper is organized as follows. In Section 2, we present the model problem. Then, in Section 3, we describe the VE spaces/projections and show how to build the linear and bilinear form for a diffusion-reaction problem. In this section, we also outline key implementation aspects to aid and inform future developments. Finally, in Section 4, we show some numerical 3D experiments.

**Notation** *During the paper, we adopt standard notation for Sobolev spaces and norms [12].*

*We denote by  $E$  a polyhedron with  $\bar{x}_E$ ,  $h_E$ , and  $|E|$  representing its barycentre, diameter, and volume, respectively. Similarly, we denote a generic face by  $f$ , with  $\mathbf{x}_f$ ,  $h_f$ , and  $|f|$  its barycentre, diameter and area. Then,  $\Sigma_h^E$  is the set of the faces of the element  $E$  and  $\mathbf{n}^E$  is the outward unit normal vector to the polyhedron  $E$ , similarly  $\mathbf{n}^f$  is one of the two outward normal vectors with respect to a face  $f$ .*

*We define the jump operator for a function  $v$  on a face shared by two polyhedrons as*

$$[[v]]_f = v|_{E^+} - v|_{E^-} ,$$

*where  $E^+$  and  $E^-$  are such that  $f \in \overline{E^+} \cap \overline{E^-}$  while, if  $f$  is shared by only one polyhedron, the jump is defined as  $[[v]]_f = v|_E$ .*

*Then, given  $\mathcal{O} \subset \mathbb{R}^d$  and a positive integer  $k$ , we denote by  $\mathbb{P}_k(\mathcal{O})$  the space of polynomials in  $d$  variables of degree less than or equal to  $k$ . Following common practice in the VEM, we choose as a basis of  $\mathbb{P}_k(\mathcal{O})$  the set of scaled monomials:*

$$\mathcal{M}^s(\mathcal{O}) := \left\{ \left( \frac{\mathbf{x} - \mathbf{x}_{\mathcal{O}}}{h_{\mathcal{O}}} \right)^{\boldsymbol{\ell}} : |\boldsymbol{\ell}| \leq s \right\} ,$$

*where  $\boldsymbol{\ell}$  is a multi-index whose length depends on  $d$ . It is worth noting that when  $\mathcal{O}$  is a polyhedron, the variable  $\mathbf{x}$  refers to the physical coordinates of a point in 3D space, whereas when  $\mathcal{O}$  is a polygon (specifically, a face of a polyhedron),  $\mathbf{x}$  denotes the local coordinates on the plane defined by the polygon.*

## 2 The model problem

Let  $\Omega \subset \mathbb{R}^3$  be a polytopal domain with boundary  $\Gamma$ , we consider the following general diffusion-reaction problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\operatorname{div}(\boldsymbol{\kappa}\nabla u) + u = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma, \end{cases} \tag{1}$$

where  $f \in L^2(\Omega)$  is the load term, and  $\boldsymbol{\kappa} \in [L^\infty(\Omega)]^{3 \times 3}$  is the symmetric and elliptic diffusive tensor. We remark that if  $\boldsymbol{\kappa}$  coincides with the identity matrix, the term  $\operatorname{div}(\boldsymbol{\kappa}\nabla u)$  coincides with the Laplacian operator  $\Delta u$ .

**Remark 1** In the model problem, we assume that the reaction term  $u$  is not multiplied by a reaction coefficient. This is justified by the fact that the equation can be rescaled by dividing through by the reaction coefficient, thereby simplifying the analysis without loss of generality.

The variational formulation of the model problem reads as:

$$\begin{cases} \text{find } u \in H_0^1(\Omega) \text{ such that:} \\ a(u, v) + m(u, v) = F(v) \quad \forall v \in H_0^1(\Omega), \end{cases} \tag{2}$$

where the linear and bilinear forms are defined as

$$a(u, v) := \int_{\Omega} \boldsymbol{\kappa} \nabla u \cdot \nabla v \, d\Omega \quad \text{for all } u, v \in H_0^1(\Omega), \tag{3}$$

$$m(u, v) := \int_{\Omega} u v \, d\Omega \quad \text{for all } u, v \in H_0^1(\Omega), \tag{4}$$

$$F(v) := \int_{\Omega} f v \, d\Omega \quad \text{for all } v \in H_0^1(\Omega). \tag{5}$$

Problem (2) is well-posed as a consequence of the ellipticity of  $\boldsymbol{\kappa}$ , the coercivity of the bilinear forms  $a(\cdot, \cdot)$  and  $m(\cdot, \cdot)$ , and the application of the Lax–Milgram lemma.

### 3 The VEM nonconforming discretization

After describing the mesh assumption, we introduce the nonconforming virtual element spaces and provide implementation details for constructing both the projection operators and the linear and bilinear forms appearing in (2).

#### 3.1 Mesh assumptions

Let  $\{\Omega_h\}_h$  be a family of tessellations of the domain  $\Omega$  into non-overlapping polyhedra. The set of all the faces in the decomposition  $\Omega_h$  is denoted by  $\Sigma_h$ . We assume that each mesh  $\Omega_h$  satisfies the following properties: there exists a constant  $\rho$ , independent of  $h$ , such that

- each polyhedron  $E$  and each face  $f$  is star-shaped with respect to a ball of radius greater than or equal to  $\rho h_E$  and  $\rho h_f$ , respectively;
- $h_f \geq \rho h_E$  holds for every face  $f \in \Sigma_h^E$ .

It is worth noting that such mesh assumptions are mainly employed for the theoretical analysis of the VEM. In practice, however, numerical experiments have shown that the method is more robust and exhibits the expected behaviour even when these assumptions are not satisfied.

### 3.2 The nonconforming VEM spaces

Let  $E \in \Omega_h$  be a polyhedron, and let  $k$  be a positive integer. We introduce the enhanced nonconforming virtual element space defined as

$$V_h^k(E) := \left\{ v_h \in H^1(E) : \begin{aligned} &\frac{\partial v_h}{\partial \mathbf{n}} \in \mathbb{P}_{k-1}(f) \quad \forall f \in \Sigma_h^E, \\ &\Delta v_h \in \mathbb{P}_k(E), \\ &\left( v_h - \Pi_k^{\nabla, E} v_h, p_k \right)_{0,E} = 0 \quad \forall p_k \in \mathbb{P}_k(E) \setminus \mathbb{P}_{k-2}(E) \end{aligned} \right\}, \tag{6}$$

where  $\mathbb{P}_k(E) \setminus \mathbb{P}_{k-2}(E)$  denotes the space of polynomials of degree exactly  $k$  or  $k - 1$ . Here,  $\Pi_k^{\nabla, E} v_h$  is a projection operator that will be defined later in Section 3.3.2. The dimension of this discrete space is given by the formula

$$N_E := \frac{k(k+1)}{2} n_f + \frac{k(k-1)(k+1)}{6}, \tag{7}$$

where  $n_f$  is the number of faces of the polyhedron  $E$ . Then, to uniquely determine a function  $v_h \in V_h^k(E)$ , we can take the following set of linear operators as Dofs:

- D1: the face moments on each face  $f \in \Sigma_h^E$ :

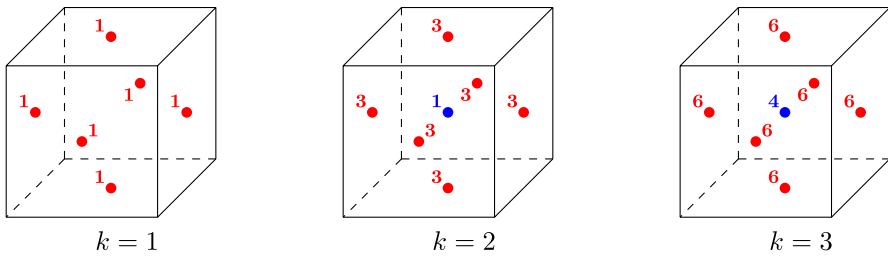
$$\frac{1}{|f|} \int_f v_h m_\ell \, ds, \quad \forall m_\ell \in \mathcal{M}^{k-1}(f); \tag{8}$$

- D2: the bulk moments in  $E$ :

$$\frac{1}{|E|} \int_E v_h m_\ell \, dE, \quad \forall m_\ell \in \mathcal{M}^{k-2}(E). \tag{9}$$

The proof that such choice of Dofs is unisolvent for  $V_h^k(E)$  can be found in [1] In Fig. 1, we represent the DoFs on a cube for. Before showing how to use such a space to build the global one, we provide some comments.

First, the construction of a 3D nonconforming VE space is more natural than that of its conforming counterpart. In the conforming case, the definition of the VE space



**Fig. 1** Degrees of freedom for a cube. In red we have the Dofs associated to a face of the polyhedron, while in blue we have the moments associated to the interior of the polyhedron

on a polyhedron requires the introduction of a VE space also on each of its faces. Moreover, in order to compute the projection operators, the virtual element spaces used are not those originally proposed in [3], but are enriched with the so-called enhancing property [13]. In contrast, as shown in (6), for nonconforming case there is no need to define such a space on faces: the VE space is directly defined inside the polyhedron.

Second, a comparison with the two-dimensional counterpart [1] shows that both the local space  $V_h^k(E)$  and Dofs exhibit a dimensional extension (with edges becoming faces, and faces becoming volumes) while preserving their definitions and the underlying structure of the 2D formulation.

Finally, the VE space introduced in (6) differs from the one proposed in [6]. The main difference lies in the regularity of the Laplacian of a generic function  $v_h \in V_h^k(E)$ . In our case, we have  $\Delta v_h \in \mathbb{P}_k(E)$ , while in [6],  $\Delta v_h \in \mathbb{P}_{k-2}(E)$ . This difference stems from the presence of the enhancing property in the space  $V_h^k(E)$  defined here:

$$\left( v_h - \Pi_k^{\nabla, E} v_h, p_k \right)_E = 0 \quad \forall p_k \in \mathbb{P}_k(E) \setminus \mathbb{P}_{k-2}(E).$$

Indeed, in order to endow the space with this property, its dimension must first be increased and then appropriately restricted to enforce it [13].

Then, the global VE space of order  $k$  is obtained by gluing the local spaces across the faces of the polyhedra:

$$V_h^k(\Omega_h) := \left\{ v_h \in H^1(\Omega_h) \quad : \quad v_h|_E \in V_h^k(E) \quad \forall E \in \Omega_h, \right. \\ \left. \int_f \llbracket v_h \rrbracket q \, df = 0 \quad \forall q \in \mathbb{P}_{k-1}(f) \quad \forall f \in \Sigma_h \right\}. \tag{10}$$

This space is not globally continuous; instead, continuity is enforced weakly across interior faces by requiring that the jumps vanish when tested against polynomials of degree up to  $k - 1$ .

### 3.3 Projectors

A generic function  $v_h \in V_h^k(E)$  is virtual, i.e., it is only known through its degrees of freedom. The VEM defines suitable projection operators that are essential to approxi-

mate the linear and bilinear forms in Problem (2). Following the approach of [2, 14], this section provides a *guide* on how to implement these projection operators. In particular, compared to the previously cited works, we place greater emphasis on the implementation details and the structure of the matrices involved.

### 3.3.1 The projection $\Pi_{k-1}^{0,f}$

Given a face  $f \in \Sigma_h^E$  of a polyhedron  $E$ , we define the projection operator  $\Pi_{k-1}^{0,f} : V_h^k(E)|_f \rightarrow \mathbb{P}_{k-1}(f)$  as

$$(p_{k-1}, v_h - \Pi_{k-1}^{0,f} v_h)_f = 0 \quad \forall p_{k-1} \in \mathbb{P}_{k-1}(f), \tag{11}$$

where  $(\cdot, \cdot)_f$  denotes the  $L^2$  inner product on the face  $f$ .

Since  $\Pi_{k-1}^{0,f} v_h$  is an element of  $\mathbb{P}_{k-1}(f)$ , in (11), we can let  $p_{k-1}$  vary in  $\mathcal{M}^{k-1}(f)$ , and we can represent such projection operator in the basis  $\mathcal{M}^{k-1}(f)$ :

$$\Pi_{k-1}^{0,f} v_h = \sum_{|j| \leq k-1} q^j m^j.$$

Then, (11) becomes:

$$\sum_{|j| \leq k-1} q^j (m^j, m^z)_f = (v_h, m^z)_f \quad \forall |z| \leq k-1.$$

This linear system can be written in a more compact form

$$\mathbb{H}_f \underline{q} = \underline{b},$$

where  $\underline{q}$  is a vector with the unknown polynomial coefficients. The entries of the matrix  $\mathbb{H}_f$  are given by inner products of scaled monomials defined on the face  $f$ . Therefore, by applying an appropriate quadrature rule for polygons [15], they can be computed explicitly. The components of the vector  $\underline{b}$  correspond exactly to the degrees of freedom of type D1 of the face  $f$  we are considering scaled by  $|f|$ , and are thus computable even though they involve the integration of the virtual function  $v_h$ .

**Remark 2** Given a face  $f$  and a virtual basis function  $\phi_h$  associated with a degree of freedom of type D1, the projection operator is nonzero *only* if  $\phi_h$  is associated with the same face  $f$ . In particular, the corresponding vector  $\underline{b}$  is different from zero only in the component corresponding to the polynomial that appears in the definition of the degree of freedom of type D1.

Moreover, for all basis functions associated with degrees of freedom of type D2, the projection operator vanishes.

**Remark 3** If we were in two dimensions, the faces correspond to edges and one can simply verify that the projection  $\Pi_{k-1}^{0,f}$  on an edge does not depend on the edge length. In fact, if we look at the entries of the matrix  $H_f$  and at the definition of the monomials  $m^j$  in one dimension, we can observe that, after a change of variables, the integrals do not depend on the edge size. This implies that it is possible to compute the projection one time for all the edges. Similarly, in three dimensions, it is possible to verify that if two faces have the same shape, the projection does not depend on the face size. This is particularly useful in the case where the polyhedrons are formed by the same faces like a cube decomposition of a mesh.

### 3.3.2 The projection $\Pi_k^{\nabla,E}$

Given a polyhedron  $E \in \Omega_h$ , we define a projection operator from the VE space to the polynomial space of order  $k$ ,  $\Pi_k^{\nabla,E} : V_h^k(E) \rightarrow \mathbb{P}_k(E)$  based on the  $H^1(E)$  semi-inner product. The operator  $\Pi_k^{\nabla,E}$  is such that, for any  $p_k \in \mathbb{P}_k(E)$ , the following condition holds:

$$(\nabla p_k, \nabla(v_h - \Pi_k^{\nabla,E} v_h))_E = 0 \quad \forall p_k \in \mathbb{P}_k(E). \tag{12}$$

As before,  $\Pi_k^{\nabla,E} v_h$  is an element of  $\mathbb{P}_k(E)$  so we can represent such projection operator in the basis  $\mathcal{M}^k(E)$ :

$$\Pi_k^{\nabla,E} v_h = \sum_{|j| \leq k} r^j m^j. \tag{13}$$

Since (12) involves only the gradients of the polynomials, it does not provide enough conditions to uniquely determine all the coefficients of  $\Pi_k^{\nabla,E} v_h$  in the chosen polynomial basis  $\mathcal{M}^k(E)$ . In particular, the component in the kernel of the gradient operator, i.e., the constant part, remains undetermined. To ensure uniqueness, we complement the definition by enforcing the following condition, which effectively fixes the average value of the projection on the boundary:

$$\int_{\partial E} \Pi_k^{\nabla,E} v_h \, df = \int_{\partial E} v_h \, df. \tag{14}$$

This choice guarantees the projection is well-posed and computable using only available degrees of freedom. With a slight abuse of notation, we refer to the integrals in (14) as  $(\cdot, 1)_{\partial E}$ .

Therefore, to find the coefficients  $r^j$ , we set the following linear system

$$\sum_{|j| \leq k} r^j (m^j, 1)_{\partial E} = (v_h, 1)_{\partial E}, \tag{15}$$

$$\sum_{|j| \leq k} r^j (\nabla m^j, \nabla m^z)_E = (\nabla v_h, \nabla m^z)_E \quad \forall 0 < |z| \leq k. \tag{16}$$

This linear system can be written in a more compact form as

$$G \underline{r} = \underline{c}. \tag{17}$$

The entries of the matrix  $G$  are given by integrals of monomials or inner product between gradients of monomials defined inside the polyhedron  $E$  so they are computable [15]. In order to understand if we are able to compute the unknown coefficients of the vector  $\underline{r}$ , we have to check if the right hand side is computable although it contains the virtual function  $v_h$ . To compute the first component of the vector  $\underline{c}$ , we observe that

$$(v_h, 1)_{\partial E} = \sum_{f \in \partial E} \int_f v_h \, df = \sum_{f \in \partial E} \int_f \Pi_{k-1}^{0,f} v_h \, df.$$

**Remark 4** The integral in (14) is nonzero only if  $v_h$  corresponds to a basis function associated with a degree of freedom of type D1, where  $m_\ell = m_0 = 1$ . In this case, the integral is equal to the face size.

Then, for all the other components, we integrate by parts and we get

$$\begin{aligned} (\nabla v_h, \nabla m^z) &= \int_E \nabla v_h \cdot \nabla m^z \, dE \\ &= - \int_E v_h \Delta m^z \, dE + \sum_{f \in \partial E} \int_f (\nabla m^z \cdot \mathbf{n}^E) v_h \, df \\ &= - \int_E v_h \Delta m^z \, dE + \sum_{f \in \partial E} \int_f (\nabla m^z \cdot \mathbf{n}^E) \Pi_{k-1}^{0,f} v_h \, df. \end{aligned}$$

Since  $\Delta m^z$  is a polynomial of degree  $k - 2$ , it is computable from the degrees of freedom D2. Then, since  $\nabla m^z \cdot \mathbf{n}^E$  is also a polynomial of degree  $k - 1$ , we can exploit the projection operator introduced in Section 3.3.1, to compute it.

**Remark 5** The previous computations were carried out for a generic virtual function  $v_h \in V_h^k(E)$ . However, if  $v_h$  is taken to be one of the basis functions associated with the degrees of freedom D2, then the boundary integrals vanish identically. Conversely, if  $v_h$  corresponds to a basis function associated with a boundary degree of freedom D1 and face  $\bar{f}$ , then the volume integral vanishes, and only the boundary contribution on  $\bar{f}$  remains.

### 3.3.3 The projection $\Pi_k^{0,E}$

Given a polyhedron  $E \in \Omega_h$ , we define a projection operator from the VE space to the polynomial space of order  $k$ ,  $\Pi_k^{0,E} : V_h^k(E) \rightarrow \mathbb{P}_k(E)$  via the  $L^2$  inner product:

$$(p_k, v_h - \Pi_k^{0,E} v_h)_E = 0 \quad \forall p_k \in \mathbb{P}_k(E). \tag{18}$$

As before, the conditions on  $p_k \in \mathbb{P}_k(E)$  can be equivalently replaced by the basis elements  $\mathcal{M}^k(E)$  and we can also expand the polynomial  $\Pi_k^{0,E} v_h$  considering the same basis, i.e.,

$$\Pi_k^{0,E} v_h = \sum_{|\mathcal{J}| \leq k} s^{\mathcal{J}} m^{\mathcal{J}}. \tag{19}$$

Hence, to find the coefficients  $s^{\mathcal{J}}$ , we build the following linear system

$$\sum_{|\mathcal{J}| \leq k} s^{\mathcal{J}} (m^{\mathcal{J}}, m^{\mathcal{z}})_E = (v_h, m^{\mathcal{z}})_E \quad \forall 0 \leq |\mathcal{z}| \leq k, \tag{20}$$

which can be written in a more compact form as

$$H \underline{s} = \underline{d}. \tag{21}$$

The entries of the matrix  $H$  are given by inner product between monomials defined inside the polyhedron  $E$ , hence they are computable [15]. The right hand side,  $\underline{d}$ , contains the virtual function  $v_h$ , and we need to check if these components can be computed. First, we observe that for  $0 \leq |\mathcal{z}| \leq k - 2$

$$(v_h, m^{\mathcal{z}})_E = \int_E v_h m^{\mathcal{z}} \, dE, \tag{22}$$

we have exactly the degrees of freedom  $D_2$  so these integrals can be computed. Finally, when  $k - 1 \leq |\mathcal{z}| \leq k$ , we can exploit the enhancing property of the space  $V_h^k(E)$  to compute them,

$$(v_h, m^{\mathcal{z}})_E = \int_E v_h m^{\mathcal{z}} \, dE = \int_E \Pi_k^{\nabla,E} v_h m^{\mathcal{z}} \, dE. \tag{23}$$

The left hand side of this last equation is an integral of the product of two polynomials and thus can be computed using appropriate quadrature rules defined on polyhedra.

**Remark 6** It is worth noticing that for  $k = 1$  the projection  $\Pi_k^{0,E} v_h$  is defined *only* with the conditions of (23). As a consequence, the projection  $\Pi_k^{\nabla,E} v_h$  and  $\Pi_k^{0,E} v_h$  coincide.

**Remark 7** The previous computations were carried out for a generic virtual function  $v_h \in V_h^k(E)$ . If  $v_h$  is taken to be one of the basis functions associated with the degrees of freedom  $\mathcal{D}1$ , (22) is always zero. On the other hand, if it is a basis functions associated with the degrees of freedom  $\mathcal{D}2$ , only one integral of (22) is different from zero and it is equal to the volume of the polyhedron  $E$ . On the other hand, if we need to compute (23) for a basis function associated with a degree of freedom, we exploit the fact that

$$(\Pi_k^{\nabla,E} v_h, m^z)_E = \sum_{|j| < k} s^j (m^j, m^z)_E,$$

where  $\{s^j\}$  are the coefficients of  $\Pi_k^{\nabla,E} v_h$  obtained in (17), and  $(m^j, m^z)_E$  corresponds to the column of  $H$  associated with  $m^z$ . Hence, all these quantities are already available.

### 3.3.4 The projection $\Pi_{k-1}^{0,E}$

Given a polyhedron  $E \in \Omega_h$ , we define a projection operator that acts on the gradients of virtual element functions  $\Pi_{k-1}^{0,E} : \nabla V_h^k(E) \rightarrow [\mathbb{P}_{k-1}(E)]^3$ . This projection operator is based on the  $L^2$  inner product through the condition

$$\left( \mathbf{p}_{k-1}, \nabla v_h - \Pi_{k-1}^{0,E} \nabla v_h \right)_E = 0 \quad \forall \mathbf{p}_{k-1} \in [\mathbb{P}_{k-1}(E)]^3. \tag{24}$$

Starting from the scalar monomial basis  $\mathcal{M}^{k-1}(E)$ , we construct the following basis for vector evaluated polynomials

$$\mathcal{M}^{k-1}(E) := \left\{ \begin{bmatrix} m^j \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ m^j \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ m^j \end{bmatrix} : 0 \leq |j| \leq k-1 \right\}.$$

The conditions on  $\mathbf{p}_{k-1} \in [\mathbb{P}_{k-1}(E)]^3$  can be equivalently replaced by the elements of  $\mathcal{M}^{k-1}(E)$ . Hence, the resulting linear system can be written in the form

$$\sum_{m^z \in \mathcal{M}^{k-1}(E)} t^j (m^j, m^z)_E = (\nabla v_h, m^z)_E \quad \forall m^z \in \mathcal{M}^{k-1}(E), \tag{25}$$

which can be written in a more compact form as

$$H_v \underline{t} = \underline{e}. \tag{26}$$

**Implementation Detail 1** Exploiting the structure of the basis elements of  $\mathcal{M}^{k-1}(E)$ , and ordering its elements so that the first group corresponds to the first component,

the second to the second, and the third to the third one, the matrix  $H_v$  acquires the following block-diagonal structure

$$H_v := \begin{bmatrix} H & 0 & 0 \\ 0 & H & 0 \\ 0 & 0 & H \end{bmatrix},$$

where  $H$  is exactly the matrix in (21) resized to the polynomials of degree up to  $k - 1$ . The right hand side contains virtual functions, hence we have to check if we are able to compute all the integrals. As for the projection operator  $\Pi_k^{\nabla, E} v_h$ , we use the integration by parts and the projection operator  $\Pi_{k-1}^{0, f}$ :

$$\begin{aligned} \int_E \nabla v_h \cdot \mathbf{m}^z \, dE &= - \int_E v_h \operatorname{div}(\mathbf{m}^z) \, dE + \int_{\partial E} (\mathbf{n} \cdot \mathbf{m}^z) v_h \, df \\ &= - \int_E v_h \operatorname{div}(\mathbf{m}^z) \, dE + \sum_{f \in \partial E} \int_f (\mathbf{n} \cdot \mathbf{m}^z) v_h \, df \\ &= - \int_E v_h \operatorname{div}(\mathbf{m}^z) \, dE + \sum_{f \in \partial E} \int_f (\mathbf{n} \cdot \mathbf{m}^z) \Pi_{k-1}^{0, f} v_h \, df. \end{aligned}$$

Since we are considering a vector evaluated polynomial space of order  $k - 1$ ,  $\operatorname{div}(\mathbf{m}^z)$  is a polynomial of degree  $k - 2$ , and it is computable from the degrees of freedom D2.

**Remark 8** The previous computations were made considering a generic VE function  $v_h$  but, if we take a basis VE function, these computations simplify in a similar way to Remark 7.

**Implementation Detail 2** Since we are considering straight faces, the normal vector is constant, hence in the computation of the boundary integral only the component of the normal is varying. In fact, if we consider the following vector evaluated monomial

$$\mathbf{m}_x = \begin{bmatrix} m^z \\ 0 \\ 0 \end{bmatrix},$$

we have

$$\int_f (\mathbf{n}_x m^z) \Pi_{k-1}^{0, f} v_h \, df = \mathbf{n}_x \int_f m^z \Pi_{k-1}^{0, f} v_h \, df,$$

where  $\mathbf{n}_x$  is the first component of the normal  $\mathbf{n}$ . For a practical point of view, it is possible to exploit this fact and save computations.

### 3.4 Linear and bilinear forms

In this section, we focus on the computation of all the linear and bilinear forms introduced in Section 2. Specifically, since each form admits a decomposition into element-wise contributions, we focus on the definition and implementation of the local forms on a generic element  $E$ .

#### 3.4.1 The bilinear form $a(\cdot, \cdot)$

In this subsection we consider the bilinear form defined in (3). As usual in the VE framework, the local contributions of this bilinear form are replaced by a discrete approximation of them:

$$a(u, v) = \sum_{E \in \Omega_h} a^E(u, v) \approx \sum_{E \in \Omega_h} a_h^E(u_h, v_h),$$

where we defined the bilinear form  $a_h^E : V_h^k(E) \times V_h^k(E) \rightarrow \mathbb{R}$  as

$$a_h^E(u_h, v_h) := \int_E \kappa \Pi_{k-1}^{0,E} \nabla u_h \cdot \Pi_{k-1}^{0,E} \nabla v_h \, dE + \bar{\kappa}_E \mathcal{S}_a^E((I - \Pi_k^{\nabla,E})u_h, (I - \Pi_k^{\nabla,E})v_h), \tag{27}$$

where  $\bar{\kappa}_E$  is the trace of the tensor  $\kappa$  evaluated at the barycentre of  $E$ . The form  $\mathcal{S}^E$  is any positive definite bilinear form that satisfies the following properties:

- it is zero if at least one of the entry is a polynomial of degree  $k$ ,
- it scales as the continuous form  $a^E(\cdot, \cdot)$ , i.e., there exists two positive values  $\alpha_*$  and  $\alpha^*$  such that

$$\alpha_* a^E(v_h, v_h) \leq \mathcal{S}_a^E(v_h, v_h) \leq \alpha^* a^E(v_h, v_h) \quad \forall v_h \in \ker(\Pi_k^{\nabla,E}). \tag{28}$$

There are different choices for the bilinear form  $\mathcal{S}^E$ , we refer the reader to [16] for a complete survey.

In Section 4.1, we make a numerical analysis on the robustness of the method with the most common ones: the classical Dofi–Dofi [3] and the D-Recipe [13].

**Remark 9** A more classical approach to discretize the bilinear form  $a^E(\cdot, \cdot)$  is considering this approximation

$$\tilde{a}_h^E(u_h, v_h) := \int_E \kappa \nabla \Pi_k^{\nabla,E} u_h \cdot \nabla \Pi_k^{\nabla,E} v_h \, dE + \bar{\kappa}_E \mathcal{S}_a^E((I - \Pi_k^{\nabla,E})u_h, (I - \Pi_k^{\nabla,E})v_h). \tag{29}$$

However, it was shown in [17] for conforming VEM in 2D that such a choice results in a significant loss of convergence when  $k \geq 3$ . In Section 4.3, we give the numerical evidence about this fact for the VE spaces here proposed. The VE approxima-

tion (27) and (29) are indeed equivalent for any degree  $k$  only in the case where  $\kappa \nabla u_h$  is itself a gradient of a function.

**Implementation Detail 3** During the assembling the stiffness matrix, the integral in (27) can be computed for each polyhedron basis function using a simple matrix-by-matrix product. Let  $\Pi_*^{0,G}$  be a matrix defined as

$$\Pi_*^{0,G} = \begin{bmatrix} t_1 & t_2 & \dots & t_n \end{bmatrix}$$

whose columns  $t_i$  are the solution of the linear system (26) associated with the  $i$ -th VE basis function, and let  $H_\kappa$  be a matrix whose entries are

$$(\kappa m^j, m^i)_E \quad \forall m^i, m^j \in \mathcal{M}^{k-1}(E).$$

Hence, we can obtain the local matrix associated with the element  $E$  that represent the first integral of (27) via this chain of matrix product

$$(\Pi_*^{0,G})^\top H_\kappa \Pi_*^{0,G}.$$

### 3.4.2 The bilinear form $m(\cdot, \cdot)$

The procedure for constructing the bilinear form  $m(\cdot, \cdot)$  is similar to the one described in Section 3.4.1. Similarly, the continuous form is decomposed over the polyhedra and replaced by a discrete approximation:

$$m(u, v) = \sum_{E \in \Omega_h} m^E(u, v) \approx \sum_{E \in \Omega_h} m_h^E(u_h, v_h),$$

where we defined the bilinear form  $m_h^E : V_h^k(E) \times V_h^k(E) \rightarrow \mathbb{R}$  as

$$m_h^E(u_h, v_h) := \int_E \Pi_k^{0,E} u_h \Pi_k^{0,E} v_h \, dE + \mathcal{S}_m^E((I - \Pi_k^{0,E})u_h, (I - \Pi_k^{0,E})v_h). \tag{30}$$

The stabilization term  $\mathcal{S}_m^E(\cdot, \cdot)$  is a positive definite bi-linear form that vanishes if at least one of the entries is a polynomial of degree  $k$ , and it must scale consistently with the continuous form  $m^E(\cdot, \cdot)$ , i.e.,

$$\mu_* m^E(v_h, v_h) \leq \mathcal{S}_m^E(v_h, v_h) \leq \mu^* m^E(v_h, v_h) \quad \forall v_h \in \ker(\Pi_k^{0,E}). \tag{31}$$

**Implementation Detail 4** During the assembly of the mass term of the linear system, the integral in (30) can be efficiently computed for each polyhedron basis function

via a simple matrix-by-matrix product as its counterpart in (27). Let  $\Pi_*^0$  be a matrix defined as

$$\Pi_*^0 = \begin{bmatrix} \underline{s}_1 & \underline{s}_2 & \dots & \underline{s}_n \end{bmatrix}$$

whose columns  $s_i$  are the solutions of the linear system (21) associated with the  $i$ -th VE basis function. Hence, we can obtain the local matrix associated with the element  $E$ , representing the first integral of (30), can be obtained via the following chain of matrix products:

$$(\Pi_*^0)^\top H \Pi_*^0,$$

where  $H$  is the *same* matrix as the linear system (21).

**Implementation Detail 5** Considering the classical Dofi–Dofi stabilization, both stabilization terms  $\mathcal{S}_a^E(\cdot, \cdot)$  and  $\mathcal{S}_m^E(\cdot, \cdot)$  can be expressed as a sequence of matrix products [14]. Let  $D$  be the matrix defined as follows

$$D = \begin{bmatrix} dof_1(m^{j_1}) & dof_1(m^{j_2}) & \dots & dof_1(m^{j_{n_c}}) \\ dof_2(m^{j_1}) & dof_2(m^{j_2}) & \dots & dof_2(m^{j_{n_c}}) \\ \vdots & \vdots & \ddots & \vdots \\ dof_{N_E}(m^{j_1}) & dof_{N_E}(m^{j_2}) & \dots & dof_{N_E}(m^{j_{n_c}}) \end{bmatrix}.$$

where  $dof_i(\cdot)$  is a linear functional representing the  $i$ -th degree of freedom. The stabilization  $\mathcal{S}_a^E(\cdot, \cdot)$  becomes

$$h_E (I - D \Pi_*^\nabla)^\top (I - D \Pi_*^\nabla), \tag{32}$$

where  $\Pi_*^\nabla$  is a matrix whose columns are the solution of the linear system (17), and  $I$  is the identity matrix. Similarly, the stabilization  $\mathcal{S}_m^E(\cdot, \cdot)$  is given by

$$|E| (I - D \Pi_*^0)^\top (I - D \Pi_*^0), \tag{33}$$

where  $\Pi_*^0$  is defined in Implementation Detail 4.

The so-called D-Recipe stabilization, introduced in [13], was proposed as an alternative to the Dofi–Dofi term, which may become too restrictive for large  $k$ , particularly in three-dimensional settings. In this approach, different basis functions may lead to different energy values. To account for this, the following diagonal stabilization form is adopted:

$$\tilde{\mathcal{S}}_a^E(\varphi_i, \varphi_i) := \max\{h_E, (\Pi_{k-1}^{0,E} \nabla \varphi_i, \Pi_{k-1}^{0,E} \nabla \varphi_i)_E\},$$

where the set  $\{\varphi_i\}$  denotes the nodal basis functions of  $V_h^k(E)$ . Accordingly, the matrix (32) (and analogously (33)) should be replaced by

$$(\mathbf{I} - \mathbf{D} \mathbf{\Pi}_*^\nabla)^\top \tilde{\mathbf{R}} (\mathbf{I} - \mathbf{D} \mathbf{\Pi}_*^\nabla),$$

where  $\tilde{\mathbf{R}}$  is a diagonal matrix defined by

$$(\tilde{\mathbf{R}})_{ii} = \max\{h_E, (\mathbf{\Pi}_{k-1}^{0,E} \nabla \varphi_i, \mathbf{\Pi}_{k-1}^{0,E} \nabla \varphi_i)_E\}.$$

We conclude by observing that the quantity  $(\mathbf{\Pi}_{k-1}^{0,E} \nabla \varphi_i, \mathbf{\Pi}_{k-1}^{0,E} \nabla \varphi_i)_E$  is already required in the assembly of the local stiffness matrix. Therefore, no additional computations are needed to construct the stabilization matrix.

### 3.4.3 The right-hand side

The load term is also split among the contributions of each polyhedron, and the virtual function is simply replaced with its  $L^2$  projection:

$$F(v) = \sum_{E \in \Omega_h} \int_E f v \, dE \approx \sum_{E \in \Omega_h} F_h^E(v_h),$$

where we define the linear form  $F_h^E : V_h^k(E) \rightarrow \mathbb{R}$  as

$$F_h^E(v_h) := \int_E f \Pi_k^{0,E} v_h \, dE. \tag{34}$$

It is worth noting that in this paper we consider a more accurate approximation of the load term than those proposed in [1, 6]. This choice does not improve the convergence rate but marginally reduces the error of the discrete solution. As shown in [1, 6], using a lower order approximation does not affect the convergence rate of the error measured in the  $L^2$  norm or in the  $H^1$  seminorm.

**Implementation Detail 6** When assembling the Right-Hand Side of the linear system, the integral in (34) can be computed for each polyhedron basis function using a simple matrix-vector product. Let  $\mathbf{H}_f$  be a vector whose entries are

$$(f, m^z)_E \quad \forall m^z, \forall 0 \leq |z| \leq k.$$

Hence, we can obtain the local vector associated with the element  $E$  via the following matrix-vector product:

$$(\mathbf{\Pi}_*^0)^\top \mathbf{H}_f,$$

where  $\mathbf{\Pi}_*^0$  is the matrix introduced in Implementation Detail 4.

### 3.5 Discrete problem

Based on the definitions of the global spaces, projection operators, and the linear and bilinear forms, we can now introduce the discrete problem:

$$\begin{cases} \text{find } u_h \in V_h^k(\Omega_h) \text{ such that:} \\ a_h(u_h, v_h) + m_h(u_h, v_h) = F_h(v_h) \quad \forall v_h \in V_h^k(\Omega_h), \end{cases} \tag{35}$$

where the global bilinear forms are defined as

$$a_h(u_h, v_h) := \sum_{E \in \Omega_h} a_h^E(u_h, v_h) \quad \text{and} \quad m_h(u_h, v_h) := \sum_{E \in \Omega_h} m_h^E(u_h, v_h),$$

and the linear form

$$F_h(v_h) := \sum_{E \in \Omega_h} F_h^E(v_h).$$

## 4 Numerical Experiments

In this section, we investigate the numerical behaviour of the nonconforming scheme in different settings. First, we conduct a convergence analysis with respect to the polynomial degree  $k$ , considering different choices for the stabilization term (see Section refsec:num:kconv), and a convergence analysis with respect to the mesh size  $h$  (see Section 4.2). In Section 4.3, we provide numerical evidence supporting Remark 9. Finally, in Section 4.4, we compare the conforming and nonconforming VEM from an algebraic view point, i.e., we analyse execution time, dimension of the linear system, and the sparsity of the global matrix. All the numerical experiments are implemented using the C++ library `vem++` [19].

In all these experiments, we consider the following error indicators and numerically verify the expected theoretical convergence rate

- $H^1$ -seminorm error

$$e_{H^1} := \sqrt{\sum_{E \in \Omega_h} \left\| \nabla(u - \Pi_k^{\nabla, E} u_h) \right\|_{0,E}^2} \approx h^k,$$

- $L^2$ -norm error

$$e_{L^2} := \sqrt{\sum_{E \in \Omega_h} \left\| u - \Pi_k^{0,E} u_h \right\|_{0,E}^2} \approx h^{k+1}.$$

To this end, we consider the problem defined in (1), where the domain  $\Omega$  is the truncated octahedron,  $\kappa$  is the identity tensor and the right-hand side  $f$  is chosen so that the exact solution is

$$u(x, y, z) := \sin(\pi x) \cos(\pi y) \cos(\pi z), \tag{36}$$

and we construct a sequence of four meshes with decreasing mesh size  $h$ . In order to evaluate the robustness of the nonconforming VEM with respect to element distortion, we consider three different mesh types of Voronoi tessellations:

- `stru`: seed points are uniformly distributed inside the domain,
- `voro`: the positions of the seed points is optimized via a Lloyd algorithm [19],
- `rand`: seed points are randomly distributed inside the domain.

These three types of tessellation are characterised by an increasing level of distortion of the polyhedra. Indeed, `stru` meshes have almost hexahedral elements, `voro` meshes may contain polyhedra with small faces and/or edges, and `rand` meshes typically exhibit highly distorted elements. In Fig. 2, we show the interior of these mesh types to have an idea of the distortion of each mesh type. In the following examples, we refer to these meshes using their type and an integer number from 1 to 4. Hence, `rand2` refers to the second refinement level of the domain using seed points randomly distributed inside the domain.

### 4.1 Convergence analysis in $k$

In this subsection, we test the convergence with respect to the approximation degree  $k$ , considering two choices for the stabilization term  $S_a^E$ : the Dofi-Dofi stabilization introduced in [3] and the D-Recipe proposed in [13]. According to Remark 6.3 in [14], the stabilization term  $S_m^E$  is only required when the equation is reaction-dominated, and thus can be set to zero in this case.

We fix the mesh `voro2` and show in Fig. 3 the convergence rates of both  $e_{H^1}$  and  $e_{L^2}$ . Similar results have been observed for the meshes `stru2` and `rand2`.

The results show exponential convergence in terms of  $k$  when using the D-Recipe stabilization, whereas both errors exhibit a slight increase for  $k = 4$  and  $k = 5$

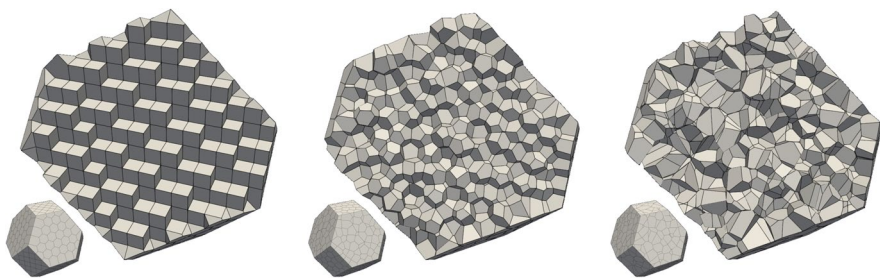


Fig. 2 The interior of each type of mesh where the domain  $\Omega$  is a truncated octahedron

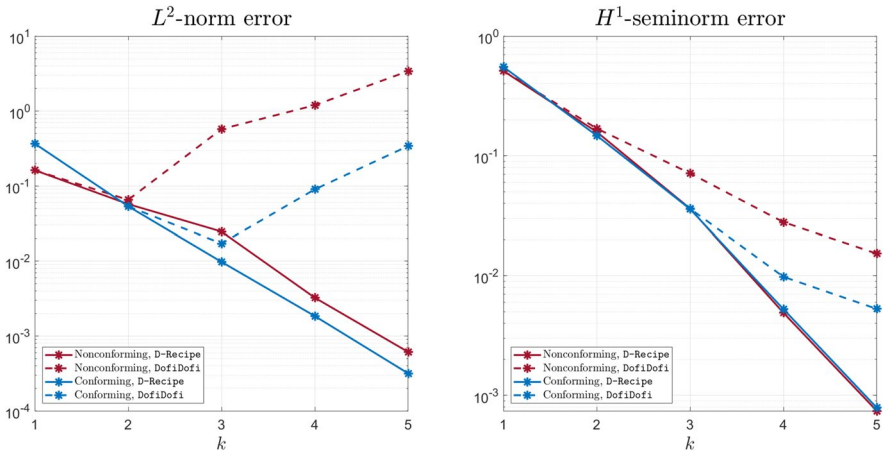


Fig. 3 Convergence analysis in  $k$ : the trends of  $e_{H^1}$  and  $e_{L^2}$  as the approximation degree  $k$  varies

when using the Dof1–Dof1 stabilization. These results are consistent with those reported in [13] for the conforming case (see Section 3.4 therein).

Furthermore, to better highlight this similar behaviour, we have conducted the same experiments using conforming VE spaces as well; see the blue lines in Fig. 3.

### 4.2 Convergence analysis in $h$

In this section, we analyse the convergence rate for the nonconforming VEM for  $k = 1, 2,$  and  $3,$  comparing the results obtained with the mesh types `stru`, `vor`, and `rand`. We employ the D-Recipe stabilization term, since the previous example has shown that this choice provides a more accurate trend for higher degrees.

In Fig. 4, we show the error trends. The convergence lines follow the expected theoretical decay:  $e_{L^2}$  and  $e_{H^1}$  converge as  $h^{k+1}$  and  $h^k$ , respectively.

Furthermore, the convergence lines associated with the different mesh types are close to each other, indicating that the proposed method is robust with respect to element distortion.

### 4.3 Effects of the discretization of the diffusion term

In this section, we compare the discretizations (27) and (29). As already stated in Remark 9, if the bilinear form  $a(\cdot, \cdot)$  involves a non-constant diffusion tensor, the approximation defined in (29) does not achieve the optimal convergence rate, whereas the discretization in (27) behaves as expected.

To this end, we modify the right-hand side  $f$  so that the exact solution remains the one defined in (36), but now

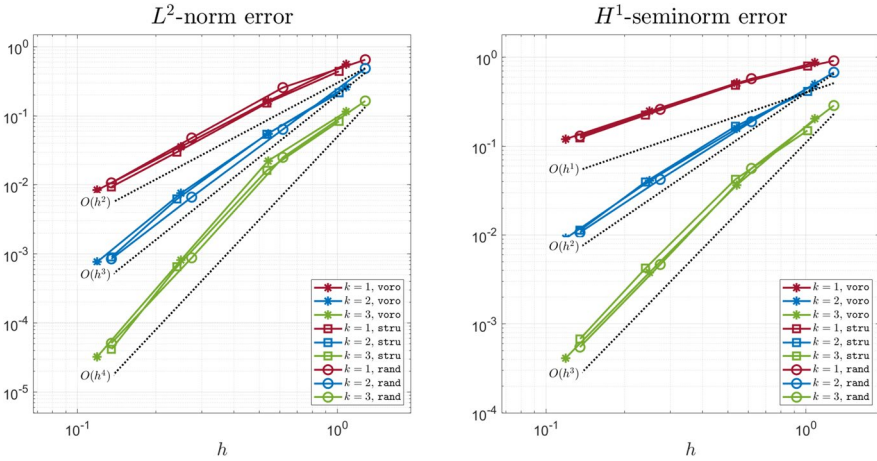


Fig. 4 Convergence analysis in  $h$ : the trends of  $e_{H^1}$  and  $e_{L^2}$  varying  $h$  and mesh types

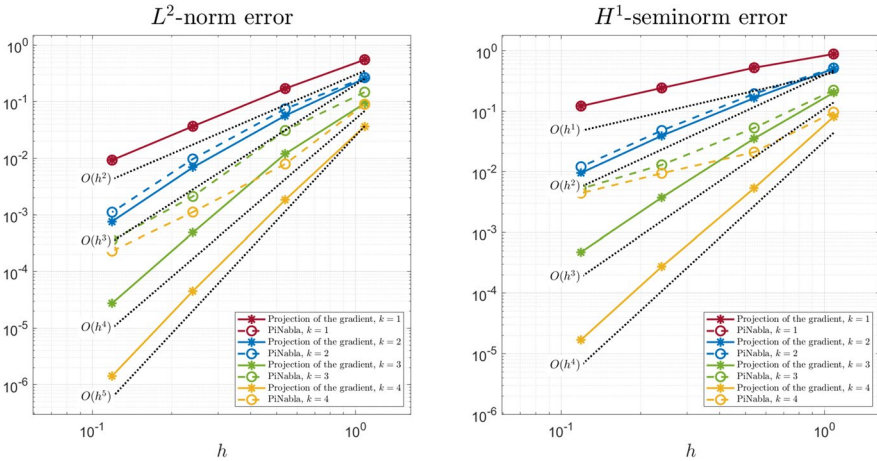


Fig. 5 Effects of the discretization of the diffusion term: comparison between the usage of (29) and (27)

$$\kappa(x, y, z) := \begin{bmatrix} y^2 + 1 & -xy & -xz \\ -xy & z^2 + 1 & -yz \\ -xz & -yz & x^2 + 1 \end{bmatrix}.$$

We analyse the convergence rate of the nonconforming VEM for  $k = 1, 2, 3$ , and  $4$ , using the D-Recipe stabilization and considering only the voro mesh type. Similar results were observed for the other mesh types.

In Fig. 5, we show the convergence lines. The behaviour of the nonconforming VEM in 3D is consistent with the conforming case in 2D: there is a significant loss of convergence rate for  $k \geq 3$  when using the discretization (29). However, we also observe that for  $k = 1$ , the two methods produce the same results. This makes the choices in (29) preferable in the case  $k = 1$ , since they require one fewer projection.

This result is in line with the analysis in [17]. For  $k = 2$ , the two methods behave similarly, but the errors are not exactly the same, although they are comparable.

#### 4.4 Comparison with the conforming method

In this section, we analyse in more detail the characteristic of the global matrix arising from the VE discretization of the model problem (1) and the execution time of the solver used.

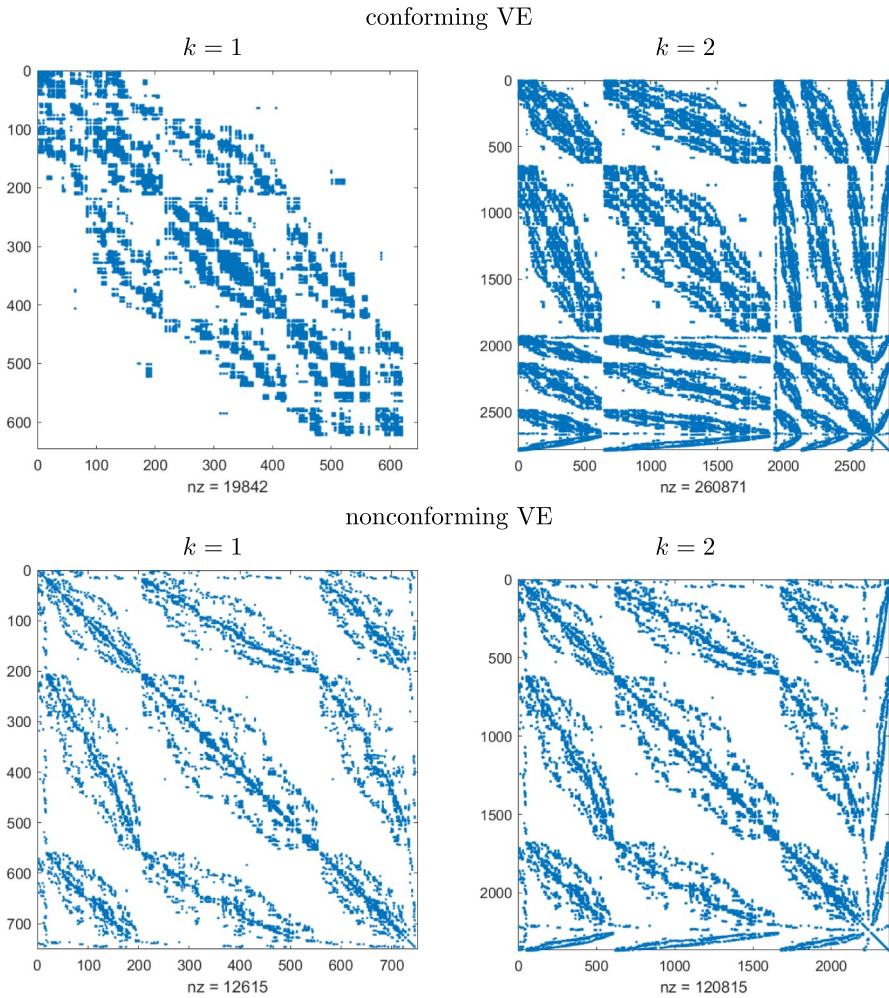
To this end, we focus on the meshes `voro1`, `voro2` and `voro3`. and the direct solver MUMPS [20] which is integrated in `vem++` via the `Petsc` library [21]. Similar results were observed with other types of meshes.

The data are reported in Table 1. The nonconforming method requires less execution time, and this advantage becomes more evident increasing the order  $k$  and/or decreasing  $h$ . It is worth noticing that we take the execution time provided by MUMPS, so the comparison is fair and does not depend on the type of the method but only on the global matrix itself.

Regarding the structure of the global matrix, the data shows that the one arising from the nonconforming VEM is sparser than the conforming one. The difference in sparsity decreases as the polynomial order  $k$  increases. This behaviour is expected, given the locality of interactions in the nonconforming formulation. Indeed, in the nonconforming method each element interacts only with elements that share a face, whereas in the conforming method interactions also occur through shared edges and vertices. Moreover, from Fig. 6, we can compare the sparsity patterns of these two discretizations, We present only the sparsity patterns corresponding to the `voro2` mesh. For  $k = 1$ , the sparsity patterns of the two methods are similar. However, for  $k = 2$  we notice a saddle-point structure for the nonconforming case and a more complex, triple-level saddle-point structure for the conforming one, i.e., the global matrices have the following structure

**Table 1** Comparison with the conforming method: matrix statistics

Elements	Conforming			Nonconforming		
	#Dofs	%Nonzero	Time	#Dofs	%Nonzero	Time
<i>k</i> = 1						
20	41	62.04%	0.102	77	20.02%	0.083
116	404	12.15%	0.454	597	3.53%	0.392
1001	4643	1.31%	4.364	6064	0.39%	3.677
<i>k</i> = 2						
20	237	32.94%	0.498	251	18.45%	0.344
116	2003	6.50%	2.599	1907	3.32%	1.820
1001	21415	0.74%	29.457	19193	0.37%	17.830
<i>k</i> = 3						
20	550	26.49%	1.964	542	17.17%	1.555
116	4431	5.23%	10.268	4046	3.13%	8.240
1001	46253	0.60%	123.262	40388	0.36%	83.053



**Fig. 6** Comparison with the conforming method: sparsity pattern of the global matrices obtained from the mesh `vorO2` using VE approximations of degree  $k = 1$  and  $k = 2$

$$S_c = \begin{bmatrix} A & E & F & G \\ E^\top & B & H & J \\ F^\top & H^\top & C & K \\ G^\top & J^\top & K^\top & D \end{bmatrix} \quad \text{and} \quad S_{nc} = \begin{bmatrix} A & C \\ C^\top & B \end{bmatrix},$$

for the conforming and nonconforming case, respectively. This structure is primarily due to the ordering of the degrees of freedom. Consider the conforming case, in `vem++` the degrees of freedom are organized as follows: first the vertex degrees of freedom, second the ones associated with edges, faces and polyhedra. Since the non conforming case has only degrees of freedom associated with faces and polyhedra (D1 and D2), the linear system arising from this discretization has a saddle-point

structure: the first matrix entries are the degrees of freedom of the type  $D_1$  and the last part of the matrix entries are the degrees of freedom  $D_2$ .

**Acknowledgements** F.D. and M.T. are also members of “Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica” (GNCS-INdAM). This organization is gratefully acknowledged.

**Funding** Open access funding provided by Università degli Studi di Milano - Bicocca within the CRUI-CARE Agreement. The author F.D. and M.T. was partially supported by the European Research Council project NEMESIS (Grant No. 101115663). This support is gratefully acknowledged.

**Data Availability** No datasets were generated or analysed during the current study.

## Declarations

**Ethical Approval** Not Applicable

**Competing interests** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Ayuso De Dios, B., Lipnikov, K., Manzini, G.: The nonconforming virtual element method. *ESAIM: Math. Model. Numer. Anal.* **50**(3), 879–904 (2016)
2. Cangiani, A., Manzini, G., Sutton, O.: Conforming and nonconforming virtual element methods for elliptic problems. *IMA J. Numer. Anal.* **37** (2015). <https://doi.org/10.1093/imanum/drw036>
3. Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: Basic principles of virtual element methods. *Math. Models Methods Appl. Sci.* **23**(01), 199–214 (2013)
4. Beirão da Veiga, L., Lovadina, C., Vacca, G.: Virtual Elements for the Navier–Stokes Problem on Polygonal Meshes. *SIAM J. Numer. Anal.* **56**(3), 1210–1242 (2018)
5. Dassi, F., Lovadina, C., Visinoni, M.: A three-dimensional Hellinger-Reissner virtual element method for linear elasticity problems. *Comput. Methods Appl. Mech. Eng.* **364**, 112910 (2020). <https://doi.org/10.1016/j.cma.2020.112910>
6. Park, H., Kwak, D.Y.: An analysis of nonconforming virtual element methods on polytopal meshes with small faces. *Math. Models Methods Appl. Sci.* **34**(04), 571–596 (2024). <https://doi.org/10.1142/S021820252450009X>
7. Lovadina, C., Perugia, I., Trezzi, M.: A nonconforming virtual element method for advection-diffusion-reaction problems with cip stabilization. *ESAIM: Math. Model. Numer. Anal.* (2025). <https://doi.org/10.1051/m2an/2025023>
8. Cangiani, A., Gyrya, V., Manzini, G.: The nonconforming virtual element method for the stokes equations. *SIAM J. Numer. Anal.* **54** (2016). <https://doi.org/10.1137/15M1049531>
9. Liu, X., Chen, Z.: The nonconforming virtual element method for the navier-stokes equations. *Adv. Comput. Math.* **45**, 51–74 (2018)

10. Zhang, B., Zhao, J., Yang, Y., Chen, S.: The nonconforming virtual element method for elasticity problems. *J. Comput. Phys.* **378**, 394–410 (2019). <https://doi.org/10.1016/j.jcp.2018.11.004>
11. Trezzi, M.: A family of virtual element methods for advection-dominated problems. PhD thesis, Università degli Studi di Pavia (2024). <https://iris.unipv.it/handle/11571/1513196>
12. Adams, R.A., Fournier, J.J.F.: Pure and applied mathematics. Acad. press, (2003)
13. Beirão da Veiga, L., Dassi, F., Russo, A.: High-order virtual element method on polyhedral meshes. *Computers & Mathematics with Applications* **74**(5), 1110–1122 (2017)
14. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: The Hitchhiker’s Guide to the Virtual Element Method. *Math. Models Methods Appl. Sci.* **24**(8), 1541–1573 (2014)
15. Chin, E.B., Sukumar, N.: Scaled boundary cubature scheme for numerical integration over planar regions with affine and curved boundaries. *CMAME* **380**, 113796 (2021). <https://doi.org/10.1016/j.cma.2021.113796>
16. Mascotto, L.: The role of stabilization in the virtual element method: a survey. *Comput. Math. Appl.* **151**, 244–251 (2023)
17. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: Virtual element method for general second-order elliptic problems on polygonal meshes. *Mathematical Models and Methods in Applied Sciences* **26**(04), 729–750 (2016). <https://doi.org/10.1142/S0218202516500160>
18. Dassi, F.: Vem++, a c++ library to handle and play with the virtual element method. *Numerical Algorithms* (2025). <https://doi.org/10.1007/s11075-025-02059-z>
19. Du, Q., Faber, V., Gunzburger, M.: Centroidal voronoi tessellations: Applications and algorithms. *SIAM Rev.* **41**(4), 637–676 (1999)
20. Amestoy, P.R., Duff, I.S., L’Excellent, J.-Y., Koster, J.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.* **23**(1), 15–41 (2001). <https://doi.org/10.1137/s0895479899358194>
21. Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Dener, A., Eijkhout, V., Groppe, W., Karpeyev, D., Kaushik, D., Knepley, M., May, D., Curfman McInnes, L., Mills, R., Munson, T., Rupp, K., Sanan, P., Smith, B., Zampini, S., Zhang, H., Zhang, H.: PETSc users manual. Technical Report ANL-95/11 - Revision 3.13, Argonne National Laboratory (2021)

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.