Computer Methods in Applied Mechanics and Engineering 421 (2024) 116826

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

3D stabilization-free virtual element method for linear elastic analysis

Bing-Bing Xu [*], Peter Wriggers

Institute of Continuum Mechanics, Leibniz University Hannover, Hannover, Germany

ARTICLE INFO

Keywords:
Virtual element method
Stabilization-free
Nonmatching Mesh

ABSTRACT

We present a first-order stabilization-free virtual element method (VEM) for three-dimensional linear elastic problems in this paper. VEM has been increasingly used in various fields of engineering, but the need of stabilization yields a method that cannot be used without care, e.g. in nonlinear engineering applications. In this work, by increasing the order of the strain model, a new virtual element formulation is constructed for three-dimensional problems that does not require any stabilization term. The core concept involves adapting the virtual element space to enable the computation of a higher-order L_2 projection operator, guaranteeing an accurate representation of the element energy in terms of strain and stress. This work describes the calculation process of the original $\mathcal{H}_1$ projection operator and the higher-order L_2 projection operator for three-dimensional problems. Eigenvalue analysis allows to derive an approximate relation between the polynomial order and the number of element vertices. Some benchmark problems illustrate the capability of the stabilization-free VEM for three-dimensional linear elastic problems.

1. Introduction

The Virtual Element Method (VEM), as introduced in [1,2], is a Galerkin projection technique that allows arbitrary polygonal and polyhedral meshes in two- and three-dimensions respectively, including non-convex elements. Different from polygonal techniques such as the Polygonal Finite Element Method (PFEM) [3,4] and Generalized FEM (GFEM) [5], it is not necessary to explicitly construct and evaluate the basis function or shape function in the polygonal or polyhedral elements. This not only allows the method to adapt to more complex meshes but also yields a better match with adaptation techniques to improve calculation efficiency [6,7]. Since this method was proposed in 2013, it has been widely applied in different engineering simulations, including the linear elastic problems [8–12], hyperelastic materials at finite deformations [13–16], contact problems [17–20], elastodynamics problems [21–24], and finite elastoplastic deformations [25–27]. Further applications include phase field modeling [28,29], topology optimization [30], and eigenvalue problems [31,32]. Most of the current VEM applications remain in two dimensions.

The main idea of the virtual element method is the split of the primary variable u into a projection part Πu and a remainder. Then polynomials (with order k) are selected for the calculation of the projection operator Π. This treatment results in an incorrect rank of the element stiffness matrix (except for 2D triangular elements and 3D tetrahedral elements). This means that for all applications mentioned above, a stabilization is necessary to avoid rank deficiency. In the existing literature, two primary types of stabilization techniques have proven effective for classical solid mechanics problems. The first stabilization is based on the degrees of freedom (dof) [8,12]. Another very effective stabilization technique is energy stabilization [14], which has outstanding behavior in the nearly

[*] Corresponding author.
E-mail address: bingbing.xu@ikm.uni-hannover.de (B.-B. Xu).

https://doi.org/10.1016/j.cma.2024.116826
Received 23 November 2023; Received in revised form 12 January 2024; Accepted 30 January 2024
Available online 2 February 2024

incompressible limit and some other nonlinear problems [33]. However, it is undeniable that the existence of the stabilization term in VEM increases the robustness of the method when dealing with nonlinear problems (e.g. elasto-plasticity).

In addition to the aforementioned discussion on constructing the stabilization term, another possibility is to eliminate the need for a stabilization term. In the classical VEM, the incorrect rank of the element stiffness matrix is due to the low order of the strain model. For example, for a first-order VEM ($k = 1$), the strain inside the polygonal or polyhedral element is always constant regardless of the number of element vertices n_E. Therefore, one way to avoid the stabilization term is to construct a mixed VEM in which the strain is independent of the displacement and has a higher-order strain ($p > k - 1$). A first approach in this direction can be found in [34], which proposes an enhanced VEM formulation where the self-stabilized VEM was constructed for $k = 1$. A stabilization-free VEM was also introduced in [35] for the 2D Poisson equation. The basic idea is to modify the first-order virtual element space to allow the computation of a higher-order L_2 projection of the gradient. Besides, the well-posedness for sufficient stabilization was proven for the first-order formulation. It should be mentioned that a similar idea can be found in [36], which proposes a self-stabilized virtual element formulation based on the Hu–Washizu variational approach. The stabilization-free VEM (SFVEM) has been extended also to the Laplacian eigenvalue problem [37] and linear plane elasticity in [38,39]. The above works focus on 2D linear problems. Recently, the stabilization-free VEM ($k = 1$ and $k = 2$) has been extended to the 2D hyperelastic problems, see [40]. Since no stabilization term is required, this method can be applied to solve many nonlinear problems. But so far, there has been no detailed description of extending this method to three-dimensional problems. First work on stabilization-free three-dimensional virtual elements can be found in [41] for the thermal problem.

The main aim of this work is to extend the work in [34,38,40] to three-dimensional elasticity problems. Compared with two-dimensional problems, the three-dimensional approach is more complex. In this work, only linear elastic problems are calculated and analyzed. Similar to the two-dimensional problem, the original function space and the enhancement function space on polygonal faces and polyhedral elements are provided in the first step. Based on the function spaces, the original $\mathcal{H}_1$ projection operation matrix $\boldsymbol{\Pi}^{\nabla}$ for variables and L_2 projection matrix $\boldsymbol{\Pi}^{m}$ for gradients are developed in detail. Using the L_2 projection operator $\Pi^0_{l,E}\nabla$ of the gradient of functions in $\mathcal{H}^1(E)$, a stabilization-free virtual element formulation for three-dimensional linear elastic problems can be obtained. Similar to the technique used in [38,39], the eigenvalue analysis is used to numerically establish a choice of $l \in \mathbb{N}$. To show the generality of a discretization with virtual elements, a nonmatching mesh using polyhedral elements [42,43] is analyzed in this work.

This paper is organized as follows. The basic governing equations for elastic problems are reviewed in Section 2. Then in Section 3, the necessary polynomial basis and different virtual element spaces on polygonal faces and polyhedral elements are given. Next, the $\mathcal{H}_1$ and L_2 projection operators for polyhedral elements and the associated calculation process are provided in Section 4. Based on the above work, the stabilization-free VEM for 3D linear elasticity is provided in Section 5. In Section 6, an eigenvalue analysis will be employed to numerically establish the choice of the order of l of the polynomial necessary in the projection to avoid rank deficiency. Other 3D examples show the robustness of the new stabilization-free virtual element method. The conclusion and discussion will be provided in Section 7.

2. Governing equations

Let us consider an elastic body that occupies the bounded domain $\Omega \in \mathbb{R}^3$ with boundary $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$. Γ_D is the Dirichlet boundary for displacements $\boldsymbol{u}_D \in \mathcal{H}^1(\Omega;\mathbb{R}^3)$ and Γ_N is the Neumann boundary for tractions $\bar{\boldsymbol{t}} \in \mathcal{L}^2(\Gamma_N;\mathbb{R}^3)$ such that $\Gamma_D \cap \Gamma_N = \emptyset$. The strong form of the boundary value problem for linear elasticity is: find $\boldsymbol{u}(\boldsymbol{x}) : \bar{\Omega} \to \mathbb{R}^3$ such that

$$\operatorname{div}\boldsymbol{\sigma} + \boldsymbol{f} = 0, \quad \boldsymbol{x} \in \Omega, \tag{1}$$

$$\boldsymbol{u} = \boldsymbol{u}_D, \quad \boldsymbol{x} \in \Gamma_D \tag{2}$$

$$\boldsymbol{\sigma} \cdot \boldsymbol{n}_N = \bar{\boldsymbol{t}}, \quad \boldsymbol{x} \in \Gamma_N, \tag{3}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor and $\boldsymbol{f}$ is the body force, $\boldsymbol{n}_N$ is the outward normal, div $(\bullet) = \frac{\partial(\bullet)_{ij}}{\partial x_j}\boldsymbol{e}_i$ is the divergence of a tensor quantity. In this work, the body is assumed to undergo small displacements and the engineering strain is used and given by

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left[\nabla\boldsymbol{u} + [\nabla\boldsymbol{u}]^T\right], \tag{4}$$

For small-strain kinematics, the strain energy density function Ψ for the Venant-Kirchhoff model follows as

$$\Psi(\boldsymbol{\varepsilon}) = \frac{\lambda}{2}(\mathrm{tr}(\boldsymbol{\varepsilon}))^2 + \mu\,\mathrm{tr}(\boldsymbol{\varepsilon}^2). \tag{5}$$

where λ and μ are material coefficients. The Cauchy stress tensor can be calculated by

$$\boldsymbol{\sigma} = \frac{\partial\Psi}{\partial\boldsymbol{\varepsilon}} = \lambda\mathrm{tr}(\boldsymbol{\varepsilon})\boldsymbol{I} + 2\mu\boldsymbol{\varepsilon}, \tag{6}$$

where $\boldsymbol{I}$ is the second order identity tensor. Then the stress–strain relation is given by

$$\boldsymbol{\sigma} = \boldsymbol{C} : \boldsymbol{\varepsilon}, \tag{7}$$

where $\boldsymbol{C} := \frac{\partial\boldsymbol{\sigma}}{\partial\boldsymbol{\varepsilon}}$ is the fourth-order constitutive tensor.

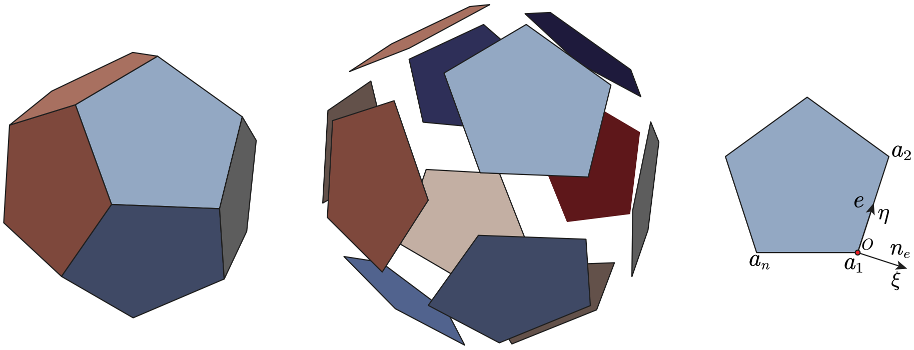


Fig. 1. A polyhedral element and its polygonal surfaces.

The weak form of the above problem is to find the displacement $\boldsymbol{u} \in \mathcal{V}$ where

$$\mathcal{V} := \left\{ \boldsymbol{u} : \boldsymbol{u} \in \left[\mathcal{H}^1(\Omega)\right]^3, \boldsymbol{u} = \bar{\boldsymbol{u}} \text{ on } \Gamma_D \right\}, \tag{8}$$

such that

$$a(\boldsymbol{u}, \boldsymbol{v}) = \ell(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in \mathcal{V}, \tag{9}$$

where the bilinear form $a(\cdot,\cdot) : \left[\mathcal{H}^1(\Omega)\right]^3 \times \left[\mathcal{H}^1(\Omega)\right]^3 \to \mathbb{R}$ and linear function $\ell(\cdot) : \left[\mathcal{H}^1(\Omega)\right]^3 \to \mathbb{R}$ are defined by

$$a(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\, \mathrm{d}\Omega, \tag{10}$$

$$\ell(\boldsymbol{v}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v}\, \mathrm{d}\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{v}\, \mathrm{d}\Gamma. \tag{11}$$

3. Polynomial basis and virtual element spaces

In the virtual element method, a decomposition $\mathcal{T}_h = \left\{\Omega_h\right\}_h$ is introduced where Ω_h is a partition of the computational domain Ω into non-overlapping polyhedron elements. For the polyhedral element E with boundary ∂E as shown in Fig. 1, some parameters are given as:volume $|E|$, barycenter $\boldsymbol{x}_E = (x_E, y_E, z_E)^T$, and diameter h_E. A mesh face $F \in \partial E$ is a planar, two-dimensional subset of $\mathbb{R}^3$ and we denote the set of polygon faces by $\mathcal{F}_h$. We can define a local coordinate system (ξ, η) as shown in Fig. 1. The parameters for the local polygon face are given as area $|F|$, barycenter $\boldsymbol{\xi}_F = (\xi_F, \eta_F)^T$, and diameter h_F. It should be noted that since we need to construct a stabilization-free format, we can only consider the scalar function space here.

Here, we will briefly review the function space required for first-order three-dimensional VEM. See [44–46] for details. Although some high-order enhancement space will be mentioned here, it will be further discussed in Section 4. If readers are confused about the content of this section, they can refer to [44–46] and read Section 4 directly.

3.1. Polynomial basis and scaled monomials

In this work, we denote two spaces of two- and three-dimensional linear polynomials over a polygon F and a polyhedron E by $\mathbb{P}_p(F)$, and $\mathbb{P}_p(E)$, $p \in \mathbb{N}$, respectively. Moreover, the dimensions of the function spaces are given as

$$\begin{aligned} N_p &:= \dim\left(\mathbb{P}_p(E)\right) = \frac{(p+1)(p+2)(p+3)}{6}, \\ N_p^f &:= \dim\left(\mathbb{P}_p(F)\right) = \frac{(p+1)(p+2)}{2}. \end{aligned} \tag{12}$$

For polyhedron elements ($d = 3$) we introduce scaled monomials $\mathcal{M}_p(E)$ as

$$\mathcal{M}_p(E) := \left\{ m_\alpha = \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \text{ for } \boldsymbol{\alpha} \in \mathbb{N}^d \text{ with } |\boldsymbol{\alpha}| \leq p \right\}, \tag{13}$$

where the dimension of the function space is N_p, and for multi-index $\boldsymbol{\alpha} \in \mathbb{N}^d$ we denote the usual notation

$$|\boldsymbol{\alpha}| := \alpha_1 + \cdots + \alpha_d, \quad \boldsymbol{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} \cdots x_1^{\alpha_d}. \tag{14}$$

Besides, we also define the set of scaled monomials $\mathcal{M}_p^*(E)$ as

$$\mathcal{M}_p^*(E) := \left\{ m_\alpha = \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E}\right)^{\boldsymbol{\alpha}} \text{ for } \boldsymbol{\alpha} \in \mathbb{N}^d \text{ with } |\boldsymbol{\alpha}| = p \right\}. \tag{15}$$

With the help of the local coordinates (as shown in Fig. 1), similarly scaled monomials $\mathcal{M}_p^f(F)$ and $\mathcal{M}_p^{f,*}(F)$ for the polygon face F of a polyhedron can also be defined as

$$\mathcal{M}_p^f(F) := \left\{ m_\alpha^f = \left(\frac{\xi - \xi_F}{h_F} \right)^{\boldsymbol{\alpha}} \text{ for } \boldsymbol{\alpha} \in \mathbb{N}^{d-1} \text{ with } |\boldsymbol{\alpha}| \le p \right\}, \tag{16}$$

and

$$\mathcal{M}_p^{f,*}(F) := \left\{ m_\alpha^f = \left(\frac{\xi - \xi_F}{h_F} \right)^{\boldsymbol{\alpha}} \text{ for } \boldsymbol{\alpha} \in \mathbb{N}^{d-1} \text{ with } |\boldsymbol{\alpha}| = p \right\}. \tag{17}$$

3.2. Virtual element spaces on polygonal faces

As shown in Fig. 1, each polyhedral face $F \in \partial E$ is a two-dimensional planar polygon embedded in $\mathbb{R}^3$, which can be treated as a two-dimensional polygon using the local coordinates. As given in [1], the virtual element spaces for two-dimensional problems are defined as

$$\mathcal{V}_k(F) := \left\{ u_h \in \mathcal{H}^1(F) \cap C^0(F) : \Delta_F u_h \in \mathbb{P}_{k-2}(F), u_h|_e \in \mathbb{P}_k(e), e \in \partial F \right\}. \tag{18}$$

The first-order element is considered in this work so that we have $k = 1$ and the scaled monomials $m_\alpha^f \in \mathcal{M}_k^f(F)$ have the form as

$$m_1^f = 1, \quad m_2^f = \frac{\xi - \xi_F}{h_F} \quad m_3^f = \frac{\eta - \eta_F}{h_F}. \tag{19}$$

To ensure the computability of the L_2 projector $\Pi_{r,F}^0$ $(r \ge 0)$ on polygonal faces, the lifting virtual element space should be considered as

$$\tilde{\mathcal{V}}_{1,r}(F) := \left\{ u_h \in \mathcal{H}^1(F) \cap C^0(F) : \Delta_F u_h \in \mathbb{P}_r(F), u_h|_e \in \mathbb{P}_1(e), e \in \partial F \right\}. \tag{20}$$

To avoid additional degrees of freedom, we can follow the technique mentioned by Ahmad in [47] to modify the lifting space to a local enhancement space (always for $r \ge 0$)

$$\mathcal{W}_{1,r}(F) := \left\{ w_h \in \tilde{\mathcal{V}}_{1,r}(F) : \int_F \Pi_{1,F}^\nabla u_h m^f \, \mathrm{d}\Omega_F = \int_F u_h m^f \, \mathrm{d}\Omega_F, \forall m^f \in \mathcal{M}_s^{f,*}, 0 \le s \le r \right\}, \tag{21}$$

where $\Pi_{1,F}^\nabla : \tilde{\mathcal{V}}_{1,r}(F) \to \mathbb{P}_1(F)$ is the elliptic projection operator.

By using integrals by parts and the Green formula, the elliptic projection operator $\Pi_{k,F}^\nabla$ can be computed without using the extra degrees of freedom. ($\Pi_{k,F}^\nabla$ is only related to the boundary degrees of freedom). Since we can compute $\Pi_{k,F}^\nabla$ explicitly, the moments of order s with $k - 2 < s \le r$ are computable. But in fact, the maximum available accuracy is still k, which has nothing to do with s. As discussed in [36], if we use the method by adding additional degrees of freedom, the results is better, especially for nearly incompressible problems. In fact, for $k = 1$, the L_2 projection operator $\Pi_{1,F}^0$ coincides with $\Pi_{1,F}^\nabla$, see [47].

3.3. Virtual element spaces on polyhedral elements

For the polyhedral element E with boundary ∂E and $F \in \partial E$ and $k = 1$, we consider the virtual element space

$$\mathcal{V}_1(E) := \left\{ u_h \in \mathcal{H}^1(E) \cap C^0(E) : \Delta u_h \in \mathbb{P}_0(E), u_h|_e \in \mathcal{W}_{1,r}(F), F \in \partial E \right\}, \tag{22}$$

where the boundary spaces $\mathbb{B}(\partial E)$ are given by the virtual element spaces on polygonal faces $\mathcal{W}_{1,r}(F)$.

Similar to the virtual element spaces on polygonal faces, to ensure the computability of the L_2 projector $\Pi_{r,E}^0$ $(r \ge 0)$, we define the lifting virtual element space

$$\tilde{\mathcal{V}}_{1,r}(E) := \left\{ u_h \in \mathcal{H}^1(E) \cap C^0(E) : \Delta u_h \in \mathbb{P}_1(E), u_h|_e \in \mathcal{W}_{1,r}(F), F \in \partial E \right\}, \tag{23}$$

so that the local enhancement space is given by

$$\mathcal{W}_{1,r}(E) := \left\{ w_h \in \tilde{\mathcal{V}}_1(E) : \int_F \Pi_{1,E}^\nabla u_h \cdot m \, \mathrm{d}\Omega = \int_E u_h \cdot m \, \mathrm{d}\Omega, \forall m \in \mathcal{M}_s^*, 0 \le s \le r \right\}. \tag{24}$$

The L_2 projector $\Pi_{r,E}^0$ can be defined by

$$\int_E \Pi_{r,E}^0 u_h \cdot p \, \mathrm{d}\Omega = \int_E u_h \cdot p \, \mathrm{d}\Omega, \quad \forall u_h \in \tilde{\mathcal{V}}_{k,r}(E), p \in \mathbb{P}_r(E). \tag{25}$$

Similar to $\Pi_{1,F}^\nabla$, by using integration by parts and the Green formula, the elliptic projection operator $\Pi_{1,E}^\nabla$ can be computed in relation to the boundary degrees of freedom. We note that the L_2 projection operator $\Pi_{1,E}^0$ coincides with $\Pi_{1,E}^\nabla$, see [47].

The projection operators given above are all computable, and the corresponding traditional virtual element method format can be employed as given in [44–46]. However, the virtual element formulation requires the use of additional stabilization terms to ensure that the stiffness matrix has the correct rank. To obtain the stabilization-free virtual element method, the variable gradient needs to be approximated using high-order polynomials as well as the high-order L_2 projection operator. Next, we will show in detail how to calculate the high-order projection operator and how to use high-order polynomials to approximate the variable gradient to further construct the stabilization-free virtual element method.

4. Projection operators and numerical implementation

For the conventional VEM, the displacement ansatz contains polynomials of k, and the strain (gradient) is approximated by polynomials of order $k-1$. To construct the stabilization-free virtual element method, the gradient ($p > k-1$) is independent of the displacement model. In this section, we focus on the projection operators $\Pi^{\nabla}_{k,E}$ mentioned above for the first-order three-dimensional virtual element method ($k=1$). Besides, to construct the stabilization-free virtual element method, a high-order L_2 projection operator $\Pi^{0}_{l,E}$ is introduced.

4.1. $\mathcal{H}_1$ Projection operator on polyhedral elements

The first-order element is considered in this work and the scaled monomials $m_\alpha \in \mathcal{M}_1(E)$ have the form

$$m_1 = 1, \quad m_2 = \frac{x - x_E}{h_E}, \quad m_3 = \frac{y - y_E}{h_E}, \quad m_4 = \frac{z - z_E}{h_E}. \tag{26}$$

As introduced in Eq. (22), the degrees of freedom are selected as the values of u_h at the vertices for $k=1$ and the dimension of $\mathcal{V}_1(E)$ can be obtained as

$$N_E := \dim(\mathcal{V}_1(E)) = n_E, \tag{27}$$

where n_E is the number of vertices. As discussed in [44], the projector $\Pi^{\nabla}_{1,E} : \mathcal{V}_1(E) \to \mathbb{P}_1(E)$ can be calculated by

$$\int_E \nabla \Pi^{\nabla}_{1,E} u_h \cdot \nabla p \, \mathrm{d}\Omega = \int_E \nabla u_h \cdot \nabla p \, \mathrm{d}\Omega, \forall p \in \mathbb{P}_1(E) \tag{28}$$

with the additional condition

$$P_0 \left(\Pi^{\nabla}_{1,E} u_h - u_h \right) = 0, \tag{29}$$

where

$$P_0(u_h) := \frac{1}{n_E} \sum_{i=1}^{n_E} u_h \quad \text{for} \quad k = 1. \tag{30}$$

Considering the Green formula for Eq. (28), we have

$$\int_E \nabla \Pi^{\nabla}_{1,E} u_h \cdot \nabla p \ \mathrm{d}\Omega = -\int_E u_h \cdot \Delta p \ \mathrm{d}\Omega + \int_{\partial E} u_h \cdot \frac{\partial p}{\partial \boldsymbol{n}} \, \mathrm{d}\Gamma, \tag{31}$$

where the projection operator $\Pi^{\nabla}_{1,E}$ can be expanded in different bases as

$$\Pi^{\nabla}_{1,E} \phi_i = \sum_{\alpha=1}^{N_P} a_{\alpha,i} m_\alpha = \sum_{j=1}^{N_E} s_{j,i} \phi_j \quad \forall m_\alpha \in \mathcal{M}_1, \forall \phi_j \in \mathcal{V}_1, \tag{32}$$

with the following matrix form

$$\Pi^{\nabla}_{1,E} \boldsymbol{\phi}^T = \boldsymbol{m}^T \boldsymbol{\Pi}^{\nabla}_{1*,E} = \boldsymbol{\phi}^T \boldsymbol{\Pi}^{\nabla}_{1,E}, \tag{33}$$

where $N_P := \dim(\mathcal{M}_1) = 4$ in the current case (see Eq. (12) N_p). Besides, $\boldsymbol{\phi} = [\phi_1, \dots, \phi_{N_E}]^T$ is the unknown shape function vector with ϕ_i associated with the vertex i of E. The size of $\boldsymbol{\Pi}^{\nabla}_{1*,E}$ is $N_P \times N_E$ and the size of $\boldsymbol{\Pi}^{\nabla}_{1,E}$ is $N_E \times N_E$. The relation in Eq. (33) can be combined by defining the transition matrix $\boldsymbol{D}$ as $\boldsymbol{D}_{j,\alpha} = \mathrm{dof}_j(m_\alpha)$ such that

$$\boldsymbol{m}^T = \boldsymbol{\phi}^T \boldsymbol{D}, \quad \boldsymbol{\Pi}^{\nabla}_{1,E} = \boldsymbol{D} \boldsymbol{\Pi}^{\nabla}_{1*,E}. \tag{34}$$

Substituting Eqs. (34) and (32) into Eq. (31), yields

$$\int_E \nabla \boldsymbol{m} \cdot \nabla \boldsymbol{m}^T \, \mathrm{d}\Omega \boldsymbol{\Pi}^{\nabla}_{1*,E} = -\int_E \Delta \boldsymbol{m} \cdot \boldsymbol{\phi}^T \ \mathrm{d}\Omega + \sum_{F \subset \partial E} \int_F \left(\nabla \boldsymbol{m} \cdot \boldsymbol{n}_F \right) \boldsymbol{\phi}^T \, \mathrm{d}\Gamma, \tag{35}$$

and then leads to the matrix form

$$\boldsymbol{G}^{\nabla} \boldsymbol{\Pi}^{\nabla}_{1*,E} = \boldsymbol{B}^{\nabla}, \tag{36}$$

where

$$\boldsymbol{G}^{\nabla} := \int_E \nabla \boldsymbol{m} \cdot \nabla \boldsymbol{m}^T \, \mathrm{d}\Omega, \quad \boldsymbol{B}^{\nabla} := \sum_{F \subset \partial E} \int_F \left(\nabla \boldsymbol{m} \cdot \boldsymbol{n}_F \right) \boldsymbol{\phi}^T \, \mathrm{d}\Gamma. \tag{37}$$

The size of $\boldsymbol{G}^{\nabla}$ is $N_P \times N_P$ and the size of $\boldsymbol{B}^{\nabla}$ is $N_P \times N_E$. Considering the transition matrix $\boldsymbol{D}$, it can be proved that the $\boldsymbol{G}^{\nabla}$ matrix can be calculated as

$$\boldsymbol{G}^{\nabla} = \boldsymbol{B}^{\nabla} \boldsymbol{D}. \tag{38}$$

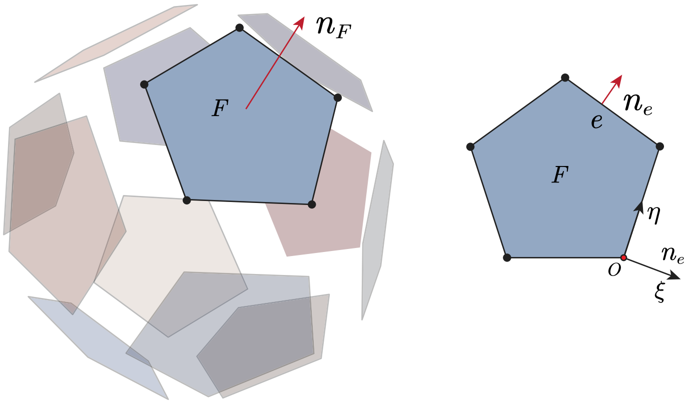


Fig. 2. A polyhedral element and its polygonal surfaces.

The computationally most involved step is to calculate the matrix $\boldsymbol{B}^{\nabla}$ since the basis function $\boldsymbol{\phi}$ is unknown at the boundary (the boundaries of a polyhedron are polygons). Due to the definitions of $\mathcal{V}_1(E)$ (see Eq. (22)) and $\mathcal{V}_1(F)$ (see Eq. (18)), the matrix $\boldsymbol{B}^{\nabla}$ can be obtained by

$$\begin{aligned}\boldsymbol{B}^{\nabla} &= \sum_{F\subset\partial E}\int_F \left(\nabla\boldsymbol{m}\cdot\boldsymbol{n}_F\right)\boldsymbol{\phi}^T\,\mathrm{d}\Gamma \\ &= \sum_{F\subset\partial E}\left[\left(\nabla\boldsymbol{m}\cdot\boldsymbol{n}_F\right)\int_F \boldsymbol{\phi}_F^T\boldsymbol{\Pi}^{\nabla}_{1,F}\,\mathrm{d}\Gamma\right] = \sum_{F\subset\partial E}\left[\left(\nabla\boldsymbol{m}\cdot\boldsymbol{n}_F\right)\int_F \left(\boldsymbol{m}^f\right)^T\,\mathrm{d}\Gamma\cdot\boldsymbol{\Pi}^{\nabla}_{1*,F}\right],\end{aligned} \tag{39}$$

where $\boldsymbol{m}^f$ is used to refer to an element of $\mathcal{M}_1^f(F)$, $\boldsymbol{n}_F$ is the normal of the polygon face F, see Fig. 2, $\Pi^{\nabla}_{1*,F}$ is the Ritz projection operator (see projection operator $\Pi^{\nabla}_{1,F}$). Given the local coordinates, each polyhedral face $F\in\partial E$ is a two-dimensional planar polygon, see Fig. 2. Similar to the two-dimensional problem, the projection operator $\Pi^{\nabla}_{1*,F}$ and its matrix form $\boldsymbol{\Pi}^{\nabla}_{1*,F}$ can be computed. The calculation process can be found in the relevant two-dimensional papers on VEM [1,40,48]. For $k=1$, the following simpler formula holds

$$\int_F \boldsymbol{\phi}_F^T\boldsymbol{\Pi}^{\nabla}_{1,F}\,\mathrm{d}\Gamma = \int_F\left(\boldsymbol{m}^f\right)^T\,\mathrm{d}\Gamma\cdot\boldsymbol{\Pi}^{\nabla}_{1*,F} = [|F|,0,0]\cdot\boldsymbol{\Pi}^{\nabla}_{1*,f}, \tag{40}$$

where $|F|$ is the area of F.

Substituting Eq. (40) into Eq. (39), yields the matrix $\boldsymbol{B}^{\nabla}$ and the Ritz projection matrix $\boldsymbol{\Pi}^{\nabla}_{1*,E}$ can be solved

$$\boldsymbol{\Pi}^{\nabla}_{1*,E} = \left(\boldsymbol{G}^{\nabla}\right)^{-1}\boldsymbol{B}^{\nabla}. \tag{41}$$

It should be mentioned that the condition (29) has been considered in Eq. (41) so that matrix $\boldsymbol{G}^{\nabla}$ is invertible. Note that it is not necessary to calculate matrix $\boldsymbol{G}^{\nabla}$ since it is given by $\boldsymbol{G}^{\nabla}=\boldsymbol{B}^{\nabla}\boldsymbol{D}$.

The matrix $\boldsymbol{\Pi}^{\nabla}_{1,E}$ as well as $\boldsymbol{\Pi}^{\nabla}_{1*,E}$ are two important projection matrices used in the traditional VEM. It is precisely because of the use of projection matrices that traditional VEM requires additional stabilization terms. In the next, we will provide a higher-order polynomial L_2 projection for the gradient to ensure that the stiffness matrix has the correct rank. To avoid the introduction of additional degrees of freedom, the projection matrices $\boldsymbol{\Pi}^{\nabla}_{1*,E}$ and $\boldsymbol{\Pi}^{\nabla}_{1,E}$ will be used. This is why we spend some effort introducing the calculation process of these projection matrices.

4.2. L_2 Projection operator on polyhedral elements

Similar to the work [38,39], the gradient is independent of the displacement ansatz and a higher-order L_2 projection operator is introduced to ensure that the strain and stress can represent the element energy accurately. Different from [38,39] where the elastic energy enters the projection, the L_2 projection is designed for the gradient in this work. Similar to the L_2 projection operator $\Pi^0_{l,E}$ defined in Eq. (25), we define the L_2 projection operator $\boldsymbol{\Pi}^0_{l,E}\nabla$ of the gradient of the variable field, which is

$$\boldsymbol{\Pi}^0_{l,E}\nabla : \mathcal{H}^1(E)\rightarrow\left[\mathbb{P}_l(E)\right]^3, \tag{42}$$

where $l\in\mathbb{N}$ is a parameter defining the order of the polynomial space and depending on the number of vertices n_E. The selection of parameter l to establish a well-posed and stable virtual element formulation is an ongoing concern that will be addressed in Section 6.

To define our bilinear form, the L_2 projection operator of the gradient of function in $\mathcal{H}^1(E)$ can be defined by the orthogonality condition

$$\left(\boldsymbol{\Pi}^0_{l,E}\nabla u,\boldsymbol{p}\right)_E = (\nabla u,\boldsymbol{p})_E \quad \forall\boldsymbol{p}\in\left[\mathbb{P}_l(E)\right]^3, \tag{43}$$

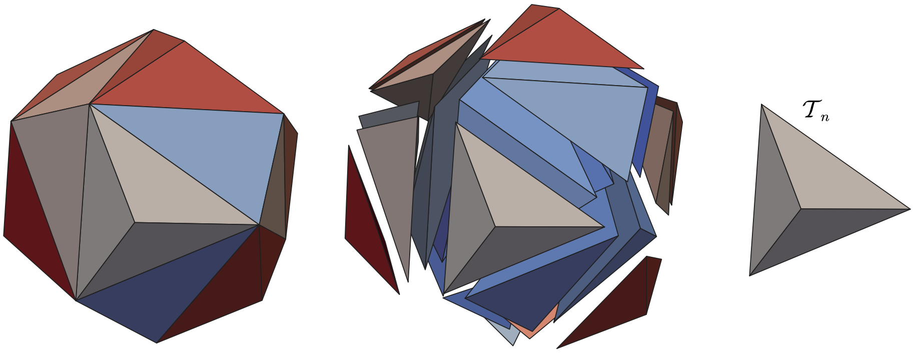


Fig. 3. Polyhedral triangulation and numerical integration.

or

$$\int_E \boldsymbol{p}^T \boldsymbol{\Pi}_{l,E}^0 \nabla u_h \,\mathrm{d}\Omega = \int_E \boldsymbol{p}^T \nabla u_h \,\mathrm{d}\Omega, \tag{44}$$

where the right hand can be written as

$$\int_E \boldsymbol{p}^T \nabla u_h \,\mathrm{d}\Omega = \int_{\partial E} \left(\boldsymbol{p}^T \cdot \boldsymbol{n}_F\right) u_h \,\mathrm{d}\Gamma - \int_E (\mathrm{div}\boldsymbol{p})\, u_h \,\mathrm{d}\Omega. \tag{45}$$

By representing the virtual variable field u_h, gradient ∇u_h, and polynomial $\boldsymbol{p}$ in terms of the basis function, we have

$$u_h = \boldsymbol{\phi}^T \tilde{\boldsymbol{u}}, \quad \nabla u_h = \left(\boldsymbol{N}^p\right)^T \tilde{\epsilon}, \quad \boldsymbol{p} = \left(\boldsymbol{N}^p\right)^T \tilde{\boldsymbol{p}}, \tag{46}$$

where $\tilde{\square}$ represents a vector, $\epsilon := \nabla u_h$, $\boldsymbol{\phi}$ is the basis function in $\mathcal{V}_1(E)$ and $\boldsymbol{N^p}$ is a matrix which contains the polynomial basis

$$\left(\boldsymbol{N}^p\right)^T := \begin{bmatrix} \boldsymbol{m}_l^T & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{m}_l^T & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{m}_l^T \end{bmatrix}, \tag{47}$$

where $\boldsymbol{m}_l$ is the complete polynomial basis function of order l. Besides, we can use $\boldsymbol{\Pi}^m$ to represent the matrix form of the operator $\boldsymbol{\Pi}_{l,E}^0 \nabla$ and the projected gradient yields

$$\boldsymbol{\Pi}_{l,E}^0 \nabla u_h = \left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m \tilde{\boldsymbol{u}}. \tag{48}$$

Substituting Eqs. (46) and (48) into Eq. (44) and considering Eq. (45), we have

$$\tilde{\boldsymbol{p}}^T \int_E \boldsymbol{N}^p \left(\boldsymbol{N}^p\right)^T \,\mathrm{d}\Omega \boldsymbol{\Pi}^m \tilde{\boldsymbol{u}} = \tilde{\boldsymbol{p}}^T \int_{\partial E} \left(\boldsymbol{N}^p \cdot \boldsymbol{n}_F\right) \boldsymbol{\phi}^T \,\mathrm{d}\Gamma \tilde{\boldsymbol{u}} - \tilde{\boldsymbol{p}}^T \int_E \left(\mathrm{div}\boldsymbol{N}^p\right) \boldsymbol{\phi}^T \,\mathrm{d}\Omega \tilde{\boldsymbol{u}}. \tag{49}$$

Since this is true for all $\tilde{\boldsymbol{u}}$ and $\hat{\boldsymbol{p}}$, we can rewrite the equation as

$$\int_E \boldsymbol{N}^p \left(\boldsymbol{N}^p\right)^T \,\mathrm{d}\Omega \boldsymbol{\Pi}^m = \int_{\partial E} \left(\boldsymbol{N}^p \cdot \boldsymbol{n}_F\right) \boldsymbol{\phi}^T \,\mathrm{d}\Gamma - \int_E \left(\mathrm{div}\boldsymbol{N}^p\right) \boldsymbol{\phi}^T \,\mathrm{d}\Omega, \tag{50}$$

which then leads to the matrix form

$$\boldsymbol{G}\boldsymbol{\Pi}^m = \boldsymbol{B}, \tag{51}$$

where

$$\boldsymbol{G} := \int_E \boldsymbol{N}^p \left(\boldsymbol{N}^p\right)^T \,\mathrm{d}\Omega, \tag{52}$$

$$\boldsymbol{B} := \int_{\partial E} \left(\boldsymbol{N}^p \cdot \boldsymbol{n}_F\right) \boldsymbol{\phi}^T \,\mathrm{d}\Gamma - \int_E \left(\mathrm{div}\boldsymbol{N}^p\right) \boldsymbol{\phi}^T \,\mathrm{d}\Omega. \tag{53}$$

It is necessary to calculate the matrix $\boldsymbol{G}$ and matrix $\boldsymbol{B}$ in order to solve Eq. (51) for the L_2 projection matrix $\boldsymbol{\Pi}^m$. Then the approximate expression of the gradient follows from Eq. (48). Matrix $\boldsymbol{G}$ can be calculated by partitioning E into a set of tetrahedrons $\mathcal{T}_n$ (as shown in Fig. 3) and adopting Gauss integration on the tetrahedron. Of course, we can also use the divergence theorem to transform the domain integral onto the polygon surface, and then further transform it onto the edge to simplify the calculation, see e.g. [49]. It is undeniable that this will complicate the integration and increase the order of the integral. But it will also improve the adaptability to complex polyhedra and provide an analytical result for the integral.

The calculation of matrix $\boldsymbol{B}$ is similar to matrix $\boldsymbol{B}^\nabla$ mentioned in Eq. (39). At present work, we use

$$\boldsymbol{B} = \boldsymbol{I}_1 - \boldsymbol{I}_2. \tag{54}$$

Considering the boundary space defined in Eq. (22) ($s = l$), the first term in Eq. (53) follows

$$\begin{aligned}\boldsymbol{I}_1 &:= \sum_{F\subset\partial E}\int_e \left(\boldsymbol{N}^p\cdot\boldsymbol{n}_F\right)\boldsymbol{\phi}^T\,\mathrm{d}\Gamma \\ &= \sum_{F\subset\partial E}\int_e \left(\boldsymbol{N}^p\cdot\boldsymbol{n}_F\right)\Pi^{\nabla}_{1,f}\boldsymbol{\phi}^T\,\mathrm{d}\Gamma = \sum_{F\subset\partial E}\int_e \left(\boldsymbol{N}^p\cdot\boldsymbol{n}_F\right)\boldsymbol{m}_f^T\,\mathrm{d}\Gamma\,\boldsymbol{\Pi}^{\nabla}_{1*,f},\end{aligned} \tag{55}$$

where $\Pi^{\nabla}_{1*,F}$ is the Ritz projection operator (with matrix form $\boldsymbol{\Pi}^{\nabla}_{1*,f}$). The integral of Eq. (55) can be calculated by partitioning $F\subset\partial E$ into a set of triangles and adopting a Gauss quadrature rule.

For the second term in Eq. (53), we have

$$\boldsymbol{I}_2 := \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\boldsymbol{\phi}^T\,\mathrm{d}\Omega, \tag{56}$$

where the basis function $\boldsymbol{\phi}$ is unknown in the polyhedral E. To calculate the matrix $\boldsymbol{I}_2$, we define $\Pi^0_{l-1,E}$ as the standard scalar $L_2(E)$ projection from $\tilde{\mathcal{V}}_{k,l-1}(E)$ to $\mathbb{P}_{l-1}$

$$\left(\Pi^0_{l-1,E}v_h, p\right)_E = \left(v_h, p\right)_E \quad \forall v_h\in\tilde{\mathcal{V}}_{k,l-1}(E), p\in\mathbb{P}_{l-1}(E). \tag{57}$$

Considering the L_2 projection defined in Eq. (25) and the local enhancement space defined in Eq. (24), we can directly replace the L_2 projection $\Pi^0_{l-1,E}$ with the $\mathcal{H}_1$ projection $\Pi^{\nabla}_{1*,E}$ as

$$\begin{aligned}\boldsymbol{I}_2 &= \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\boldsymbol{\phi}^T\,\mathrm{d}\Omega = \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\left(\Pi^{\nabla}_{1,E}\boldsymbol{\phi}^T\right)\mathrm{d}\Omega \\ &= \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\boldsymbol{\phi}^T\boldsymbol{\Pi}^{\nabla}_{1,E}\,\mathrm{d}\Omega = \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\boldsymbol{m}_1^T\boldsymbol{\Pi}^{\nabla}_{1*,E}\,\mathrm{d}\Omega \\ &= \int_E \left(\mathrm{div}\boldsymbol{N}^p\right)\boldsymbol{m}_1^T\,\mathrm{d}\Omega\,\boldsymbol{\Pi}^{\nabla}_{1*,E}.\end{aligned} \tag{58}$$

Substituting matrices $\boldsymbol{G}$ and $\boldsymbol{B}$ into Eq. (36), the projection matrix $\boldsymbol{\Pi}^m$ is obtained

$$\boldsymbol{\Pi}^m = \boldsymbol{G}^{-1}\boldsymbol{B}. \tag{59}$$

5. Stabilization-free VEM for 3D linear elasticity

As discussed in the previous section, we have calculated the L_2 projection operator for the gradient of a scalar field u, which can be approximated as (see Eq. (48))

$$\nabla u \approx \boldsymbol{\Pi}^0_{l,E}\nabla u_h = \left(\boldsymbol{N}^p\right)^T\boldsymbol{\Pi}^m\tilde{\boldsymbol{u}}. \tag{60}$$

For the elastic problems, we can use this approximation directly to calculate the strain in the Eq. (4). Of course, we can also calculate the L_2 projection operator for the gradient of a vector field, but it can be found that the matrix formulation increases in dimensionality and results in additional computational costs.

On using Voigt (engineering) notation, we can write the stress and strain in terms of 6 × 1 arrays:

$$\hat{\boldsymbol{\sigma}} = \left\{\sigma_{11} \quad \sigma_{22} \quad \sigma_{33} \quad \sigma_{12} \quad \sigma_{23} \quad \sigma_{31}\right\}^T, \tag{61}$$

$$\hat{\boldsymbol{\varepsilon}} = \left\{\varepsilon_{11} \quad \varepsilon_{22} \quad \varepsilon_{33} \quad 2\varepsilon_{12} \quad 2\varepsilon_{23} \quad 2\varepsilon_{31}\right\}^T. \tag{62}$$

Based on the definition of the strain, considering Eq. (60), we write

$$\begin{aligned}\hat{\boldsymbol{\varepsilon}} = \begin{Bmatrix}\varepsilon_{11}\\ \varepsilon_{22}\\ \varepsilon_{33}\\ 2\varepsilon_{12}\\ 2\varepsilon_{23}\\ 2\varepsilon_{31}\end{Bmatrix} &= \begin{bmatrix}1&0&0\\0&0&0\\0&0&0\\0&1&0\\0&0&0\\0&0&1\end{bmatrix}\begin{Bmatrix}\frac{\partial u}{\partial x}\\ \frac{\partial u}{\partial y}\\ \frac{\partial u}{\partial z}\end{Bmatrix} + \begin{bmatrix}0&0&0\\0&1&0\\0&0&0\\1&0&0\\0&0&1\\0&0&0\end{bmatrix}\begin{Bmatrix}\frac{\partial v}{\partial x}\\ \frac{\partial v}{\partial y}\\ \frac{\partial v}{\partial z}\end{Bmatrix} + \begin{bmatrix}0&0&0\\0&0&0\\0&0&1\\0&0&0\\0&1&0\\1&0&0\end{bmatrix}\begin{Bmatrix}\frac{\partial w}{\partial x}\\ \frac{\partial w}{\partial y}\\ \frac{\partial w}{\partial z}\end{Bmatrix} \\ &= \left[\boldsymbol{A}_1\left(\boldsymbol{N}^p\right)^T\boldsymbol{\Pi}^m \quad \boldsymbol{A}_2\left(\boldsymbol{N}^p\right)^T\boldsymbol{\Pi}^m \quad \boldsymbol{A}_3\left(\boldsymbol{N}^p\right)^T\boldsymbol{\Pi}^m\right]\begin{Bmatrix}\tilde{\boldsymbol{u}}\\ \tilde{\boldsymbol{v}}\\ \tilde{\boldsymbol{w}}\end{Bmatrix},\end{aligned} \tag{63}$$

which can be combined as

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{A}\boldsymbol{N}_p^T\boldsymbol{\Pi}_m\tilde{\boldsymbol{U}}, \tag{64}$$

with

$$\boldsymbol{N}_p^T = \begin{bmatrix} (\boldsymbol{N}^p)^T & & \\ & (\boldsymbol{N}^p)^T & \\ & & (\boldsymbol{N}^p)^T \end{bmatrix}, \quad \boldsymbol{\Pi}_m = \begin{bmatrix} \boldsymbol{\Pi}^m & & \\ & \boldsymbol{\Pi}^m & \\ & & \boldsymbol{\Pi}^m \end{bmatrix}. \tag{65}$$

$$\tilde{\boldsymbol{U}} = \begin{bmatrix} \tilde{\boldsymbol{u}} & \tilde{\boldsymbol{v}} & \tilde{\boldsymbol{w}} \end{bmatrix}^T, \tag{66}$$

and

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 & \boldsymbol{A}_3 \end{bmatrix}. \tag{67}$$

The element stiffness matrix follows from the bilinear form a_h^E as

$$\begin{aligned} a_h^E \left(\boldsymbol{u}_h, \boldsymbol{v}_h\right) &= \int_E \boldsymbol{\varepsilon}(\boldsymbol{u}_h)^T \hat{\boldsymbol{C}} \boldsymbol{\varepsilon}(\boldsymbol{v}_h)\, \mathrm{d}\Omega \\ &= \int_E \tilde{\boldsymbol{U}}^T \boldsymbol{\Pi}_m^T \boldsymbol{N}_p \boldsymbol{A}^T \hat{\boldsymbol{C}} \boldsymbol{A} \boldsymbol{N}_p^T \boldsymbol{\Pi}_m \tilde{\boldsymbol{V}}\, \mathrm{d}\Omega \\ &= \tilde{\boldsymbol{U}}^T \boldsymbol{\Pi}_m^T \int_E \boldsymbol{N}_p \boldsymbol{A}^T \hat{\boldsymbol{C}} \boldsymbol{A} \boldsymbol{N}_p^T\, \mathrm{d}\Omega\, \boldsymbol{\Pi}_m \tilde{\boldsymbol{V}}, \end{aligned} \tag{68}$$

where $\hat{\boldsymbol{C}}$ is the associated matrix representation of the fourth-order constitutive tensor

$$\hat{\boldsymbol{C}} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & 0 & 0 & 0 \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix}. \tag{69}$$

with E being Young's modulus and ν the Poisson's ratio of the material.

Following Eq. (68) the element stiffness matrix has the form

$$\boldsymbol{K}_E = \boldsymbol{\Pi}_m^T \int_E \boldsymbol{N}_p \boldsymbol{A}^T \hat{\boldsymbol{C}} \boldsymbol{A} \boldsymbol{N}_p^T\, \mathrm{d}\Omega \boldsymbol{\Pi}_m = \boldsymbol{\Pi}_m^T \boldsymbol{G}_0 \boldsymbol{\Pi}_m, \tag{70}$$

where

$$\boldsymbol{G}_0 = \int_E \boldsymbol{N}_p \boldsymbol{A}^T \hat{\boldsymbol{C}} \boldsymbol{A} \boldsymbol{N}_p^T\, \mathrm{d}\Omega \tag{71}$$

can be calculated by partitioning E into tetrahedrons and adopting a Gauss quadrature rule. Lastly, the system of equations can be written as

$$\boldsymbol{K}\tilde{\boldsymbol{U}} = \boldsymbol{F}, \tag{72}$$

where $\boldsymbol{F}$ is the load vector. Besides, strain and stress should be calculated at Gauss integration points, e.g.

$$\hat{\boldsymbol{\varepsilon}} = \boldsymbol{A}\boldsymbol{N}_p \boldsymbol{\Pi}_m \tilde{\boldsymbol{U}}, \quad \hat{\boldsymbol{\sigma}} = \hat{\boldsymbol{C}} \hat{\boldsymbol{\varepsilon}}. \tag{73}$$

In order to explain more clearly how to use this method to calculate the element stiffness matrix, a specific flow chart is given in Fig. 4. As can be seen from the figure, compared with traditional VEM, SFVEM requires an additional calculation of the L2 projection operator (step 2) and the stiffness matrix (step 3). Since SFVEM involves the domain integral of high-order polynomials, the polyhedral element needs to be divided into multiple tetrahedrons and calculated using Gaussian integrals, so the second step will be very time-consuming. In the third step, similar technique can be used to calculate the integral, but the matrix $\boldsymbol{G}$ can also be obtained indirectly from the matrix $\boldsymbol{G}_0$, so it is not very time-consuming.

6. Numerical examples

In this section, several typical three-dimensional elastic mechanical examples will be provided to prove the correctness and accuracy of the stabilization-free virtual element method. All computations are performed with self-written Matlab codes. Before this, it is necessary to discuss the polynomial order l used for variable gradient approximation and to conduct convergence analysis.

6.1. Choice of the polynomial order l

As mentioned in the previous section, $l \in \mathbb{N}$ is a parameter that defines the order of the polynomial and depends on the number of vertices n_E. For two-dimensional problems, some relations between the polynomial order l and the number of vertices n_E of a

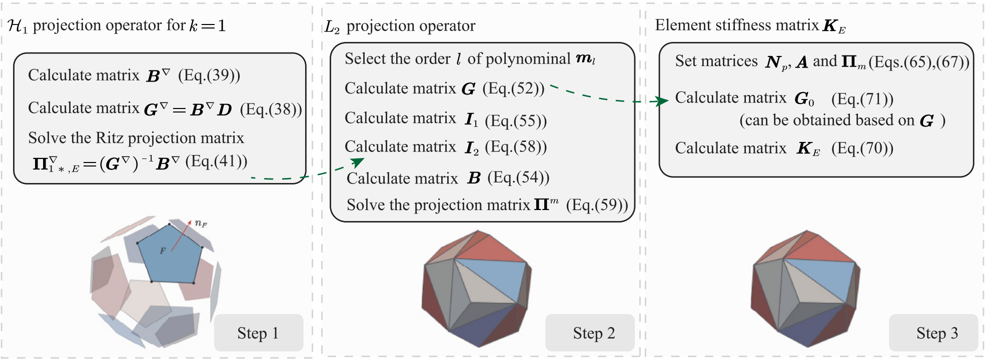

$\mathcal{H}_1$ projection operator for $k=1$
Calculate matrix $\boldsymbol{B}^{\nabla}$ (Eq.(39))
Calculate matrix $\boldsymbol{G}^{\nabla}=\boldsymbol{B}^{\nabla}\boldsymbol{D}$ (Eq.(38))
Solve the Ritz projection matrix
$\boldsymbol{\Pi}^{\nabla}_{1*,E}=(\boldsymbol{G}^{\nabla})^{-1}\boldsymbol{B}^{\nabla}$ (Eq.(41))
n_F
F
Step 1
L_2 projection operator
Select the order l of polynominal $\boldsymbol{m}_l$
Calculate matrix $\boldsymbol{G}$ (Eq.(52))
Calculate matrix $\boldsymbol{I}_1$ (Eq.(55))
Calculate matrix $\boldsymbol{I}_2$ (Eq.(58))
Calculate matrix $\boldsymbol{B}$ (Eq.(54))
Solve the projection matrix $\boldsymbol{\Pi}^m$ (Eq.(59))
Step 2
Element stiffness matrix $\boldsymbol{K}_E$
Set matrices $\boldsymbol{N}_p$, $\boldsymbol{A}$ and $\boldsymbol{\Pi}_m$ (Eqs.(65),(67))
Calculate matrix $\boldsymbol{G}_0$ (Eq.(71))
(can be obtained based on $\boldsymbol{G}$)
Calculate matrix $\boldsymbol{K}_E$ (Eq.(70))
Step 3

Fig. 4. Calculation flow chart of element stiffness matrix.

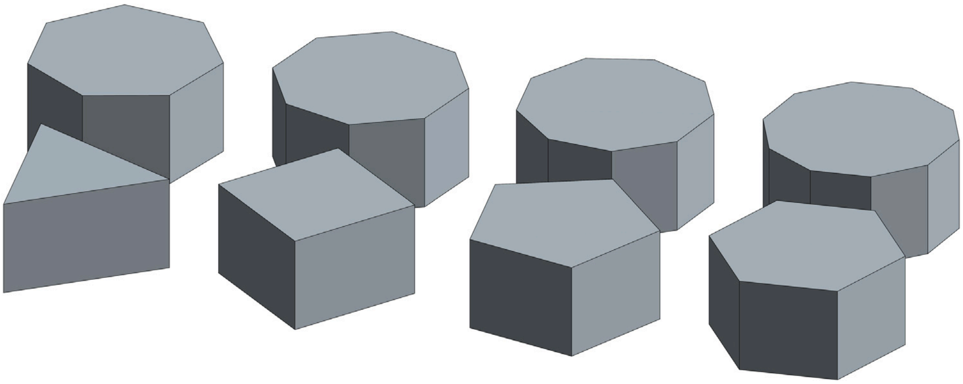


Fig. 5. Regular polyhedra used for the eigenvalue analysis.

virtual element and discussions to construct a formulation with correct rank can be found in [35,38,39,50]. Unfortunately, there are no relevant equations and proofs that help to select l for three-dimensional problems.

Basically, for a given number of nodes there is no rigorous proof that a projection onto polynomials of a given order guarantees the stability of a polygon (in 2D) or of a polyhedron (in 3D) for any possible shape of the polygon or polyhedron. Whatever is the element shape, there is always a minimum number of monomials in the polynomial used for the projection. If the number of monomials is too small, a simple algebraic count of the number of rows and columns of the projection operator shows that the element stiffness matrix will certainly not have the correct rank. As used in [38,39], an eigenvalue analysis was selected for plane elasticity to numerically establish a choice of l for two-dimensional problems. Similar to this work, we analyze the eigenvalue problem $\boldsymbol{K}_E\boldsymbol{d}_E = \lambda\boldsymbol{d}_E$ for the three-dimension virtual element, thereby providing the relationship between polynomial order l and the number of vertices n_E. In this example, we analyze the eigenvalue for polyhedra with different numbers of vertices n_E (shown in Figs. 5 and 6) and test the number of spurious modes for different orders of $l = 1, 2, 3$.

The polyhedra shown in Fig. 5 have an even number of vertices. To obtain a polyhedron with odd vertices, an additional point can be added at the bottom of the polyhedra as shown in Fig. 6. The resulting number of spurious eigenvalues as a function of the number of nodes for regular (symmetric) and irregular (asymmetric) elements are depicted in Figs. 7 and 8. We found that for irregular elements, with the same order l, more nodes can be used in the polyhedral elements. This means that for symmetric elements, there will be some spurious zero eigenvalues. It can be tested that for irregular elements, even if the polygonal surface is not triangulated and approximated directly using the two-dimensional Π^{∇}operator, the same eigenvalues as shown in Fig. 8 can be obtained.

In order to apply elements with more vertices, the order of l can be appropriately increased according to the descriptions in Figs. 7 and 8. For $l = 1$, the applicable number of nodes is $n_E \leq 10$ or $n_E \leq 13$ for symmetric and asymmetric elements, respectively. For $l = 2$, a 25-node asymmetric element has the correct rank. Indeed, the eigenvalue analysis would be valid only for the specific considered element shape and could not be reliably used for other elements, with the same number of nodes, but different shape. So the authors do not attempt to provide a corresponding relationship similar to [35,38,39,50]. Thus the following examples will automatically select the order of the polynomial based on the number of vertices and the results provided in Figs. 7 and 8.

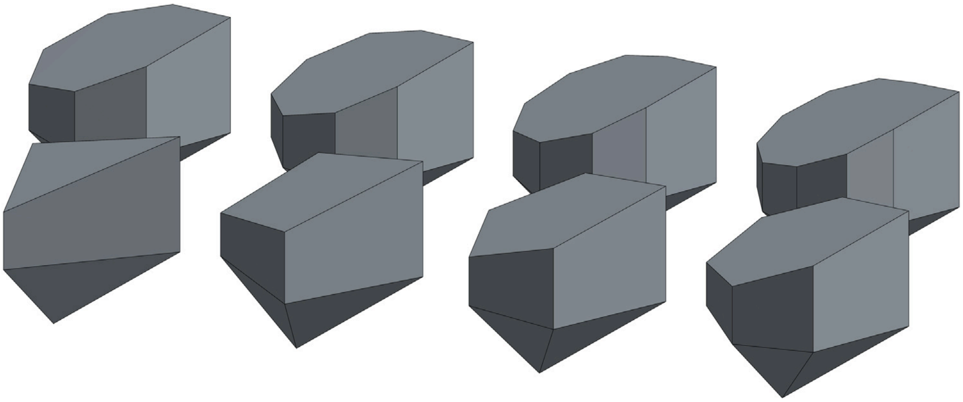


Fig. 6. Irregular polyhedra used for the eigenvalue analysis.

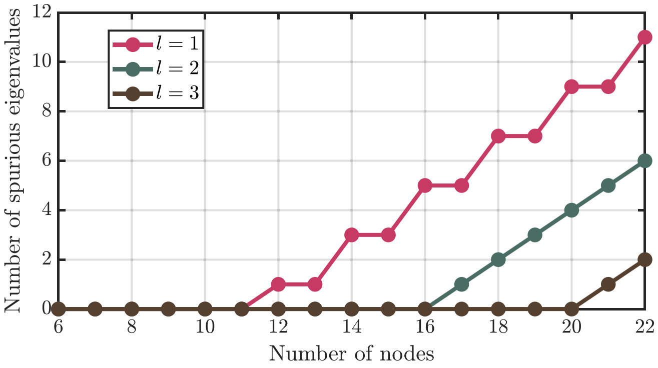


Fig. 7. Eigenvalue analysis on regular polyhedra for different order l.

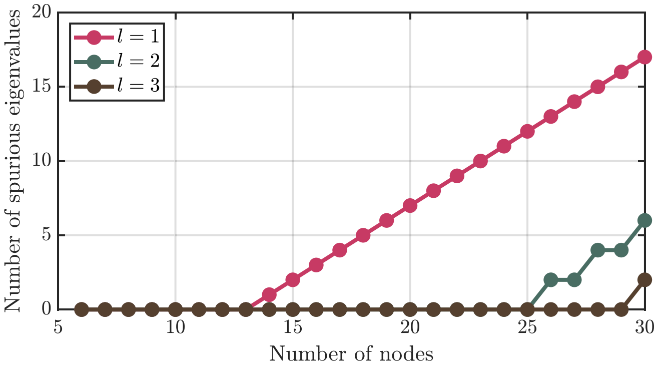


Fig. 8. Eigenvalue analysis on irregular polyhedra for different order l.

6.2. Convergence study

The convergence behavior of the new proposed three-dimensional SFVEM is studied for the Poisson equation. To test convergence, we examine the errors using the L^2 and H^1 norms. The discrete measures are given as

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2} = \sqrt{\sum_{E\in\mathcal{J}} \|\boldsymbol{u} - \boldsymbol{\Pi}^{\nabla}_{1,E}\boldsymbol{u}_h\|^2}, \tag{74}$$

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1} = \sqrt{\sum_{E\in\mathcal{J}} \|\nabla\boldsymbol{u} - \nabla\boldsymbol{\Pi}^{\nabla}_{1,E}\boldsymbol{u}_h\|^2}. \tag{75}$$

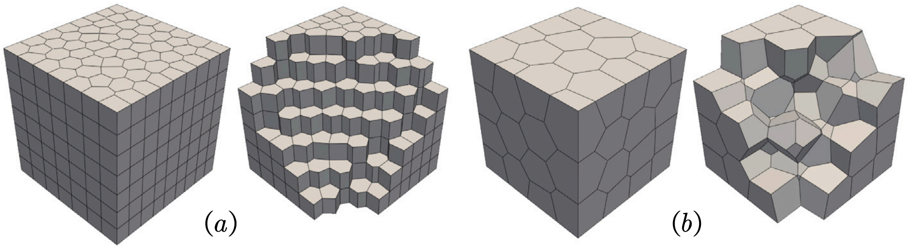


Fig. 9. Different discretization of three-dimensional problem, (a) 2D swept mesh and (b) CVT mesh.

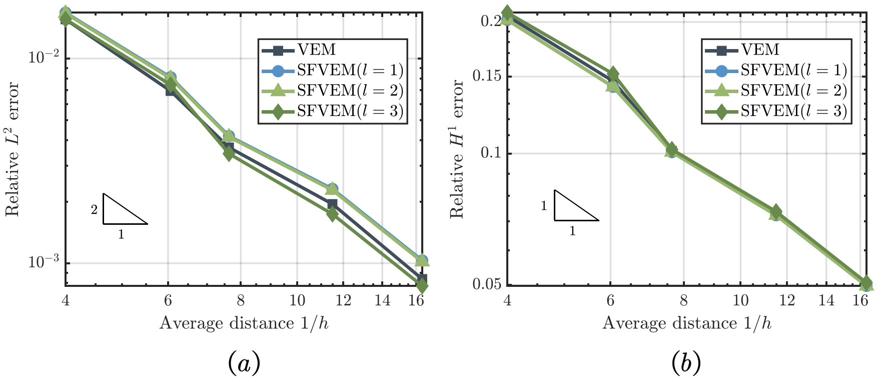


Fig. 10. Comparison of the convergence of the standard VEM and the SFVEM for 2D swept mesh (a) L^2 error and (b) H^1 error.

For a cube $\Omega = (0,1)^3$, the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + Q = 0, \tag{76}$$

is considered which has the analytical solution

$$u(x,y) = \sin(2xy)\cos(z). \tag{77}$$

A Neumann boundary condition is applied at face $x = 0$ and Dirichlet boundary conditions are assumed on the other boundaries based on the analytical solution. In this example, two different types of meshes are considered. These include a 2D swept mesh and a CVT (Centroidal Voronoi Tessellation) mesh as shown in Fig. 9. In fact, the elements in both meshes are what we mentioned earlier as irregular elements. Different parameters $l = 1, 2, 3$ are selected for the stabilization-free virtual element method. Besides, the conventional VEM is also used for comparison.

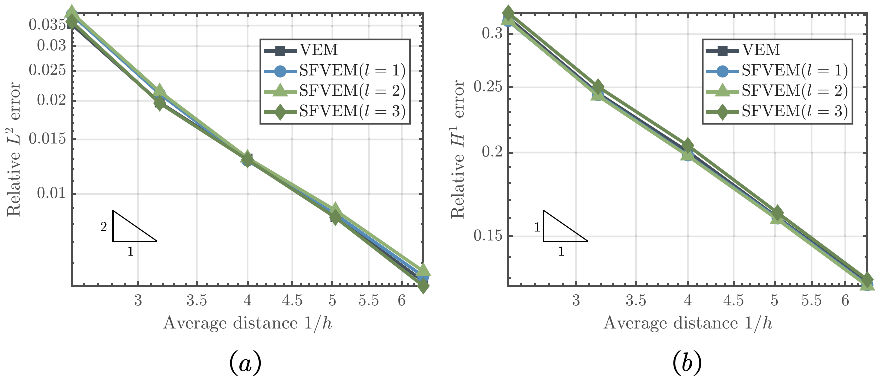
According to Eqs. (74) and (75), different errors can be calculated and the convergence curve is plotted in Figs. 10 and 11 for the 2D swept mesh and the CVT mesh, respectively. Easy to find that the solutions (accuracy, convergence rate) obtained by SFVEM are very close to the solutions obtained by conventional VEM. This shows that the proposed SFVEM is accurate and leads to similar results as provided by the traditional VEM. It should be noted that although some elements do not meet the requirements of the parameter l in Figs. 7 and 8 (especially for $l = 1$), it can be seen that the results are acceptable. This means that while an individual element is rank deficient the rank of the global assembled stiffness matrix is correct. Contour plots of the unknown u are given in Fig. 12.

6.3. 3D Cook's membrane problem

In this example, the well-known Cook's membrane problem is analyzed. The tapered beam is fixed on the left side and subjected to a constant distributed vertical load q_y. The geometric model is shown in Fig. 13. The relevant dimensions are $L = 48$, $H_1 = 44$, $H_2 = 16$, $B = 10$. The material parameters are selected as $E = 10$ and $\nu = 0.3$. The distributed vertical load is given as $q_y = 0.5$. Two different meshes are used for comparison as depicted in Fig. 13.

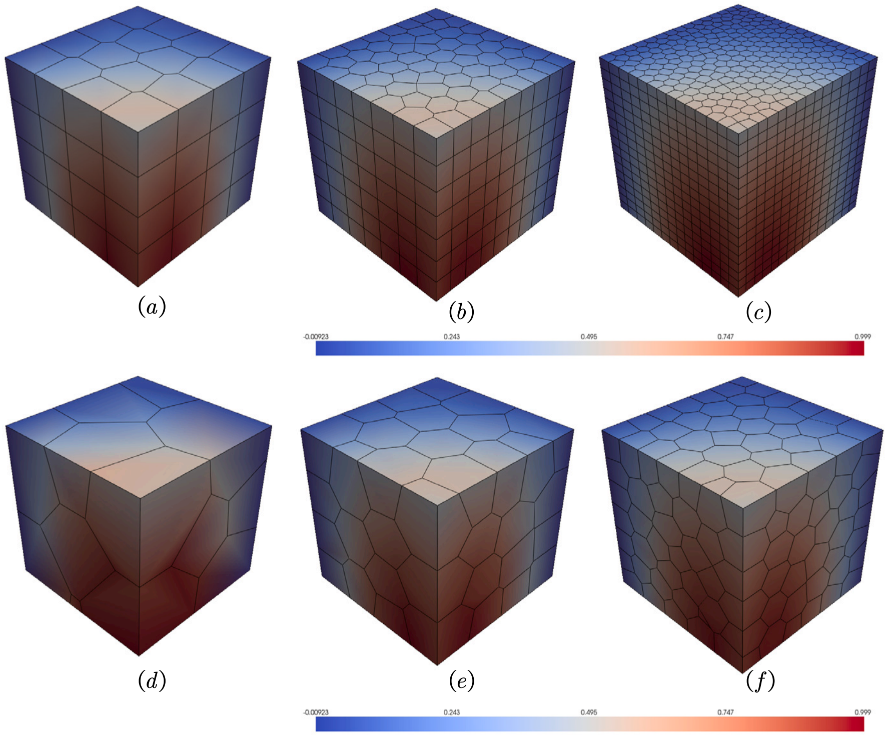
In this example, a first-order finite element method (Q1) and the conventional virtual element method (VEM) are used for comparison. In conventional VEM, the stabilization term is needed and can be selected as

$$\boldsymbol{K}_s = \frac{\beta}{n_E}\sqrt{\sum_{i=1}^{3n_E}\left(K_{ii}^c\right)^2}, \tag{78}$$



(a) (b)

Fig. 11. Comparison of the convergence of the standard VEM and the SFVEM for CVT mesh (a) L^2 error and (b) H^1 error.



(a) (b) (c) (d) (e) (f)

Fig. 12. Contour plots obtained by the SFVEM for different meshes.

where $\boldsymbol{K}^c$ is the consistent stiffness matrix, β is a parameter, which will be selected as $\beta = 0.5$ and $\beta = 1.0$ in this example.

To compare the performance of the proposed SFVEM, we perform a convergence study by adopting various meshes using uniform refinement. As shown in Fig. 13, the element division number is selected as $2^N \times 2^N$ in the horizontal and vertical directions. In this example, we select $N = 2, 3, 4, 5$. The maximum vertical displacements u_y obtained by different numerical methods are given in Fig. 14 for polyhedral swept mesh (irregular mesh) and regular mesh, respectively. The values of maximum vertical displacement u_y for different element division N are provided in Tables 1 and 2. Since $l = 3$ requires high-order numerical integration, which is associated with a large amount of computational time at the element level, in this example, only $l = 1$ and $l = 2$ are selected for comparison.

It can be seen that for both regular and irregular meshes, SFVEM has better accuracy and convergence speed than conventional VEM. For regular meshes, the results obtained by SFVEM are similar to the results obtained by FEM, while traditional VEM has not a such a good coarse mesh but accuracy convergences for a fine mesh. It should be noted that for $l = 1$, the number of vertices of some elements is greater than the limit described in Figs. 7 and 8. But it can be seen from Fig. 14 that the results are not biased. In addition, it should be noted that the parameter β influences the solution obtained by the conventional VEM. Among the two selected

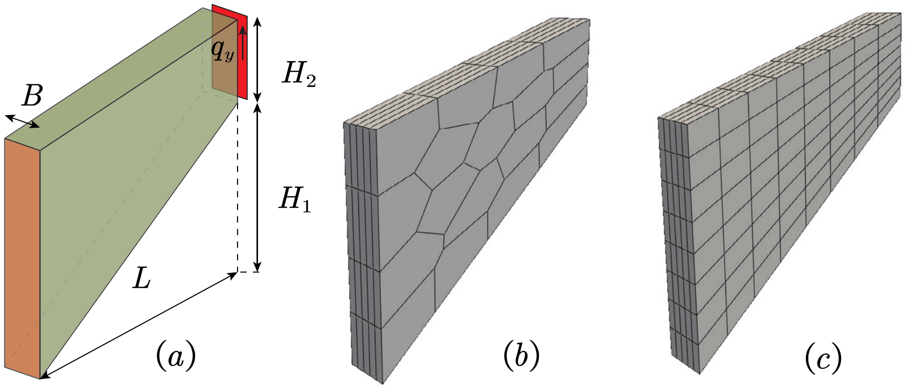


Fig. 13. 3D Cook's membrane problem for testing, (a) Geometry and boundary condition; (b) example of a polyhedral swept mesh; (c) example of a regular mesh.

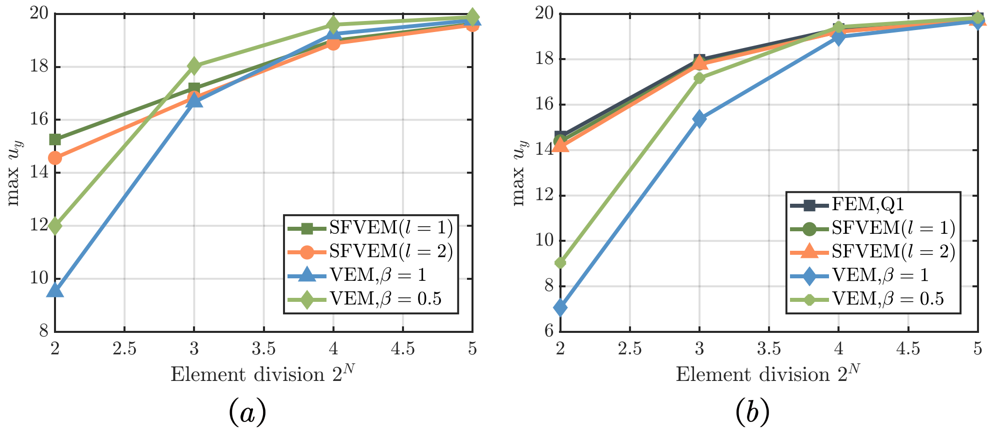


Fig. 14. Maximum value of vertical displacement u_y for different element division N, (a) for polyhedral swept mesh; (b) for regular mesh. For the polyhedral swept mesh, the FEM solutions obtained from the regular mesh are used for comparison.

Table 1
Maximum vertical displacement u_y for different element division N for polyhedral swept mesh.

N	SFVEM		VEM	
	$l = 1$	$l = 2$	$\beta = 1$	$\beta = 0.5$
2	15.2588	14.5610	9.5058	11.9912
3	17.1886	16.8322	16.6708	18.0341
4	18.9887	18.8809	19.2426	19.5925
5	19.6194	19.5820	19.7615	19.8809

Table 2
Maximum vertical displacement u_y for different element division N for regular mesh.

N	FEM	SFVEM		VEM	
	Q1	$l = 1$	$l = 2$	$\beta = 1$	$\beta = 0.5$
2	14.5985	14.3852	14.1481	7.0642	9.0339
3	17.9784	17.8170	17.7784	15.3717	17.1721
4	19.3357	19.2225	19.2180	18.9865	19.4231
5	19.8094	19.7168	19.7163	19.6870	19.8198

values ($\beta = 0.5$ and $\beta = 1.0$) in this example, the results are better when $\beta = 0.5$. Since the choice of the stabilization parameters affects the results, this is a good reason for developing SFVEM.

The contour plots of the von Mises stresses and stresses σ_{xx} obtained by SFVEM are depicted in Fig. 15 and Fig. 16 under different meshes and parameters l.

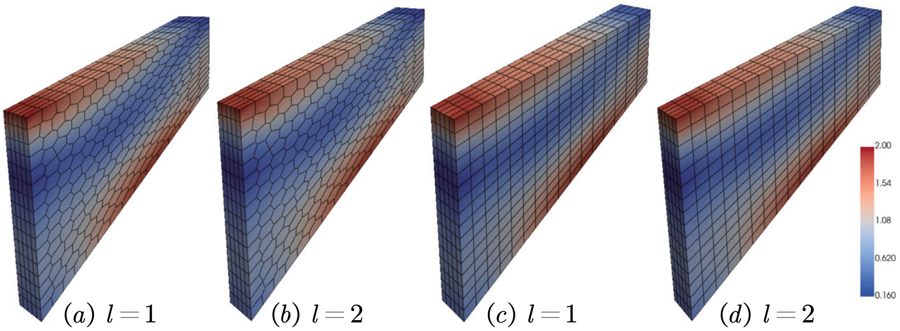


Fig. 15. Contour plots of von Mises stresses for different meshes and parameters; (a)–(b) for polyhedral swept mesh; (c)–(d) for regular mesh.

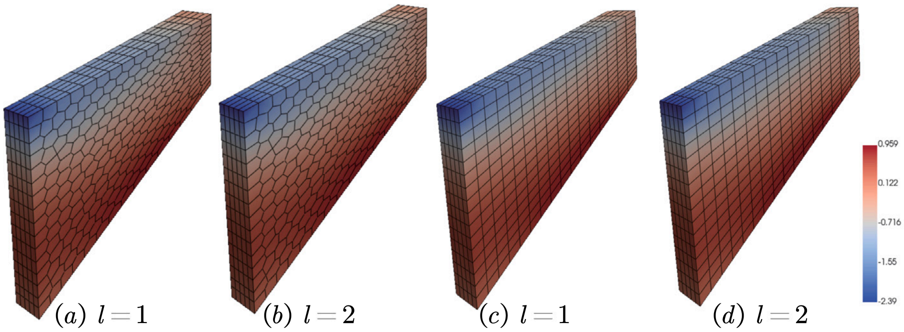


Fig. 16. Contour plots of normal stresses σ_{xx} for different meshes and parameters; (a)–(b) for polyhedral swept mesh; (c)–(d) for regular mesh.

6.4. T-shaped domain with nonmatching mesh

Since the number of nodes is arbitrary for a virtual element, parts with different sizes can be easily coupled. Hence the method can be better adapted to models with nonmatching meshes, which lead to "hanging nodes" in the finite element method. For example, a problem is modeled in patches or parts and finally assembled, or the mesh needs to be locally refined, then nonmatching meshes are easier to use with in a simulation without affecting the entire mesh due to local mesh changes.

As shown in Fig. 17, a T-shaped geometry is considered in this example. The upper and lower ends of the model are fixed, and a distributed vertical load $q_y = 1$ is applied at the right side (the direction of the load is shown in Fig. 17). The material parameters are selected as $E = 100$ and $\nu = 0.3$. Necessary dimensions are marked in Fig. 17. According to engineering experience, there is stress concentration at the chamfer of the model. To simulate this problem more accurately, it is necessary to use more elements in the stress concentration part. At the same time, to improve computational efficiency, the amount of mesh in other parts can be reduced appropriately. As shown in Fig. 18, different meshes and elements (polyhedral elements and hexahedral elements) are introduced for different parts. Between the parts, nonmatching meshes are directly converted into polyhedral meshes based on node positions. A Q1 finite element simulation, based on a fine mesh, is selected for comparison.

As shown in Fig. 17, two paths are defined to check the calculation results by different meshes and different methods. For the displacement in y-direction of the nodes on path 1, the results are shown in Fig. 19(a). Since there is no stress concentration in the part where path 1 is located, fewer nodes can be used in SFVEM. The displacement results obtained by SFVEM with different meshes compare well with the results obtained by the conventional FEM. In addition, as shown in Fig. 19(b) for path 2, more nodes are arranged at the chamfer (nodes are added directly on the edge of the element to form a polyhedral element), the stress calculation accuracy is higher and closer to the FEM calculation results based on a fine mesh.

In this example, to reduce the computing time, the order l of the polynomial was selected as $l = 1$. For $l = 2$, very accurate results can be obtained, but higher-order Gaussian integration and more computing time are necessary. In general, SFVEM (or VEM) can handle complex geometric models and problems with stress concentration thus more flexibly than FEM. The von Mises stresses using SFVEM are plotted in Fig. 20 for different meshes.

7. Conclusion

In this paper, we developed a first-order stabilization-free virtual element method for three-dimensional linear elasticity. Similar to the two-dimensional formulation, the main idea involves adapting the virtual element space to enable the computation of a higher-order L_2 projection operator of the gradient field (strain). The local enhancement space for polygonal faces and polyhedral

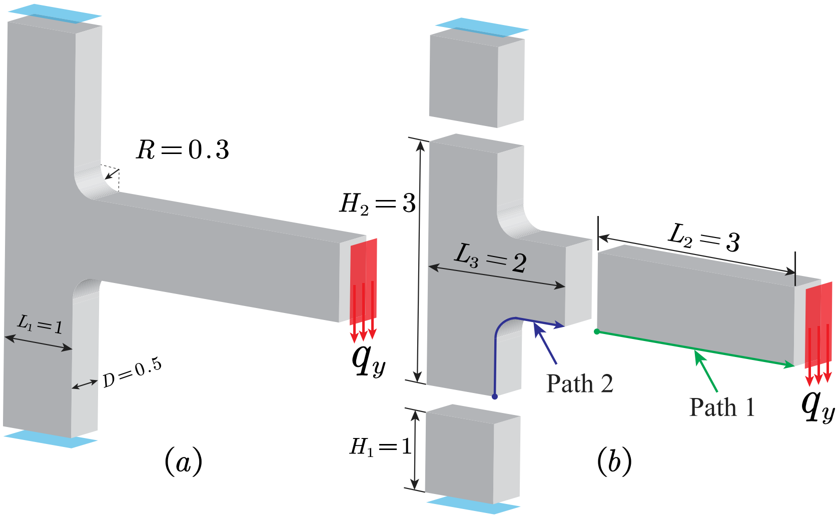


Fig. 17. T-shaped domain test, (a) computational domain; (b) different patches for the computational domain.

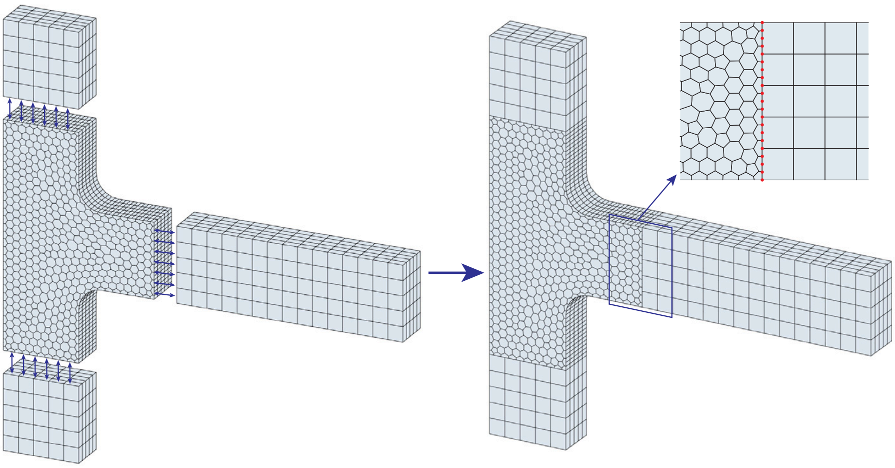


Fig. 18. Nonmatching mesh for the T-shaped domain test.

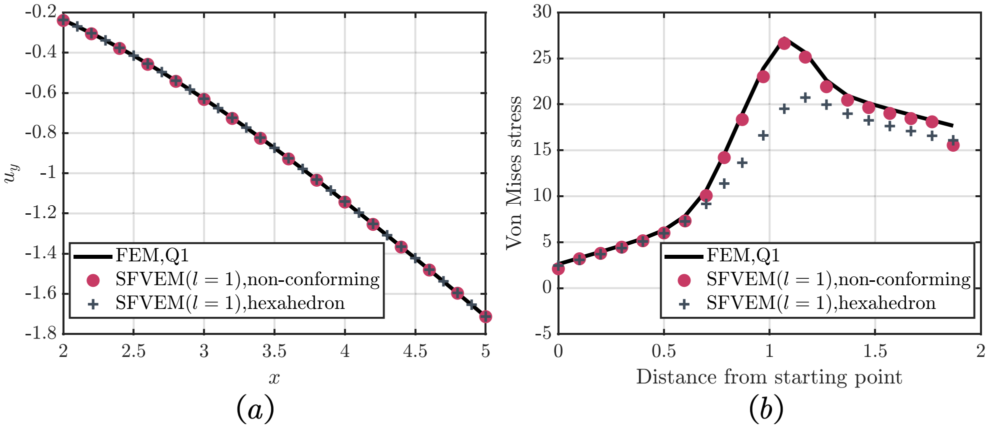


Fig. 19. Comparison of the results obtained by different methods, (a) displacements in y direction on path 1; (b) von Mises stresses on path 2.

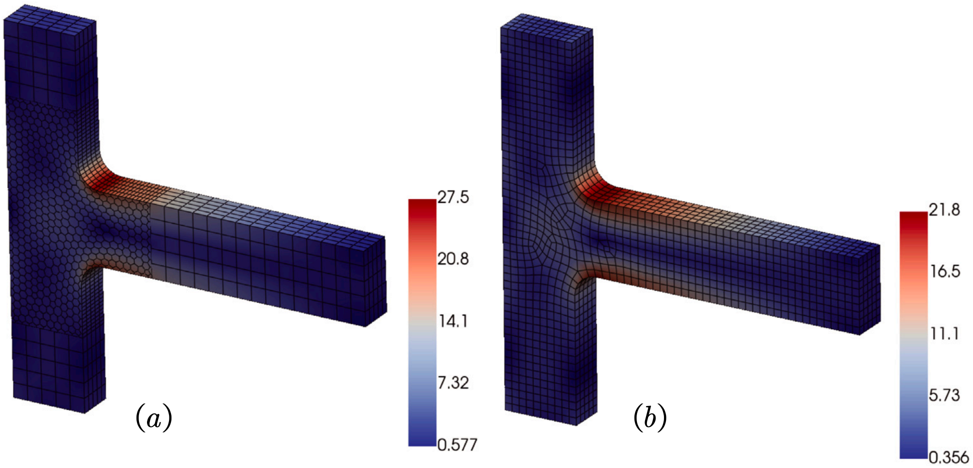


Fig. 20. Contour plots of von Mises stresses, (a) for non-conforming mesh; (b) for hexahedron mesh.

elements is introduced to calculate the higher-order L_2 projection (denoted as $\boldsymbol{\Pi}^m$). The projection operators are calculated based on the components of the gradient from which the matrix for the strain approximation is directly constructed. We numerically tested a suitable choice of the degree l of vector polynomials $\boldsymbol{N}^p$ for the interpolation of the gradient to ensure that the element stiffness matrix has the correct rank. Compared with traditional VEM, this method does not require any stabilization terms. We compared the convergence of the SFVEM under different orders l and obtained results similar to the traditional VEM. For mechanical problems exhibiting bending behavior, SFVEM yields more stable results, while the results of traditional VEM are affected by the choice of the stabilization parameter. In addition, SFVEM and VEM can deal with non-matching meshes of complex models and hence easily applicable to refine meshes locally in order to deal with stress concentration problems, thereby obtaining more accurate results. Since this stabilization-free formulation is computationally intensive, an adaptive technique to reduce the computational time should be developed, This method can be applied to 3D nonlinear problems in the future.

CRediT authorship contribution statement

Bing-Bing Xu: Writing – original draft, Visualization, Formal analysis, Data curation. **Peter Wriggers:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors are grateful for the support provided by the Alexander von Humboldt Foundation, Germany. The first author thanks Dr. Jian Meng and Professor Alessandro Russo for discussions on the enhancement space.

References

[1] L. Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Methods Appl. Sci. 23 (2012) 199–214, http://dx.doi.org/10.1142/S0218202512500492.

[2] L. Veiga, F. Brezzi, L. Marini, A. Russo, Mixed virtual element methods for general second order elliptic problems on polygonal meshes, ESAIM Math. Model. Numer. Anal. 26 (2014) 727–747, http://dx.doi.org/10.1051/m2an/2015067.

[3] N. Sukumar, A. Tabarraei, Conformal polygonal finite elements, Internat. J. Numer. Methods Engrg. 61 (2004) 2045–2066, http://dx.doi.org/10.1002/nme.1141.

[4] H. Nguyen-Xuan, A polygonal finite element method for plate analysis, Comput. Struct. 188 (2017) 45–62, http://dx.doi.org/10.1016/j.compstruc.2017.04.002.

[5] T. Strouboulis, I. Babuška, K. Copps, The design and analysis of the Generalized Finite Element Method, Comput. Methods Appl. Mech. Engrg. 181 (2000) 43–69, http://dx.doi.org/10.1016/S0045-7825(99)00072-9.

[6] D. van Huyssteen, F. Lopez Rivarola, G. Etse, P. Steinmann, On mesh refinement procedures for the virtual element method for two-dimensional elastic problems, Comput. Methods Appl. Mech. Engrg. 393 (2022) 114849, http://dx.doi.org/10.1016/j.cma.2022.114849.

[7] D. van Huyssteen, F. Lopez Rivarola, G. Etse, P. Steinmann, On adaptive mesh coarsening procedures for the virtual element method for two-dimensional elastic problems, Comput. Methods Appl. Mech. Engrg. 418 (2023) 116507, http://dx.doi.org/10.1016/j.cma.2023.116507.

[8] L. Veiga, F. Brezzi, Virtual elements for linear elasticity problems, SIAM J. Numer. Anal. 51 (2) (2013) 794–812, http://dx.doi.org/10.1137/120874746.

[9] A. Gain, C. Talischi, G. Paulino, On the virtual element method for three-dimensional elasticity problems on arbitrary polyhedral meshes, Comput. Methods Appl. Mech. Engrg. 282 (2013) 132–160, http://dx.doi.org/10.1016/j.cma.2014.05.005.

[10] E. Artioli, L. Veiga, C. Lovadina, E. Sacco, Arbitrary order 2D virtual elements for polygonal meshes: Part I, elastic problem, Comput. Mech. 60 (2017) 727–747, http://dx.doi.org/10.1007/s00466-017-1404-5.

[11] F. Dassi, C. Lovadina, M. Visinoni, A three-dimensional Hellinger-Reissner virtual element method for linear elasticity problems, Comput. Methods Appl. Mech. Engrg. 364 (2020) 112910, http://dx.doi.org/10.1016/j.cma.2020.112910.

[12] M. Mengolini, M. Benedetto, A. Aragón, An engineering perspective to the virtual element method and its interplay with the standard finite element method, Comput. Methods Appl. Mech. Engrg. 350 (2019) 995–1023, http://dx.doi.org/10.1016/j.cma.2019.02.043.

[13] H. Chi, L. Veiga, G. Paulino, Some basic formulations of the virtual element method (VEM) for finite deformations, Comput. Methods Appl. Mech. Engrg. 318 (2016) 995–1023, http://dx.doi.org/10.1016/j.cma.2016.12.020.

[14] P. Wriggers, B. Reddy, W. Rust, B. Hudobivnik, Efficient virtual element formulations for compressible and incompressible finite deformations, Comput. Mech. 60 (2017) 995–1023, http://dx.doi.org/10.1007/s00466-017-1405-4.

[15] D. van Huyssteen, B. Reddy, A virtual element method for isotropic hyperelasticity, Comput. Methods Appl. Mech. Engrg. 367 (2020) 113134, http://dx.doi.org/10.1016/j.cma.2020.113134.

[16] m.l. de bellis, P. Wriggers, B. Hudobivnik, Serendipity virtual element formulation for nonlinear elasticity, Comput. Struct. 223 (2019) 106094, http://dx.doi.org/10.1016/j.compstruc.2019.07.003.

[17] P. Wriggers, W. Rust, B. Reddy, A virtual element method for contact, Comput. Mech. 58 (2016) 995–1023, http://dx.doi.org/10.1007/s00466-016-1331-x.

[18] F. Aldakheel, B. Hudobivnik, E. Artioli, L. Veiga, P. Wriggers, Curvilinear virtual elements for contact mechanics, Comput. Methods Appl. Mech. Engrg. 372 (2020) 113394, http://dx.doi.org/10.1016/j.cma.2020.113394.

[19] W. Shen, M. Ohsaki, J. Zhang, A 2-dimentional contact analysis using second-order virtual element method, Comput. Mech. 70 (2022) 995–1023, http://dx.doi.org/10.1007/s00466-022-02165-y.

[20] M. Cihan, B. Hudobivnik, J. Korelc, P. Wriggers, A virtual element method for 3D contact problems with non-conforming meshes, Comput. Methods Appl. Mech. Engrg. 402 (2022) 115385, http://dx.doi.org/10.1016/j.cma.2022.115385.

[21] K. Park, H. Chi, G. Paulino, On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration, Comput. Methods Appl. Mech. Engrg. 356 (2019) 669–684, http://dx.doi.org/10.1016/j.cma.2019.06.031.

[22] K. Park, H. Chi, G. Paulino, Numerical recipes for elastodynamic virtual element methods with explicit time integration, Internat. J. Numer. Methods Engrg. 121 (2019) 1–31, http://dx.doi.org/10.1002/nme.6173.

[23] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, Virtual element formulation for finite strain elastodynamics, Computer Modeling in Engineering and Sciences 129 (2021) 1151–1180, http://dx.doi.org/10.32604/cmes.2021.016851.

[24] N. Sukumar, M. Tupek, Virtual elements on agglomerated finite elements to increase the critical time step in elastodynamic simulations, Internat. J. Numer. Methods Engrg. 123 (2022) 4702–4725, http://dx.doi.org/10.1002/nme.7052.

[25] P. Wriggers, B. Hudobivnik, A low order virtual element formulation for finite elasto-plastic deformations, Comput. Methods Appl. Mech. Engrg. 327 (2017) 4702–4725, http://dx.doi.org/10.1016/j.cma.2017.08.053.

[26] B. Hudobivnik, F. Aldakheel, P. Wriggers, A low order 3D virtual element formulation for finite elasto–plastic deformations, Comput. Mech. 63 (2019) 4702–4725, http://dx.doi.org/10.1007/s00466-018-1593-6.

[27] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, 3D mixed virtual element formulation for dynamic elasto-plastic analysis, Comput. Mech. 68 (2021) 1–18, http://dx.doi.org/10.1007/s00466-021-02010-8.

[28] T.-R. Liu, F. Aldakheel, M. Aliabadi, Virtual element method for phase field modeling of dynamic fracture, Comput. Methods Appl. Mech. Engrg. 411 (2023) 116050, http://dx.doi.org/10.1016/j.cma.2023.116050.

[29] F. Aldakheel, B. Hudobivnik, P. Wriggers, Virtual element formulation for phase-field modeling of ductile fracture, Int. J. Multiscale Comput. Eng. 17 (2019) 181–200, http://dx.doi.org/10.1615/IntJMultCompEng.2018026804.

[30] A.L. Gain, G.H. Paulino, L.S. Duarte, I.F. Menezes, Topology optimization using polytopes, Comput. Methods Appl. Mech. Engrg. 293 (2015) 411–430, http://dx.doi.org/10.1016/j.cma.2015.05.007.

[31] O. Certik, F. Gardini, G. Manzini, L. Mascotto, G. Vacca, The p - and h p -versions of the virtual element method for elliptic eigenvalue problems, Comput. Math. Appl. 79 (2019) 4702–4725, http://dx.doi.org/10.1016/j.camwa.2019.10.018.

[32] J. Meng, G. Wang, L. Mei, Mixed virtual element method for the Helmholtz transmission eigenvalue problem on polytopal meshes, IMA J. Numer. Anal. 43 (2022) 4702–4725, http://dx.doi.org/10.1093/imanum/drac019.

[33] P. Wriggers, B. Hudobivnik, F. Aldakheel, NURBS-based geometries: A mapping approach for virtual serendipity elements, Comput. Methods Appl. Mech. Engrg. 378 (2021) 113732, http://dx.doi.org/10.1016/j.cma.2021.113732.

[34] A. D'Altri, S. Miranda, L. Patruno, E. Sacco, An enhanced VEM formulation for plane elasticity, Comput. Methods Appl. Mech. Engrg. 376 (2021) 113663, http://dx.doi.org/10.1016/j.cma.2020.113663.

[35] S. Berrone, A. Borio, F. Marcon, Lowest order stabilization free virtual element method for the Poisson equation, 2021, arXiv preprint:2103.16896.

[36] A. Lamperti, M. Cremonesi, U. Perego, A. Russo, C. Lovadina, A Hu-Washizu variational approach to self-stabilized virtual elements: 2D linear elastostatics, Comput. Mech. 71 (2023) 1–21, http://dx.doi.org/10.1007/s00466-023-02282-2.

[37] J. Meng, X. Wang, L. Bu, L. Mei, A lowest-order free-stabilization Virtual Element Method for the Laplacian eigenvalue problem, J. Comput. Appl. Math. 410 (2022) 114013, http://dx.doi.org/10.1016/j.cam.2021.114013.

[38] A. Chen, N. Sukumar, Stabilization-free virtual element method for plane elasticity, Comput. Math. Appl. 138 (2023) 88–105, http://dx.doi.org/10.1016/j.camwa.2023.03.002.

[39] A. Chen, N. Sukumar, Stabilization-free serendipity virtual element method for plane elasticity, Comput. Methods Appl. Mech. Engrg. 404 (2023) 115784, http://dx.doi.org/10.1016/j.cma.2022.115784.

[40] B.-B. Xu, F. Peng, P. Wriggers, Stabilization-free virtual element method for finite strain applications, Comput. Methods Appl. Mech. Engrg. 417 (2023) 116555, http://dx.doi.org/10.1016/j.cma.2023.116555.

[41] R. Taylor, VEM prototypes, 2023, Privat communication, 2023.

[42] H. Fang, D. Zhang, Q. Fang, L. Cao, M. Wen, An efficient patch-to-patch method for coupling independent finite element subdomains with intersecting interfaces, Comput. Methods Appl. Mech. Engrg. 388 (2022) 114209, http://dx.doi.org/10.1016/j.cma.2021.114209.

[43] D. Sohn, S. Jin, Polyhedral elements with strain smoothing for coupling hexahedral meshes at arbitrary nonmatching interfaces, Comput. Methods Appl. Mech. Engrg. 293 (2015) 92–113, http://dx.doi.org/10.1016/j.cma.2015.04.007.

[44] L. Beirao da Veiga, F. Dassi, A. Russo, High-order virtual element method on polyhedral meshes, Comput. Math. Appl. 74 (5) (2017) 1110–1122, http://dx.doi.org/10.1016/j.camwa.2017.03.021.

[45] F. Dassi, L. Mascotto, Exploring high-order three dimensional virtual elements: Bases and stabilizations, Comput. Math. Appl. 75 (9) (2018) 3379–3401, http://dx.doi.org/10.1016/j.camwa.2018.02.005.

[46] T. Sorgente, S. Biasotti, G. Manzini, M. Spagnuolo, Polyhedral mesh quality indicator for the virtual element method, Comput. Math. Appl. 114 (2022) 151–160, http://dx.doi.org/10.1016/j.camwa.2022.03.042.

[47] B. Ahmad, A. Alsaedi, F. Brezzi, L. Marini, A. Russo, Equivalent projectors for virtual element methods, Comput. Math. Appl. 66 (2013) 376–391, http://dx.doi.org/10.1016/j.camwa.2013.05.015.

[48] L. Beirao da Veiga, F. Brezzi, L.D. Marini, A. Russo, The Hitchhiker's Guide to the virtual element method, Math. Models Methods Appl. Sci. 24 (08) (2014) 1541–1573, http://dx.doi.org/10.1142/S021820251440003X.

[49] P. Wriggers, F. Aldakheel, B. Hudobivnik, Virtual Element Methods in Engineering Sciences, first ed., Springer International Publishing, 2024, http://dx.doi.org/10.1007/978-3-031-39255-9.

[50] S. Berrone, A. Borio, F. Marcon, G. Teora, A first-order stabilization-free Virtual Element Method, Appl. Math. Lett. 142 (2023) 108641, http://dx.doi.org/10.1016/j.aml.2023.108641.