Computer Methods in Applied Mechanics and Engineering 417 (2023) 116555

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

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

Stabilization-free virtual element method for finite strain applications

Bing-Bing Xu [a,*], Fan Peng [b], Peter Wriggers [a]

[a] Institute of Continuum Mechanics, Leibniz Universität Hannover, Hannover, Germany
[b] School of Science, Chang'an University, Xi'an 710064, China

ARTICLE INFO

Keywords:
virtual element method
Stabilization-free
Hyperelastic material

ABSTRACT

In this paper, a novel higher stabilization-free virtual element method is proposed for compressible hyper-elastic materials in 2D. Different from the most traditional virtual element formulation, the method does not need any stabilization. The main idea is to modify the virtual element space to allow the computation of a higher-order polynomial L_2 projection of the gradient. Based on that the stiffness matrix can be obtained directly which greatly simplifies the analysis process, especially for nonlinear problems. Hyper-elastic materials are considered and some benchmark nonlinear problems are solved to verify the capability and accuracy of the stabilization-free virtual element method.

1. Introduction

In recent years, some advanced numerical methods based on polygonal meshes were proposed. These improve the ability and flexibility to discretize computational domains with high geometrical complexity. Some of the techniques can be listed as Polygonal Finite Element Method (PFEM) [1,2], Discontinuous Galerkin Methods (DG) [3,4], Mimetic Finite Difference Methods (MFD) [5–7], and some other methods like the Extended FEMs (XFEM) [8] and Generalized FEMs (GFEM) [9]. In addition to these established numerical methods, the Virtual element method (VEM), which was introduced in [10–12], has gained increasing attention. The VEM can be regarded as a generalization of the classical finite element method (FEM) to general polygonal meshes. Different from conventional polygonal/polyhedral methods, the basis functions in VEM are defined by the local partial differential equations (PDEs) but never need to be explicitly computed. Up to now, the VEM has been introduced and developed for the Poisson equation [10], linear elastic [13–17], hyperelastic materials at finite deformations [18–21], contact problems [22–25], elastodynamics problems [26–28], phase field modeling problems [29,30] and for eigenvalue problems [31,32].

It is well-known that the VEM used in the above-introduced works requires stabilization to avoid the zero-energy modes and to ensure the stability of the method. The stabilization techniques, discussed for classical problems in [10,29], are based on a formulation that involves the degrees of freedom. However, the stabilization methods sometimes have bad performances for bending-dominated problems and rely on a chosen stabilization parameter. Some useful stabilization approaches are given for nonlinear problems in [19]. It should be pointed out that there are various formats [14,33] for the construction of the stabilization term, including the selection of different and solution-dependent parameters, especially for complex nonlinear problems.

A stabilization-free virtual element formulation (SFVEM) only relies on the constant data stemming from the mechanical model, then, especially in the nonlinear range SFVEM will increase the use of VEM in engineering analysis. Recently, some new techniques have been proposed to construct stabilization-free or self-stabilized virtual elements schemes. An enhanced VEM formulation that can

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

https://doi.org/10.1016/j.cma.2023.116555
Received 7 July 2023; Received in revised form 16 October 2023; Accepted 16 October 2023

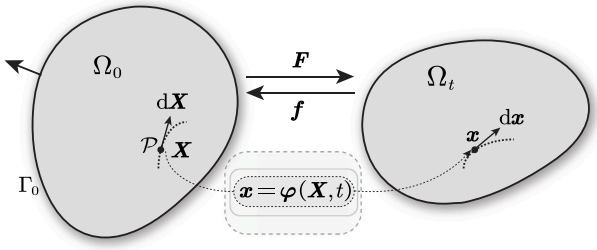

Fig. 1. Reference and current configurations in the geometrically nonlinear analysis.

bypass the need for stabilization terms for $k = 1$ was proposed in [34]. Similar to this enhanced formulation, a self-stabilized virtual element was proposed by Lamperti [35] for 2D linear elastostatics based on the Hu–Washizu variational approach. Besides, the stabilization-free virtual element method (Enlarged Enhancement Virtual Element Methods) was proposed by Berrone in [36,37] for the 2D Poisson equation. The main idea of the method is to modify the first-order virtual element space to allow the computation of a higher-order polynomial L_2 projection of the gradient [38] to delete the stabilization term. Under the right conditions, they proved the well-posedness and error estimation of the 2D Poisson equation, laying the foundation for solving complex and challenging problems. In the following, the stabilization-free method has been extended to solid mechanics ($k = 1$ and $k = 2$) [38,39], as well as to eigenvalue problems [40]. But so far, the stabilization-free formulation has not been applied to nonlinear mechanical problems.

The main aim of this work is to extend the work in [38,39] to hyperelastic materials at finite deformations. In this first step, we try to re-deduce the logical framework of the entire stabilization-free VEM, to obtain a unified approximate formulation for the gradient. The specific process of calculation of the H_1 projection matrix $\boldsymbol{\Pi}^{\nabla}$ (original elliptic projection matrix) and the L_2 projection matrix $\boldsymbol{\Pi}^{m}$ are given in a form that is more in line with the approach to mechanical problems. By employing the enhancement space given in [12], it is shown how the stabilization-free method can be extended to higher-order interpolations (for $k > 2$). Then, by using an approximate expression for the gradients, we obtain a non-linear incremental iteration format similar to conventional finite element methods. It is easy to find that the method does not need any stabilization terms for linear as well as for nonlinear problems, so it simplifies the element formulation, especially for nonlinear problems. The method can be easily extended to other types of nonlinear problems by computing the gradient of the variable in each polygonal element.

The paper is organized as follows. The governing equations for hyperelastic problems are reviewed and presented in Section 2. Then in Section 3, we introduce the function spaces used in stabilization-free VEM and calculate different forms of projection operators. The stabilization-free virtual element formulation for hyperelastic problems is given in Section 4. Some numerical examples are presented and discussed in Section 5. The paper closes with some concluding remarks in Section 6.

2. Mathematical formulation

2.1. Governing equations for finite strain elasticity

In this work, let us consider a body that occupies a bounded domain $\Omega \in \mathbb{R}^2$, see Fig. 1. Let $\Gamma = \partial\Omega = \Gamma_D \cup \Gamma_N$ be the boundary of Ω with Γ_D the Dirichlet boundary and Γ_N the Neumann boundary and $\Gamma_D \cap \Gamma_N = \varnothing$.

Consider a general solid undergoes a motion $\boldsymbol{\varphi}$ so that the geometry changes from the initial to the current state. Each material point initially at $\boldsymbol{X}$ is characterized by the position $\boldsymbol{x}$ at time t given by the motion

$$\boldsymbol{x} = \boldsymbol{\varphi}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t), \tag{1}$$

where $\boldsymbol{u}$ is the displacement. We denote by $\boldsymbol{F}$ the deformation gradient defined by

$$\boldsymbol{F} = \mathrm{Grad}\boldsymbol{\varphi} = \frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}\boldsymbol{X}} = \boldsymbol{I} + \nabla\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{I}$ is the identity tensor. The right Cauchy–Green deformation tensor can be defined as

$$\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F}, \tag{3}$$

and the Green–Lagrange strain tensor is provided by

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}). \tag{4}$$

Considering the variables related to the initial configuration, the standard equilibrium equation for the model problem is given by

$$-\mathrm{Div}\boldsymbol{P} = \boldsymbol{f}, \tag{5}$$

where $\boldsymbol{P}$ is the first Piola–Kirchhoff stress and $\boldsymbol{f}$ is the body force. The Dirichlet and Neumann boundary conditions are described as

$$\boldsymbol{u} = \bar{\boldsymbol{u}} \quad \text{on} \quad \Gamma_D, \tag{6}$$

$$\boldsymbol{P} \cdot \boldsymbol{N} = \bar{\boldsymbol{t}} \quad \text{on} \quad \Gamma_N, \tag{7}$$

where $\boldsymbol{N}$ is the outward normal vector, $\bar{\boldsymbol{u}}$ is the prescribed displacement and $\bar{\boldsymbol{t}}$ is the prescribed surface traction.

Naturally, it is simpler to work in the initial configuration with symmetrical stress tensors. Then, the second Piola–Kirchhoff stress is introduced as

$$\boldsymbol{S} = \boldsymbol{F}^{-1} \cdot \boldsymbol{P} = J\boldsymbol{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \boldsymbol{F}^{-T}, \tag{8}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress tensor (real stress), which refers to the current configuration; J is the determinant of the deformation gradient tensor, which describes the volume change between the current and initial configuration

$$J = \det(\boldsymbol{F}). \tag{9}$$

We have the following work conjugate relationship

$$\mathcal{W} = \int_{V_0} \delta\boldsymbol{E} : \boldsymbol{S} \mathrm{d}V_0 = \int_{V_0} \delta\boldsymbol{F} : \boldsymbol{P} \mathrm{d}V_0, \tag{10}$$

where $\delta\boldsymbol{E}$ is the variation of the Green strain tensor, $\delta\boldsymbol{F}$ is the variation of deformation gradient tensor,

$$\delta\boldsymbol{E} = \frac{1}{2}\delta\left(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I}\right) = \frac{1}{2}\left(\delta\boldsymbol{F}^T \cdot \boldsymbol{F} + \boldsymbol{F}^T \cdot \delta\boldsymbol{F}\right). \tag{11}$$

2.2. Constitutive equations for hyperelastic material

For a homogeneous compressible isotropic hyperelastic material, the elastic behavior of a deformable body can be specified in terms of a hyperelastic energy density Ψ. For convenience, the strain energy function Ψ is often expressed as a function of the right Cauchy–Green tensor $\boldsymbol{C} = \boldsymbol{F}^T \cdot \boldsymbol{F}$ or its invariants as

$$\Psi\left(\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{X}\right) = \Psi\left(\boldsymbol{C}(\boldsymbol{X}), \boldsymbol{X}\right) = \Psi\left(I_C, II_C, III_C, \boldsymbol{X}\right), \tag{12}$$

where the invariants I_C, II_C, III_C are defined as

$$\begin{aligned} I_C &= \mathrm{tr}\,\boldsymbol{C} = \boldsymbol{C} : \boldsymbol{I} \\ II_C &= \frac{1}{2}\left[(\mathrm{tr}\,\boldsymbol{C})^2 - \mathrm{tr}\,\boldsymbol{C}^2\right] \\ III_C &= \det\boldsymbol{C} = J^2. \end{aligned} \tag{13}$$

Based on the three invariants, a general form of strain energy density can be defined

$$\Psi(I_C, II_C, III_C) = \sum_{m+n+k=1}^{\infty} A_{mnk}\left(I_C - 3\right)^m \left(II_C - 3\right)^n \left(III_C - 1\right)^k. \tag{14}$$

In this work, we focus on the neo-Hookean hyperelastic model

$$\Psi = \frac{\mu}{2}\left(I_C - 3\right) - \mu \ln J + \frac{\lambda}{2}(\ln J)^2, \tag{15}$$

where the Lame parameters λ and μ are given as

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu = \frac{E}{2(1+\nu)}. \tag{16}$$

where E and ν denote Young modulus and Poisson ratio, respectively.

Considering $\frac{\partial J}{\partial \boldsymbol{C}} = \frac{1}{2}J\boldsymbol{C}^{-1}$, the second Piola–Kirchhoff stress tensor $\boldsymbol{S}$ for the given neo-Hookean hyperelastic model can be obtained as

$$\begin{aligned} \boldsymbol{S} &= 2\frac{\partial \Psi}{\partial \boldsymbol{C}} = \mu\boldsymbol{I} - 2\mu\frac{1}{J}\frac{\partial J}{\partial \boldsymbol{C}} + 2\lambda(\ln J)\frac{1}{J}\frac{\partial J}{\partial \boldsymbol{C}} \\ &= \mu\left(\boldsymbol{I} - \boldsymbol{C}^{-1}\right) + \lambda(\ln J)\boldsymbol{C}^{-1}. \end{aligned} \tag{17}$$

In hyperelasticity, the constitutive tensor $\boldsymbol{\mathcal{D}}$ can be achieved by differentiating the second Piola–Kirchhoff stress $\boldsymbol{S}$ as

$$\boldsymbol{\mathcal{D}} = 2\frac{\partial \boldsymbol{S}}{\partial \boldsymbol{C}} = \lambda\boldsymbol{C}^{-1} \otimes \boldsymbol{C}^{-1} + 2(\mu - \lambda \ln J)\boldsymbol{\mathcal{L}}, \tag{18}$$

where

$$\mathcal{L}_{IJKL} = -\frac{\partial\left(\boldsymbol{C}^{-1}\right)_{IJ}}{\partial C_{KL}} = \frac{1}{2}\left[\left(\boldsymbol{C}^{-1}\right)_{IK}\left(\boldsymbol{C}^{-1}\right)_{JL} + \left(\boldsymbol{C}^{-1}\right)_{IL}\left(\boldsymbol{C}^{-1}\right)_{JK}\right]. \tag{19}$$

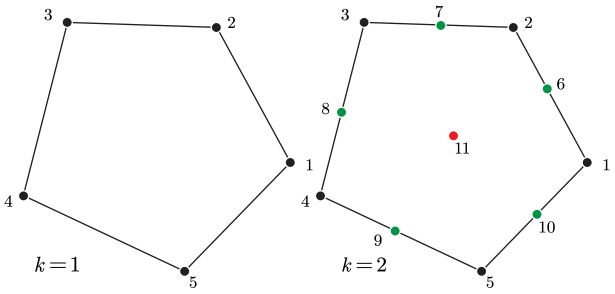

Fig. 2. First and second order virtual element.

3. Function space and projection operators

In this part, we start to describe the basic idea of the virtual element method (VEM) and the stabilization-free virtual element method (SFVEM). To describe the idea of the method more clearly, it is necessary to provide definitions and constructions of different projection operators in advance.

Let $\mathcal{J}_h$ be the decomposition of the computational domain Ω into a set of nonoverlapping polygonal elements E (as given in Fig. 2), $\mathcal{E}_h$ being the set of edges e of $\mathcal{J}_h$. The symbol n_E represents the number of edges of a polygon E and the mesh size h_E is the diameter of E. Then we can denote by $\mathcal{M}_k(E)$ the scaled $(k+1)(k+2)/2$ monomials:

$$\mathcal{M}_k(E) := \left\{ \left(\frac{\boldsymbol{x} - \boldsymbol{x}_E}{h_E} \right)^s, |s| \le k \right\}, \tag{20}$$

where $\boldsymbol{x}_E$ are the cartesian coordinates of the centroid of E, $|s| := s_1 + s_2$ and $\boldsymbol{x}^s := x_1^{s_1} x_2^{s_2}$.

Different from the finite element method, in order to deal with polygonal elements of arbitrary shape, VEM uses general spaces that may contain non-polynomial functions [17]. To compute the contribution of these non-polynomial functions to the stiffness matrix, one has to compute a local projector on the space of polynomials of degree $\le k$ [12].

3.1. Original elliptic projection operator

In the classical VEM, the projection operator is selected as the energy projector $\Pi_k^{\nabla}(E)$, which is defined as

$$\Pi_k^{\nabla}(E) : \mathcal{V}_k(E) \to \mathcal{P}_k(E). \tag{21}$$

For simplicity', the projector $\Pi_k^{\nabla}(E)$ can be written as Π_k^{∇}. The basis function $\phi_i \in \mathcal{V}_k(E)$ are defined as

$$\chi_i(\phi_j) = \mathrm{dof}(\phi_j) = \delta_{ij}, j, j = 1, \dots, N_E, \tag{22}$$

where χ is the degree of freedom, $N_E := \dim \mathcal{V}_k(E)$.

To construct the projector Π_k^{∇}, let us consider the virtual element space

$$\mathcal{V}_k(E) := \left\{ u_h \in H^1(E) : \Delta u \in \mathcal{P}_{k-2}(E) \quad \text{in} \quad E, \quad u|_{\partial E} = \mathcal{B}_k(\partial E) \right\} \tag{23}$$

where $\mathcal{P}_k$ is a polynomial with the highest order not exceeding k,

$$\mathcal{B}_k(\partial E) := \left\{ u_h \in C(\partial E) : u_e \in \mathcal{P}_k(e), \quad e \subset \partial E \right\} \tag{24}$$

In $\mathcal{V}_k(E)$, the degrees of freedom are selected as

- $\chi^1(E)$: for $k \ge 1$, the values of u_h at the vertices;
- $\chi^2(E)$: for $k > 1$, the values of u_h at $k-1$ uniformly spaced points on each edge e;
- $\chi^3(E)$: for $k > 1$, the moments

$$\frac{1}{|E|} \int_E u_h m_\alpha \mathrm{d}\Omega, \quad \forall m_\alpha \in \mathcal{M}_{k-2}(E).$$

It is not difficult to find that $\mathcal{B}_k(\partial E)$ is a linear space of dimension $n_E + n_E(k-1) = n_E k$, where n_E is the number of edges of the polygon element. The dimension of $\mathcal{V}_k(E)$ is

$$\dim \mathcal{V}_k(E) = N_E = n_E + n_E(k-1) + \frac{k(k-1)}{2} = n_E + \frac{k(k-1)}{2} \tag{25}$$

where the last term corresponds to the dimension of polynomials of degree $\le k-2$ in two dimensions.

Based on the above definition, for any given $E \in \mathcal{J}_h$, the projection $\Pi_k^{\nabla}(E)$ defined for $u \in \mathcal{V}_k(E)$ is calculated by the orthogonality condition: $\forall u \in \mathcal{H}^1(E)$,

$$\int_E \nabla\left(\Pi_k^{\nabla} u - u_h\right) \cdot \nabla p \, \mathrm{d}\Omega = 0 \Rightarrow \int_E \nabla \Pi_k^{\nabla} u \cdot \nabla p \, \mathrm{d}\Omega = \int_E \nabla u_h \cdot \nabla p \, \mathrm{d}\Omega, \quad \forall p \in \mathcal{P}_k(E). \tag{26}$$

On expanding the right-hand side of Eq. (26), using integration by parts and considering the Green formula, we obtain

$$\int_E \nabla \Pi_k^{\nabla} u \cdot \nabla p \, \mathrm{d}\Omega = -\int_E u_h \cdot \Delta p \, \mathrm{d}\Omega + \int_{\partial E} u_h \cdot \frac{\partial p}{\partial \boldsymbol{n}} \mathrm{d}\Gamma. \tag{27}$$

The condition Eq. (27) can be extended to include the constant part of u_h by prescribing a projection operator onto constants $P_0 : \mathcal{H}^1(E) \to \mathcal{P}_0(E)$,

$$P_0\left(\Pi|_k^{\nabla} u - u_h\right) = 0. \tag{28}$$

The constants is selected as

$$P_0(u) := \frac{1}{n_E} \sum_{i=1}^{n_E} u_h \quad \text{for} \quad k = 1, \tag{29}$$

$$P_0(u) := \frac{1}{|E|} \int_E u_h \mathrm{d}\Omega \quad \text{for} \quad k \geq 2. \tag{30}$$

Since $\mathcal{M}_k(E)$ is a basis for $\mathcal{P}_k(E)$, the projection $\Pi_k^{\nabla} \phi_i$ in Eq. (26) can be expanded in the basis of $\mathcal{P}_k(E)$ as

$$\Pi_k^{\nabla} \phi_i = \sum_{\alpha=1}^{N_P} a_{\alpha,i} m_\alpha = \sum_{j=1}^{N_E} s_{j,i} \phi_j, \tag{31}$$

where $N_P := \dim \mathcal{P}_k(E)$. Then Eq. (31) can be written in the matrix form as

$$\left[\Pi_k^{\nabla} \phi_1, \Pi_k^{\nabla} \phi_2, \ldots, \Pi_k^{\nabla} \phi_{N_K}\right] = \Pi_k^{\nabla} \boldsymbol{\phi}^T = \boldsymbol{m}^T \boldsymbol{\Pi}_{k*}^{\nabla} = \boldsymbol{\phi}^T \boldsymbol{\Pi}_k^{\nabla}, \tag{32}$$

where $\boldsymbol{\phi}^T = [\phi_1, \phi_2, \ldots, \phi_{N_E}]$, $\boldsymbol{m}^T = [m_1, m_2, \ldots, m_{N_P}]$, $\boldsymbol{\Pi}_k^{\nabla}$ is the matrix representation of the operator Π_k^{∇}. Furthermore, $\boldsymbol{\Pi}_{k*}^{\nabla}$ is the matrix representation of the operator Π_k^{∇} acting from $\mathcal{V}_k(E)$ to $\mathcal{P}_k(E)$. The two different basis functions are connected by

$$\boldsymbol{m}^T = \boldsymbol{\phi}^T \boldsymbol{D} \tag{33}$$

where matrix $\boldsymbol{D}$ is the transition matrix with size $N_E \times N_P$ given by

$$\boldsymbol{D}_{j\alpha} := \mathrm{dof}_j(m_\alpha), \quad j = 1, \ldots, N_E, \quad \alpha = 1, \ldots, N_P, \tag{34}$$

which yields

$$\boldsymbol{\Pi}_k^{\nabla} = \boldsymbol{D} \boldsymbol{\Pi}_{k*}^{\nabla}. \tag{35}$$

Using the above basis functions and matrix formulation, the elliptic projectors $\boldsymbol{\Pi}_{k*}^{\nabla}$ and $\boldsymbol{\Pi}_k^{\nabla}$ can be calculated by considering Eq. (27) together with the condition Eq. (28) and the definition of the degree of freedom. For a detailed description of the calculation process, see [10,11]. It should be emphasized that the elliptic projection operator plays an important role in the classical VEM. But in this work, this projection is used only as a tool to calculate the L_2 projection $\Pi_{l,E}^0 \nabla$, which will be discussed in the following.

3.2. L_2 projection operator

To define our discrete bilinear form for the stabilization-free VEM, we define the L_2 projection operator $\Pi_{l,E}^0 \nabla$ of the gradient of function in $\mathcal{H}^1(E)$, which is defined as

$$\Pi_{l,E}^0 \nabla : \mathcal{H}^1(E) \to \left[\mathcal{P}_l(E)\right]^2. \tag{36}$$

Based on the above definition, for any given $E \in \mathcal{J}_h$, let $l \in \mathbb{N}$ be given, which depends on the order k and the number of edges n_E. The selection of l is discussed in [36,37] for the first-order stabilization-free VEM as

$$(l+1)(l+2) \geq n_E - 1, \quad k = 1. \tag{37}$$

Besides, a more restrictive bound is given in [38] as

$$2l + 3 > n_E, \quad k = 1. \tag{38}$$

For the higher order method $k > 1$, by considering the element eigenvalue problem for plane elasticity, an inequality between k and the number of edges n_E to ensure the results in a well-posed, stable discrete problem is given in [39] as

$$2l - 2k + 5 \geq n_E. \tag{39}$$

Let $u_h \in \mathcal{H}^1(E)$ satisfying the orthogonality condition

$$\int_E \boldsymbol{p}^T \boldsymbol{\Pi}^0_{l,E} \nabla u_h \, \mathrm{d}\Omega = \int_E \boldsymbol{p}^T \nabla u_h \, \mathrm{d}\Omega, \tag{40}$$

where the right hand can be written as

$$\int_E \boldsymbol{p}^T \nabla u_h \, \mathrm{d}\Omega = \int_{\partial E} \left(\boldsymbol{p}^T \cdot \boldsymbol{n}\right) u_h \mathrm{d}\Gamma - \int_E (\mathrm{div}\boldsymbol{p}) \, u_h \, \mathrm{d}\Omega. \tag{41}$$

Then, Eq. (40) can be written as

$$\int_E \boldsymbol{p}^T \boldsymbol{\Pi}^0_{l,E} \nabla u_h \, \mathrm{d}\Omega = \int_{\partial E} \left(\boldsymbol{p}^T \cdot \boldsymbol{n}\right) u_h \mathrm{d}\Gamma - \int_E (\mathrm{div}\boldsymbol{p}) \, u_h \, \mathrm{d}\Omega, \quad \forall \boldsymbol{p} \in \left[\mathcal{P}_l(E)\right]^2. \tag{42}$$

In Eq. (42), the gradient of variable ∇u and $\boldsymbol{p}$ can be expanded as

$$\nabla u_h = (\boldsymbol{N}^p)^T \hat{\boldsymbol{\epsilon}}, \quad \boldsymbol{p} = (\boldsymbol{N}^p)^T \hat{\boldsymbol{p}}, \tag{43}$$

where $\hat{\square}$ represents a vector, $\epsilon := \nabla u_h$, and $\boldsymbol{N}^p$ contains the basis elements as

$$\boldsymbol{N}^p := \begin{bmatrix} 1 & \xi & \eta & \cdots & \eta^l & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 & \xi & \eta & \cdots & \eta^l \end{bmatrix}^T = \begin{bmatrix} \boldsymbol{m}_l^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{m}_l^T \end{bmatrix}, \tag{44}$$

where $\boldsymbol{m}_l$ is a basic for complete polynomials of order l.

In the next, on expanding u in terms of its basis in $\mathcal{V}_k(E)$, we obtain

$$u_h = \boldsymbol{\phi}^T \tilde{\boldsymbol{u}}, \tag{45}$$

where $\tilde{\boldsymbol{u}} \in \mathbb{R}^{N_E}$. By denoting $\boldsymbol{\Pi}^m$ the matrix representation of the operator $\Pi^0_{l,E}$ with respect to the basis $\boldsymbol{N}^p$, we have

$$\Pi^0_{l,E} \nabla u = (\boldsymbol{N}^p)^T \, \boldsymbol{\Pi}^m \tilde{\boldsymbol{u}}. \tag{46}$$

Substituting Eqs. (45) and (46) into Eq. (42) yields

$$\hat{\boldsymbol{p}}^T \int_E \boldsymbol{N}^p (\boldsymbol{N}^p)^T \, \mathrm{d}\Omega \boldsymbol{\Pi}^m \tilde{\boldsymbol{u}} = \hat{\boldsymbol{p}}^T \int_{\partial E} (\boldsymbol{N}^p \cdot \boldsymbol{n}) \, \boldsymbol{\phi}^T \mathrm{d}\Gamma \tilde{\boldsymbol{u}} - \hat{\boldsymbol{p}}^T \int_E (\mathrm{div}\boldsymbol{N}^p) \, \boldsymbol{\phi}^T \mathrm{d}\Omega \tilde{\boldsymbol{u}}. \tag{47}$$

Since this is true for all $\tilde{\boldsymbol{u}}$ and $\hat{\boldsymbol{p}}$, the above equation Eq. (47) can be written as

$$\int_E \boldsymbol{N}^p (\boldsymbol{N}^p)^T \, \mathrm{d}\Omega \boldsymbol{\Pi}^m = \int_{\partial E} (\boldsymbol{N}^p \cdot \boldsymbol{n}) \, \boldsymbol{\phi}^T \mathrm{d}\Gamma - \int_E (\mathrm{div}\boldsymbol{N}^p) \, \boldsymbol{\phi}^T \mathrm{d}\Omega. \tag{48}$$

Then the projection matrix $\boldsymbol{\Pi}^m$ can be calculated via

$$\boldsymbol{\Pi}^m = \boldsymbol{G}^{-1} \boldsymbol{B}, \tag{49}$$

where

$$\boldsymbol{G} := \int_E \boldsymbol{N}^p (\boldsymbol{N}^p)^T \, \mathrm{d}\Omega, \tag{50}$$

$$\boldsymbol{B} := \int_{\partial E} (\boldsymbol{N}^p \cdot \boldsymbol{n}) \, \boldsymbol{\phi}^T \mathrm{d}\Gamma - \int_E (\mathrm{div}\boldsymbol{N}^p) \, \boldsymbol{\phi}^T \mathrm{d}\Omega. \tag{51}$$

One possibility to compute the integral term $\boldsymbol{G}$ in Eq. (50) is to partition E into triangles (as shown in Fig. 3). Then in each triangle, the Gaussian integral is selected for numerical integration, and the integration order is l^2. Another possibility is the use of a formula obtained for polynomials from the divergence theorem

$$\int_E \xi^p \eta^q \mathrm{d}\Omega = \frac{1}{2} \int_{\partial E} \left[\frac{\xi^{p+1} \eta^q}{p+1} n_x + \frac{\xi^p \eta^{q+1}}{q+1} n_y \right] \mathrm{d}\Gamma, \tag{52}$$

where $\boldsymbol{n} = (n_x, n_y)$ is the outward normal. The first method is selected in this work.

Note that the calculation of matrix $\boldsymbol{B}$ reveals the major difference between the proposed stabilization-free VEM and the conventional VEM. As given in Eq. (51), we record

$$\boldsymbol{B} = \boldsymbol{I}_1 - \boldsymbol{I}_2, \tag{53}$$

where

$$\boldsymbol{I}_1 = \int_{\partial E} (\boldsymbol{N}^p \cdot \boldsymbol{n}) \, \boldsymbol{\phi}^T \mathrm{d}\Gamma, \quad \boldsymbol{I}_2 = \int_E (\mathrm{div}\boldsymbol{N}^p) \, \boldsymbol{\phi}^T \mathrm{d}\Omega. \tag{54}$$

For the first term $\boldsymbol{I}_1$, we have

$$\boldsymbol{I}_1 = \sum_{e \in \partial E} \int_e \boldsymbol{N}^p \boldsymbol{n} \boldsymbol{\phi}^T \mathrm{d}\Gamma = \sum_{e \in \partial E} \frac{|e|}{2} \int_{-1}^{1} \boldsymbol{N}^p \boldsymbol{n} \boldsymbol{\phi}^T \mathrm{d}t, \tag{55}$$

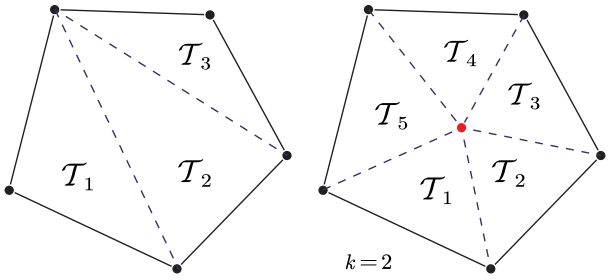

Fig. 3. Triangulation of polygonal element.

where $\boldsymbol{N}^p\boldsymbol{n}\boldsymbol{\phi}^T$ is at most a polynomial of degree $l+k$ and the integral can be calculated by the one-dimensional Gauss quadrature rule.

Generally, it is complex to calculate the integral $\boldsymbol{I}_2$ since the shape function ϕ (with order k) is unknown in the element. To calculate I_2, we need to use the approximation technique mentioned in [35,37–39], We define the $\Pi^0_{k,E}$ be the standard scalar $L^2(E)$ projection from $\mathcal{V}_k(E)$ to $\mathcal{P}_k$ that

$$\left(\Pi^0_{k,E}v, p\right)_E = (v,p)_E \quad \forall v \in \mathcal{V}_k(E), p \in \mathcal{P}_k(E), \tag{56}$$

which can be written as

$$\int_E \boldsymbol{m}\Pi^0_{k,E}\boldsymbol{\phi}^T \mathrm{d}\Omega = \int_E \boldsymbol{m}\boldsymbol{\phi}^T \mathrm{d}\Omega. \tag{57}$$

Defining the $\boldsymbol{\Pi}^0_{k*}$ to be the matrix of $\Pi^0_{k,E}$ on the basis of $m \in \mathcal{M}_k$ and $\boldsymbol{\Pi}^0_k$ being related to the basis of ϕ^T, we have

$$\Pi^0_{k,E}\boldsymbol{\phi}^T = \boldsymbol{\phi}^T\boldsymbol{\Pi}^0_k = \boldsymbol{m}^T\boldsymbol{\Pi}^0_{k*}, \tag{58}$$

where $\boldsymbol{m}$ is the same vector given in Eq. (32). Substituting Eq. (58) into Eq. (57) yields

$$\int_E \boldsymbol{m}\boldsymbol{m}^T \mathrm{d}\Omega\boldsymbol{\Pi}^0_{k*} = \int_E \boldsymbol{m}\boldsymbol{\phi}^T \mathrm{d}\Omega, \tag{59}$$

so that we arrive at

$$\boldsymbol{H}\boldsymbol{\Pi}^0_{k*} = \boldsymbol{M}_\phi, \tag{60}$$

where

$$\boldsymbol{H} = \int_E \boldsymbol{m}\boldsymbol{m}^T \mathrm{d}\Omega, \quad \boldsymbol{M}_\phi = \int_E \boldsymbol{m}\boldsymbol{\phi}^T \mathrm{d}\Omega. \tag{61}$$

The method to compute the integral term $\boldsymbol{H}$ is to partition E into triangles (as shown in Fig. 3). The calculation of matrix $\boldsymbol{M}_\phi$ will be explained in detail later.

Once the matrix $\boldsymbol{\Pi}^0_{k*}$ is determined, by considering the L_2 projection $\Pi^0_{k,E}$ given in Eq. (56) and by using the relationship given in Eq. (58), it follows

$$\begin{aligned}\boldsymbol{I}_2 &= \int_E (\mathrm{div}\boldsymbol{N}^p)\boldsymbol{\phi}^T \mathrm{d}\Omega = \int_E (\mathrm{div}\boldsymbol{N}^p)\left(\Pi^0_{k,E}\boldsymbol{\phi}^T\right)\mathrm{d}\Omega \\ &= \int_E (\mathrm{div}\boldsymbol{N}^p)\boldsymbol{\phi}^T\boldsymbol{\Pi}^0_k \mathrm{d}\Omega = \int_E (\mathrm{div}\boldsymbol{N}^p)\boldsymbol{m}_k^T\boldsymbol{\Pi}^0_{k*}\mathrm{d}\Omega \\ &= \int_E (\mathrm{div}\boldsymbol{N}^p)\boldsymbol{m}_k^T \mathrm{d}\Omega\boldsymbol{\Pi}^0_{k*}.\end{aligned} \tag{62}$$

The numerical integration method of matrix $\boldsymbol{I}_2$ is similar to that of matrix $\boldsymbol{G}$, both in the form of triangulation. Should be mentioned that the kernel function $(\mathrm{div}\boldsymbol{N}^p\boldsymbol{m}_k^T)$ is at most a polynomial of degree $l+k-1$.

Now the key to calculating matrix $\boldsymbol{B}$ becomes the calculation of projection matrix $\boldsymbol{\Pi}^0_{k*}$, which depends on $\boldsymbol{M}_\phi$ given in Eq. (60). Unfortunately, the projection operator is not computable in the current function space. We will introduce the enhancement space and discuss how to calculate the matrix $\boldsymbol{M}_\phi$.

3.3. Enhancement space

Ahmad [12] modified the VEM space $\mathcal{V}$ to a local enhancement space $\mathcal{W}_k(E)$. In that case the operator $\Pi^0_k(E)$ given in Eq. (56) can be easily computed using $\Pi^\nabla_k(E)$ and the local d.o.f.s related to $\mathcal{W}_k(E)$. We should consider the enhanced space

$$\mathcal{W}_k(E) := \left\{w_h \in \tilde{\mathcal{V}}_k(E) : \left(w_h - \Pi^\nabla_k w_h, q\right)_E = 0, q \in \mathcal{M}_k \setminus \mathcal{M}_{k-2}\right\}, \tag{63}$$

where the lifting space is

$$\tilde{\mathcal{V}}_k(E) := \left\{ v \in \mathcal{H}^1(E) : v|_{\partial E} \in \mathcal{B}_k(\partial E), \Delta v \in \mathcal{P}_k(E) \right\}. \tag{64}$$

and Π_k^{∇} is the elliptical projection. It is obvious that in the enhancement space $\mathcal{W}_k(E)$, we have

$$\int_E wq \, \mathrm{d}\Omega = \int_E \Pi_k^{\nabla} wq \, \mathrm{d}\Omega, \quad q \in \mathcal{M}_k \setminus \mathcal{M}_{k-2}. \tag{65}$$

It is easy to find that the dimensional of $\tilde{\mathcal{V}}_k(E)$ is

$$\tilde{N}_V^{E,k} := \dim \tilde{\mathcal{V}}_k(E) = n_E k + \frac{(k+1)(k+2)}{2}, \tag{66}$$

where n_E is the number of edges of the polygon E. The additional degrees of freedom is written as

$$\chi_{\text{add}} = \frac{1}{|E|} \int_E u m_\alpha \mathrm{d}\Omega, \quad \alpha = \frac{k(k-1)}{2} \sim \frac{(k+1)(k+2)}{2}. \tag{67}$$

Based on the additional degrees of freedom, the elliptic projector $\tilde{\Pi}_k^{\nabla}$ in space $\tilde{\mathcal{V}}_k(E)$ can be computed as

$$\int_E vp \mathrm{d}\Omega = \int_E \tilde{\Pi}_k^{\nabla} vp \mathrm{d}\Omega, \quad \forall p \in \mathcal{M}_k \setminus \mathcal{M}_{k-2}, \tag{68}$$

By the moments of order $k-1$ and k, provided in Eq. (68) (actually using the approximation $\chi(\tilde{\Pi}_k^{\nabla}\phi_j) = \chi(\phi_j)$, $\boldsymbol{\phi}$ is basis function of space $\tilde{\mathcal{V}}_k(E)$), the matrix $\boldsymbol{M}_\phi$ in Eq. (60) can be calculated and the matrix $\boldsymbol{H}$ can be obtained as

$$\boldsymbol{H} = \boldsymbol{M}_\phi \boldsymbol{D} \tag{69}$$

Fortunately, for $k = 1$, from Eq. (68), we deduce

$$\int_E vp \mathrm{d}\Omega = \int_E \tilde{\Pi}_k^{\nabla} vp \mathrm{d}\Omega = \int_E \Pi_k^{\nabla} vp \mathrm{d}\Omega, \quad \forall p \in \mathcal{M}_0 \cup \mathcal{M}_1, \tag{70}$$

so based on the definition of L_2 projection it follows

$$\Pi_1^{\nabla} = \Pi_1^0. \tag{71}$$

A similar rule can be found for $k = 2$. So we have for $k = 1$ and $k = 2$,

$$\boldsymbol{\Pi}_k^0 = \boldsymbol{\Pi}_k^{\nabla}, \quad \boldsymbol{\Pi}_{k*}^0 = \boldsymbol{\Pi}_{k*}^{\nabla}, \quad \boldsymbol{\Pi}_k^0 = \boldsymbol{D}\boldsymbol{\Pi}_{k*}^0 \tag{72}$$

In this work, we only consider the cases of $k = 1$ and $k = 2$. Based on the relationship given in Eq. (72), the projection matrix $\boldsymbol{\Pi}_{k*}^0$ used in Eq. (62) can be obtained directly from Eq. (27) together with the condition Eq. (28) (as demonstrated in Section 3.1). This is another reason why we discuss the elliptic projection operator.

4. Stabilization-free virtual element discretization

Based on the above discussion, the L_2 projection operator $\Pi_{l,E}^0 \nabla : \mathcal{H}^1(E) \to \left[\mathcal{P}_l(E)\right]^2$ can be computed so that the gradient of the variable u_h can be approximated as follows

$$\Pi_{l,E}^0 \nabla u_h = (\boldsymbol{N}^p)^T \boldsymbol{\Pi}^m \tilde{\boldsymbol{u}}, \tag{73}$$

where $\boldsymbol{\Pi}^m$ the matrix representation of the operator $\Pi_{l,E}^0$ on the basis $\boldsymbol{N}^p$, see Section 3.2. The projection operator $\Pi_{l,E}^0$ in Eq. (73) is solved about the Laplace operator. As given in [41], the projection has good approximation properties and is suitable also as a displacement space for the problems in solid mechanics. So in the next, the projection operator $\Pi_{l,E}^0$ will be used to solve problems with the hyperelastic material undergoing large deformation. For the first step, we want to discuss the regular total Lagrangian formulation of hyperelastic problems.

4.1. Total Lagrangian formulation of the hyperelastic problem

For hyperelastic problems, the virtual work is expressed as

$$\mathcal{W} = \int_{V_0} \delta \boldsymbol{E} : \boldsymbol{S} \mathrm{d}V_0. \tag{74}$$

The linearization of the virtual work is obtained from

$$\delta \mathcal{W} = \int_{V_0} \delta \boldsymbol{E} : \delta \boldsymbol{S} \, \mathrm{d}V_0 + \int_{V_0} \delta\left(\delta \boldsymbol{E}\right) : \boldsymbol{S} \, \mathrm{d}V_0, \tag{75}$$

where

$$\begin{aligned} \delta\left(\delta \boldsymbol{E}\right) &= \frac{1}{2}\delta\left[\delta\left(\boldsymbol{F}^T \cdot \boldsymbol{F} - \boldsymbol{I}\right)\right] = \frac{1}{2}\delta\left[\delta \boldsymbol{F}^T \cdot \boldsymbol{F} + \boldsymbol{F}^T \cdot \delta \boldsymbol{F}\right] \\ &\doteq \delta \boldsymbol{F}^T \cdot \delta \boldsymbol{F} = \delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}, \end{aligned} \tag{76}$$

and

$$\boldsymbol{D} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{X}}, \quad \boldsymbol{F} = \boldsymbol{I} + \boldsymbol{D}. \tag{77}$$

This leads to the linearization of the virtual work Eq. (75) where

$$\delta \boldsymbol{S} = \frac{\partial \boldsymbol{S}}{\partial \boldsymbol{E}} : \delta \boldsymbol{E} = \mathcal{D} : \delta \boldsymbol{E}, \tag{78}$$

so that

$$\begin{aligned} \delta \mathcal{W} &= \int_{V_0} \delta \boldsymbol{E} : \delta \boldsymbol{S} \mathrm{d}V_0 + \int_{V_0} \boldsymbol{S} : \left(\delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}\right) \mathrm{d}V_0 \\ &= \int_{V_0} \delta \boldsymbol{E} : \mathcal{D} : \delta \boldsymbol{E} \mathrm{d}V_0 + \int_{V_0} \boldsymbol{S} : \left(\delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}\right) \mathrm{d}V_0. \end{aligned} \tag{79}$$

$\mathcal{D}$ is the tangential material tensor which is discussed in Section 2.2 for different hyperelastic models.

4.2. Virtual element discretization

The Green–Lagrange strain tensor is written as

$$\hat{\boldsymbol{E}} = \begin{bmatrix} E_{XX} & E_{YY} & 2E_{XY} \end{bmatrix}^T, \tag{80}$$

in Voigt notation and the second Piola–Kirchhoff stress follows as

$$\hat{\boldsymbol{S}} = \begin{bmatrix} S_{XX} & S_{YY} & S_{XY} \end{bmatrix}^T, \tag{81}$$

then we have in matrix notation

$$\boldsymbol{S} : \delta \boldsymbol{E} = \hat{\boldsymbol{S}}^T \cdot \delta \hat{\boldsymbol{E}} = \delta \hat{\boldsymbol{E}}^T \cdot \hat{\boldsymbol{S}}, \tag{82}$$

where

$$\delta \hat{\boldsymbol{E}} = \begin{bmatrix} F_{11} \frac{\partial \delta u}{\partial X} + F_{21} \frac{\partial \delta v}{\partial X} \\ F_{12} \frac{\partial \delta u}{\partial Y} + F_{22} \frac{\partial \delta v}{\partial Y} \\ F_{11} \frac{\partial \delta u}{\partial Y} + F_{12} \frac{\partial \delta u}{\partial X} + F_{21} \frac{\partial \delta v}{\partial Y} + F_{22} \frac{\partial \delta v}{\partial X} \end{bmatrix}. \tag{83}$$

This is the key to expressing mechanical problems using gradients of variables derived from Poisson's equation. Besides, we have the incremental constitutive relationship

$$\delta \hat{\boldsymbol{S}} = \hat{\boldsymbol{D}} \cdot \delta \hat{\boldsymbol{E}}. \tag{84}$$

Considering Eqs. (73) and (77) yields

$$\boldsymbol{D} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} \\ \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} \end{bmatrix} = \left[\left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m \begin{bmatrix} \tilde{\boldsymbol{u}} & \tilde{\boldsymbol{v}} \end{bmatrix} \right]^T, \quad \boldsymbol{F} = \boldsymbol{I} + \boldsymbol{D}. \tag{85}$$

The Green–Lagrange strain tensor is calculated as

$$\begin{aligned} \delta \hat{\boldsymbol{E}} &= \begin{bmatrix} F_{11} \frac{\partial \delta u}{\partial X} \\ F_{12} \frac{\partial \delta u}{\partial Y} \\ F_{11} \frac{\partial \delta u}{\partial Y} + F_{12} \frac{\partial \delta u}{\partial X} \end{bmatrix} + \begin{bmatrix} F_{21} \frac{\partial \delta v}{\partial X} \\ F_{22} \frac{\partial \delta v}{\partial Y} \\ F_{21} \frac{\partial \delta v}{\partial Y} + F_{22} \frac{\partial \delta v}{\partial X} \end{bmatrix} \\ &= \boldsymbol{A}_1 \left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m \delta \tilde{\boldsymbol{u}} + \boldsymbol{A}_2 \left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m \delta \tilde{\boldsymbol{v}} \\ &= \begin{bmatrix} \boldsymbol{A}_1 \left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m & \boldsymbol{A}_2 \left(\boldsymbol{N}^p\right)^T \boldsymbol{\Pi}^m \end{bmatrix} \begin{Bmatrix} \delta \tilde{\boldsymbol{u}} \\ \delta \tilde{\boldsymbol{v}} \end{Bmatrix} \\ &= \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix} \begin{bmatrix} \left(\boldsymbol{N}^p\right)^T & \\ & \left(\boldsymbol{N}^p\right)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi}^m & \\ & \boldsymbol{\Pi}^m \end{bmatrix} \begin{Bmatrix} \delta \tilde{\boldsymbol{u}} \\ \delta \tilde{\boldsymbol{v}} \end{Bmatrix} \\ &= \boldsymbol{A} \boldsymbol{N}_p^T \boldsymbol{\Pi}_m \begin{Bmatrix} \delta \tilde{\boldsymbol{u}} \\ \delta \tilde{\boldsymbol{v}} \end{Bmatrix}, \end{aligned} \tag{86}$$

where

$$\boldsymbol{A}_1 = \begin{bmatrix} F_{11} & 0 \\ 0 & F_{12} \\ F_{12} & F_{11} \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} F_{21} & 0 \\ 0 & F_{22} \\ F_{22} & F_{21} \end{bmatrix}, \quad \boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}, \tag{87}$$

$$\boldsymbol{N}_p^T = \begin{bmatrix} \left(\boldsymbol{N}^p\right)^T & \\ & \left(\boldsymbol{N}^p\right)^T \end{bmatrix}, \tag{88}$$

and

$$\boldsymbol{\Pi}_m = \begin{bmatrix} \boldsymbol{\Pi}^m & \\ & \boldsymbol{\Pi}^m \end{bmatrix}. \tag{89}$$

Defining vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \begin{bmatrix} \frac{\partial u}{\partial X} & \frac{\partial u}{\partial Y} & \frac{\partial v}{\partial X} & \frac{\partial v}{\partial Y} \end{bmatrix}^T, \tag{90}$$

we obtain

$$\boldsymbol{S} : \left(\delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}\right) = \delta\boldsymbol{\theta}^T \cdot \boldsymbol{\mathcal{I}} \cdot \delta\boldsymbol{\theta}, \tag{91}$$

where

$$\boldsymbol{\mathcal{I}} = \begin{bmatrix} \boldsymbol{S} & \\ & \boldsymbol{S} \end{bmatrix}, \tag{92}$$

for two dimension problems.

Based on Eq. (73), the term $\delta\boldsymbol{\theta}$ is calculated as

$$\delta\boldsymbol{\theta} = \begin{bmatrix} (\boldsymbol{N}^p)^T & \\ & (\boldsymbol{N}^p)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi}^m & \\ & \boldsymbol{\Pi}^m \end{bmatrix} \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} = \boldsymbol{N}_p^T \boldsymbol{\Pi}_m \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} \tag{93}$$

Substituting Eq. (93) into Eq. (91), yields

$$\boldsymbol{S} : \left(\delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}\right) = \begin{Bmatrix} \delta\tilde{\boldsymbol{u}}^T & \delta\tilde{\boldsymbol{v}}^T \end{Bmatrix} \cdot \left(\boldsymbol{\Pi}_m^T \cdot \boldsymbol{N}_p\right) \cdot \boldsymbol{\mathcal{I}} \cdot \left(\boldsymbol{N}_p^T \cdot \boldsymbol{\Pi}_m\right) \cdot \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} \tag{94}$$

4.3. Element stiffness matrix and internal force

For the total Lagrangian formulation, the linearized energy form given in Eq. (79) is considered as

$$\delta\mathcal{W} = \int_{V_0} \delta\boldsymbol{E} : \delta\boldsymbol{S} \mathrm{d}V_0 + \int_{V_0} \boldsymbol{S} : \left(\delta \boldsymbol{D}^T \cdot \delta \boldsymbol{D}\right) \mathrm{d}V_0. \tag{95}$$

Based on Eqs. (82), (84) and Eq. (91), the linearized energy form can be written as

$$\delta\mathcal{W} = \int_{V_0} \delta\hat{\boldsymbol{E}}^T \cdot \hat{\boldsymbol{D}} \cdot \delta\hat{\boldsymbol{E}} \mathrm{d}V_0 + \int_{V_0} \delta\boldsymbol{\theta}^T \cdot \boldsymbol{\mathcal{I}} \cdot \delta\boldsymbol{\theta} \mathrm{d}V_0 \tag{96}$$

Substituting Eqs. (86) and (93) into Eq. (96), yields

$$\begin{aligned} \delta\mathcal{W} = & \int_{V_0} \begin{Bmatrix} \delta\tilde{\boldsymbol{u}}^T & \delta\tilde{\boldsymbol{v}}^T \end{Bmatrix} \cdot \left(\boldsymbol{\Pi}_m^T \cdot \boldsymbol{N}_p \cdot \boldsymbol{A}^T\right) \cdot \hat{\boldsymbol{D}} \cdot \left(\boldsymbol{A} \cdot \boldsymbol{N}_p^T \cdot \boldsymbol{\Pi}_m\right) \cdot \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} \mathrm{d}V_0 \\ & + \int_{V_0} \begin{Bmatrix} \delta\tilde{\boldsymbol{u}}^T & \delta\tilde{\boldsymbol{v}}^T \end{Bmatrix} \cdot \left(\boldsymbol{\Pi}_m^T \cdot \boldsymbol{N}_p\right) \cdot \boldsymbol{\mathcal{I}} \cdot \left(\boldsymbol{N}_p^T \cdot \boldsymbol{\Pi}_m\right) \cdot \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} \mathrm{d}V_0 \\ = & \begin{Bmatrix} \delta\tilde{\boldsymbol{u}}^T & \delta\tilde{\boldsymbol{v}}^T \end{Bmatrix} \cdot \int_{V_0} \left(\boldsymbol{\Pi}_m^T \cdot \boldsymbol{N}_p \cdot \boldsymbol{A}^T\right) \cdot \hat{\boldsymbol{D}} \cdot \left(\boldsymbol{A} \cdot \boldsymbol{N}_p^T \cdot \boldsymbol{\Pi}_m\right) \mathrm{d}V_0 \cdot \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix} \\ & + \begin{Bmatrix} \delta\tilde{\boldsymbol{u}}^T & \delta\tilde{\boldsymbol{v}}^T \end{Bmatrix} \cdot \int_{V_0} \left(\boldsymbol{\Pi}_m^T \cdot \boldsymbol{N}_p\right) \cdot \boldsymbol{\mathcal{I}} \cdot \left(\boldsymbol{N}_p^T \cdot \boldsymbol{\Pi}_m\right) \mathrm{d}V_0 \cdot \begin{Bmatrix} \delta\tilde{\boldsymbol{u}} \\ \delta\tilde{\boldsymbol{v}} \end{Bmatrix}. \end{aligned} \tag{97}$$

Now the tangential element stiffness matrix can be obtained as

$$\boldsymbol{K}_{K2}^t = \boldsymbol{K}_{K2}^{t1} + \boldsymbol{K}_{K2}^{t\sigma}, \tag{98}$$

where

$$\boldsymbol{K}_{K2}^{t1} = \boldsymbol{\Pi}_m^T \cdot \int_{V_0} \boldsymbol{N}_p \cdot \boldsymbol{A}^T \cdot \hat{\boldsymbol{D}} \cdot \boldsymbol{A} \cdot \boldsymbol{N}_p^T \mathrm{d}V_0 \cdot \boldsymbol{\Pi}_m, \tag{99}$$

$$\boldsymbol{K}_{K2}^{t\sigma} = \boldsymbol{\Pi}_m^T \cdot \int_{V_0} \boldsymbol{N}_p \cdot \boldsymbol{\mathcal{I}} \cdot \boldsymbol{N}_p^T \mathrm{d}V_0 \cdot \boldsymbol{\Pi}_m. \tag{100}$$

The internal force follows from Eq. (10). By using the above-defined matrices, we arrive at

$$\boldsymbol{F}_{\mathrm{int}} = \boldsymbol{\Pi}_m^T \cdot \int_{V_0} \boldsymbol{N}_p \cdot \boldsymbol{A}^T \cdot \hat{\boldsymbol{S}} \mathrm{d}V_0. \tag{101}$$

It is easy to find that neither the global stiffness matrix nor the internal force vector requires additional stabilization terms. Then Newton–Raphson iteration is used for the nonlinear equations. In each step, the second Piola–Kirchhoff stress at any integration point is calculated based on the constitutive model (as discussed in Section 2.2). Then, the Cauchy stress $\boldsymbol{\sigma}$ can be calculate as

$$\boldsymbol{\sigma} = \frac{1}{J} \boldsymbol{F} \cdot \boldsymbol{S} \cdot \boldsymbol{F}^T. \tag{102}$$

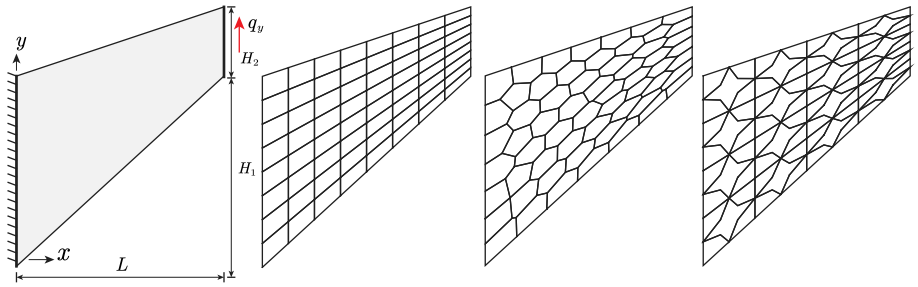

Fig. 4. Initial configuration of Cook's membrane problem with different meshes.

Table 1
Maximum vertical displacement U_y for different element division N for regular elements.

N	FEM		SFVEM			
	Q1	Q2	$k=1, l=1$	$k=1, l=2$	$k=2, l=2$	$k=2, l=3$
2	8.00	10.17	8.00	7.84	10.31	10.24
3	9.53	10.29	9.53	9.51	10.34	10.31
4	10.10	10.34	10.10	10.10	10.36	10.34
5	10.28	10.35	10.28	10.28	10.36	10.36

5. Numerical examples

In this section some numerical examples are calculated by the proposed first- and second-order stabilization-free virtual element method (SFVEM). The example problems undergo finite deformations leading to highly nonlinear responses. All the numerical tests are solved by using the Newton–Raphson algorithm. The neo-Hookean hyperelastic model and constitutive equations are given in Section 2. All the computations are performed with self-written Matlab codes. The polygon meshes can be generated by PolyMesher [42] or converted from triangular meshes. To demonstrate the convergence and accuracy of the stabilization-free VEM, a convergence study is given in Appendix.

5.1. Cook's membrane problem

In this example, we consider the standard Cook's membrane problem of a tapered cantilever beam, which was first proposed in [43]. The structure is fixed at the left side and subjected to a constant distributed vertical load q_y at the right side, as shown in Fig. 4. The relevant dimensions are $L = 48$, $H_1 = 44$, $H_2 = 16$.

The neo-Hookean hyperelastic model is selected with the constitutive parameters chosen as $\mu = 40$ and $\lambda = 100$. Besides, the distributed vertical load is given as $q_y = 4$. Three different meshes including the regular mesh, polygonal mesh generated by PolyMesher, and nonconvex mesh are considered as shown in Fig. 4. Solutions of linear and quadratic finite elements are included for comparison.

In order to achieve the stabilization-free formulation, it is necessary to increase the order l of the internal strain of the element, that is, the order of matrix $\boldsymbol{N}_p$. The relationship between the order l and the number of sides of a polygon can be seen in Eq. (39). Generally speaking, in order to avoid wasting calculations, it is necessary to select the appropriate order l and the corresponding number of integration points according to the number of element corner nodes. In this example, in order to facilitate calculation and comparison, we selected the same polynomial order for different polygon elements. In each triangle used for integration, the number of Gaussian integration points is 6.

To compare the performance of the proposed stabilization-free virtual element method, a convergence study is carried out based on different uniformly refined meshes. We denote four different meshes defined by the parameter N which corresponds to the number of divisions. In this example, we select $N = 2, 3, 4, 5$ which yields an equivalent number of elements $2^N \times 2^N$ for regular mesh and polygonal mesh. The maximal vertical displacement U_y in the y direction of the upper right node of the membrane is computed for meshes corresponding to the element division 2^N for regular and polygonal elements.

The values of maximal vertical displacement U_y are given for different element divisions N in Table 1 for regular mesh, Table 2 for polygon mesh, and Table 3 for nonconvex mesh. The associated convergence study for the maximum displacement U_y is depicted in Fig. 5.

Deformed configurations of the Cook's membrane with the contour plot of stress s_{xy} obtained by SFVEM are shown in Fig. 6, Fig. 7, and Fig. 8 for regular meshes, polygonal meshes, and nonconvex meshes, respectively.

Table 2
Maximum vertical displacement U_y for different element division N for polygon elements.

N	FEM		SFVEM			
	Q1	Q2	$k=1,l=1$	$k=1,l=2$	$k=2,l=2$	$k=2,l=3$
2	8.00	10.17	8.34	7.94	10.31	10.25
3	9.53	10.29	9.28	9.09	10.35	10.30
4	10.10	10.34	10.04	10.00	10.36	10.34
5	10.28	10.35	10.27	10.25	10.37	10.36

Table 3
Maximum vertical displacement U_y for different element division N for nonconvex elements.

N	FEM		SFVEM			
	Q1	Q2	$k=1,l=1$	$k=1,l=2$	$k=2,l=2$	$k=2,l=3$
2	8.00	10.17	7.68	7.26	10.26	10.22
3	9.53	10.29	9.42	9.22	10.33	10.31
4	10.10	10.34	10.08	10.01	10.35	10.34
5	10.28	10.35	10.27	10.25	10.36	10.35

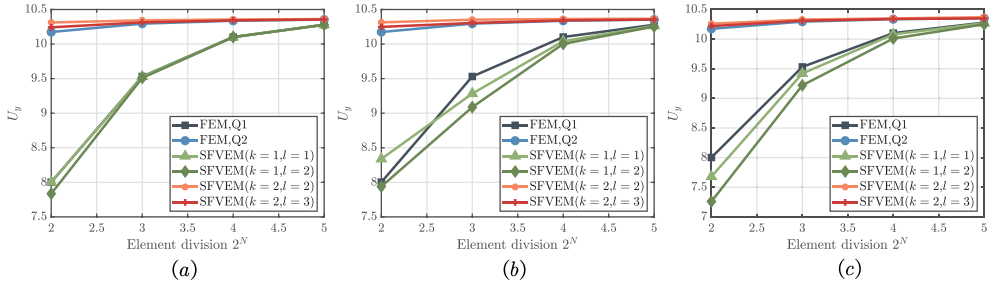

Fig. 5. Max vertical displacement U_y for different element division N, (a) regular elements, (b) polygon elements, (c) nonconvex elements.

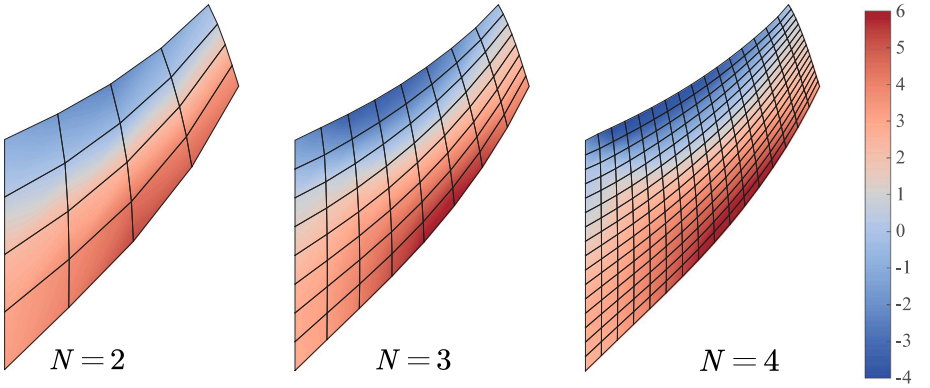

Fig. 6. Deformed shape and contour plot of the stress s_{xy} for different meshes with regular elements (obtained by SFVEM with $k=2,l=3$).

5.2. Punch problem

In this example, a rectangular plate subjected to a vertically distributed uniform load p_y is considered. The problem undergoes severe deformations and was selected as a test to demonstrate the robustness of the stabilization-free virtual element method. The top and the left side of the rectangle are fixed in the horizontal direction, while the bottom side of the rectangle plane is fixed in the vertical direction. The geometric model and dimensions are given in Fig. 9.

In this example, the uniformed load is given as $p_y = 800$ which is applied at the top left half of the plate. Constitutive parameters for the compressible case are given as $\lambda = 400.75$ and $\mu = 92.5$. The model size is given as $L = H = 1$. The problem was calculated using three different meshes and compared with finite element results. The polygonal mesh (generated by the PolyMesher), regular mesh, and nonconvex mesh are depicted in Fig. 10.

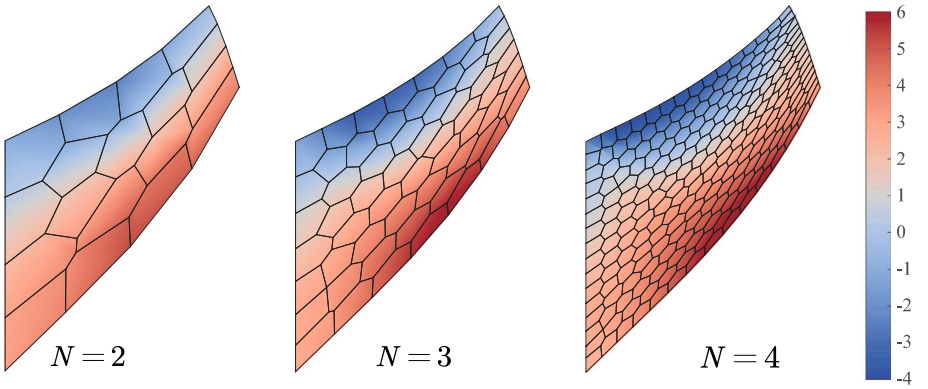

Fig. 7. Deformed shape and contour plot of the stress s_{xy} for different meshes with polygon elements (obtained by SFVEM with $k = 2, l = 3$).

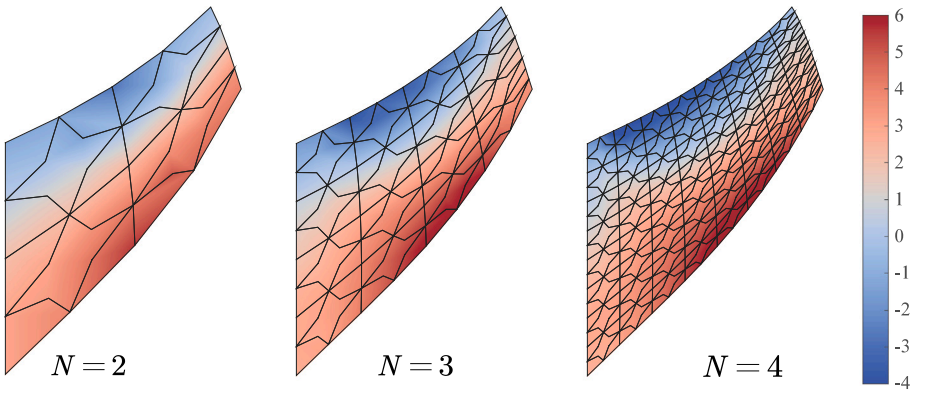

Fig. 8. Deformed shape and contour plot of the stress s_{xy} for different meshes with nonconvex elements (obtained by SFVEM with $k = 2, l = 3$).

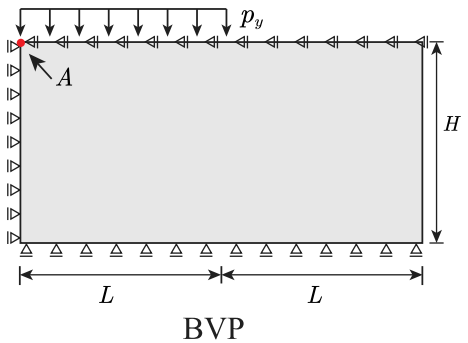

Fig. 9. Schematic diagram and dimensions of the punch problem model.

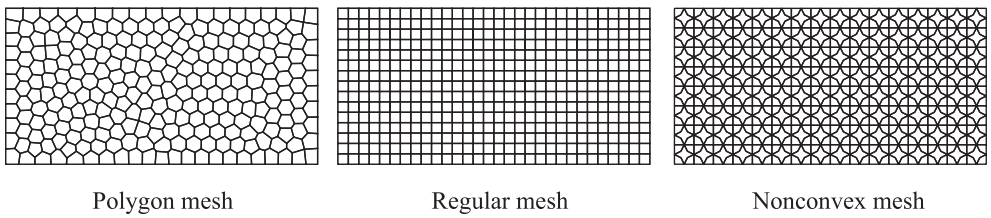

Fig. 10. Two different types of computing meshes.

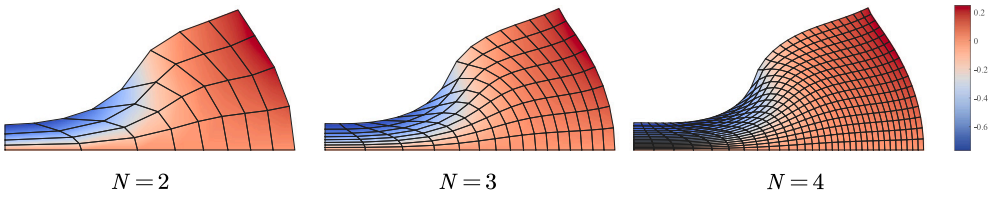

Fig. 11. Deformed shape and contour plot of the vertical displacements for different meshes with regular elements.

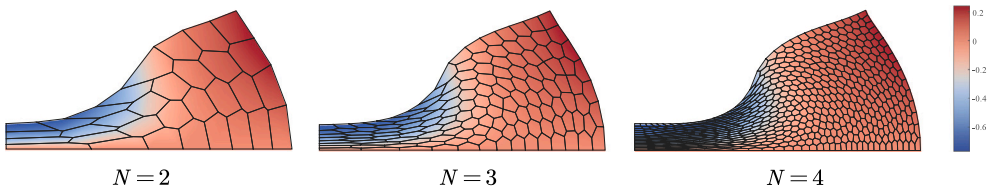

Fig. 12. Deformed shape and contour plot of the vertical displacements for different meshes with polygon elements.

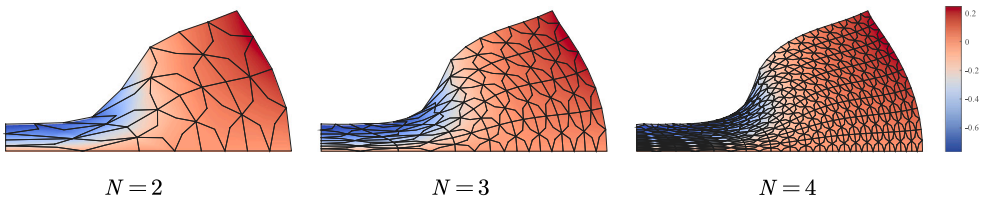

Fig. 13. Deformed shape and contour plot of the vertical displacements for different meshes with nonconvex elements.

Table 4
Vertical displacement U_y of point A for different element division N for different meshes ($l = 2$ for $k = 1$ and $l = 3$ for $k = 2$).

N	FEM		SFVEM(polygon)		SFVEM(regular)		SFVEM(nonconvex)	
	Q1	Q2	$k = 1$	$k = 2$	$k = 1$	$k = 2$	$k = 1$	$k = 2$
2	−0.784	−0.766	−0.779	−0.767	−0.779	−0.766	−0.754	−0.767
3	−0.776	−0.766	−0.778	−0.767	−0.778	−0.765	−0.766	−0.764
4	−0.770	−0.765	−0.768	−0.766	−0.767	−0.765	−0.765	−0.768
5	−0.767	−0.765	−0.766	−0.765	−0.766	−0.765	−0.765	−0.765

Similar to the previous example, a convergence study is carried out. We denote four different meshes by the parameter N which corresponds to the number of divisions. In this example, we select $N = 2, 3, 4, 5$ so that the equivalent number of elements is $2\times 2^N \times 2^N$ for different meshes. For comparison, the finite element method with quadrilateral Q1 and Q2 elements is selected (Q1 element denotes the linear element and Q2 element is the serendipity element). According to the calculation results of example 1, the parameters can be selected as $l = 2$ for $k = 1$ and $l = 3$ for $k = 2$. In each triangle used for integration, the number of Gaussian integration points is 6.

For different types of meshes and different densities, the deformed shape and contour plots of the vertical displacements are given for different meshes with regular elements in Fig. 11, polygonal elements in Fig. 12, and nonconvex elements in Fig. 13, respectively.

The values of the vertical displacements U_y of point A (illustrated in Fig. 9) are given for different element divisions N in Table 4 for regular mesh, polygonal mesh, and nonconvex mesh, respectively. The convergence of the vertical displacement U_y of point A is demonstrated in Fig. 14. For the regular mesh (the regular meshes are the same as the meshes used in FEM), the solution is similar to the solutions obtained by the FEM. This shows that the stabilization-free VEM yields very similar results to finite elements when dealing with regular meshes (quadrilateral elements). Besides, the stabilization-free VEM can use polygonal elements, so it is more suitable for complex geometries. For nonconvex meshes, the stabilization-free VEM can still get accurate results, but the stability is not as good as polygonal (convex) and quadrangular meshes, especially for second formulation.

In the next, we tested the calculation results of first-order FEM and first-order SFVEM under large loads ($q_y = 2560$, quadrilateral elements are selected for FEM and SFVEM ($l = 3$)). As given in Fig. 15, we find that SFVEM can obtain better result under current assumptions. It should be noted that for second-order elements, whether it is FEM or SFVEM, the calculation does not converge.

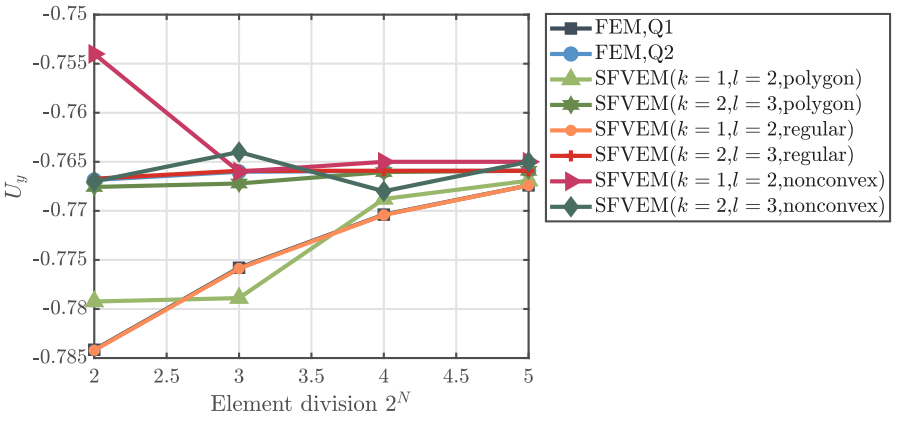

Fig. 14. Vertical displacement U_y of point A for different element division N for different types of elements.

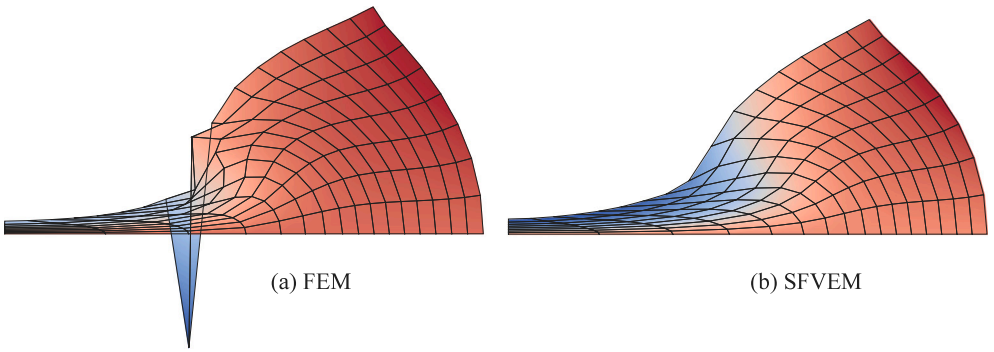

Fig. 15. Deformed shape and contour plot of the vertical displacements for $q_y = 2560$, obtained by FEM and SFVEM for $k = 1$.

6. Conclusion

In this paper, a first- and second-order stabilization-free virtual element method is derived and extended for finite strain applications. The main idea of this method is to modify the virtual element space (enlarged VEM space) to allow the computation of higher-order polynomial approximations of the gradient field. We chose the degree l of vector polynomials $\boldsymbol{N}^p$ for the interpolation of gradient (strain) in each polygon element such that the element stiffness has the correct rank. To construct the matrix form for the elastic and hyper-elastic problems, we start from the bilinear form of Poisson's equation, and obtain an expression for computing the gradient of any variable $\boldsymbol{\Pi}^0_{l,E}\nabla u = (\boldsymbol{N}^p)^T\boldsymbol{\Pi}^m\tilde{\boldsymbol{u}}$. The L_2 projection operator $\boldsymbol{\Pi}^m$ can be computed by considering the enhancement space, which leads to the virtual element discretization. Compared with the conventional VEM, the stiffness matrix is obtained directly without any stable item. The convergence of the proposed method is studied for the Poisson equation and similar results are obtained compared to conventional VEM. Benchmark examples are used to demonstrate the accuracy and flexibility of the SFVEM for hyper-elastic problems. In the future, the discretization scheme can be easily extended to other nonlinear problems since no stabilization term is required.

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.

Acknowledgment

The first and last authors are grateful for the support provided by the Alexander von Humboldt Foundation, Germany.

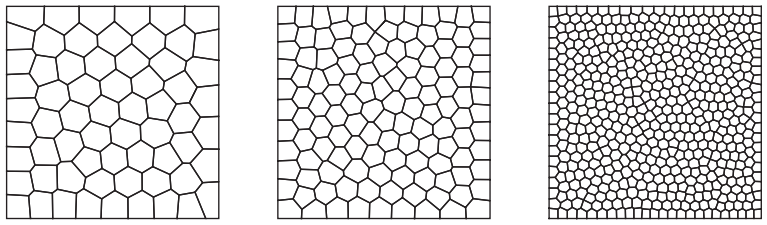

Fig. A.16. Unstructured polygonal meshes with different sizes.

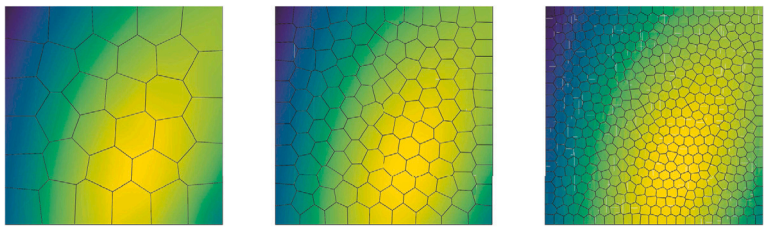

Fig. A.17. Numerical solutions obtained by the stabilization-free VEM with different meshes.

Appendix. Convergence study

In order to show the correction of the derived stabilization scheme and to demonstrate the achieved orders of the new approach, we apply the stabilization-free virtual element method and traditional virtual element method to the Poisson equation. The choice of l (given in Section 3.2 for L_2 projection) is an open question that depends on the number of vertices of element E. In [39], by considering the element eigenvalue problem for plane elasticity, a sufficient inequality is given by

$$2l - 2k + 5 \geq n_E. \tag{A.1}$$

Some necessary mathematics demonstrations and numerical examples have been given in the Refs. [37–39]. Besides, the eigenvalue analysis of individual elements used to check the element stability can be found in [39]. So in this article, we directly use their conclusions without doing cumbersome proofs.

To test convergence, we examine the errors using the L^2 and H^1 norms. The discrete measures are given as

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2} = \sqrt{\sum_{E\in\mathcal{J}} \|\boldsymbol{u} - \boldsymbol{\Pi}_{k,E}\boldsymbol{u}_h\|^2}, \tag{A.2}$$

$$\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1} = \sqrt{\sum_{E\in\mathcal{J}} \|\nabla\boldsymbol{u} - \nabla\boldsymbol{\Pi}_{k,E}\boldsymbol{u}_h\|^2}. \tag{A.3}$$

Let $\Omega = (0,1)^2$, the Poisson equation is considered with the formulation written as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \frac{x^2+y^2}{(xy+1)^2} + 5\sin\left(2x+\frac{1}{2}\right)\cos\left(y+\frac{3}{10}\right), \tag{A.4}$$

and the analytical solution is given by

$$u(x,y) = \sin(2x+0.5)\cos(y+0.3) + \log(1+xy). \tag{A.5}$$

The boundary is fully constrained based on the analytical solution.

In this example, we solve the Poisson equation on five different meshes generated by PolyMesher with size h (as shown in Fig. A.16 for three different meshes). The conventional VEM and stabilization-free VEM (SFVEM) are selected for comparison. In conventional VEM, the stabilization term is needed and the expression is selected as

$$\boldsymbol{K}_s = \left(\boldsymbol{I} - \boldsymbol{\Pi}_k^{\nabla}\right)^T \left(\boldsymbol{I} - \boldsymbol{\Pi}_k^{\nabla}\right). \tag{A.6}$$

In the stabilization-free VEM, the stabilization term is not required. Some contour plots of the numerical solutions obtained by the proposed method are given in Fig. A.17.

Next, we verify and compare the convergence of different types of virtual element methods. As described in the previous part, the choice of l in the stabilization-free VEM should follow the rule given in Eq. (A.1). Hence we compare the solutions obtained by

Table A.5
The L^2 norm for conventional VEM and stabilization-free VEM (SFVEM) for $k = 1$.

h	VEM	SFVEM($l = 1$)	SFVEM($l = 2$)	SFVEM($l = 3$)
1.768e−01	1.001e−02	1.021e−02	1.001e−02	8.130e−03
1.250e−01	4.955e−03	5.058e−03	4.973e−03	3.694e−03
8.839e−02	3.024e−03	3.082e−03	3.038e−03	2.153e−03
6.250e−02	1.371e−03	1.396e−03	1.379e−03	9.491e−04
4.419e−02	6.114e−04	6.223e−04	6.153e−04	4.133e−04

Table A.6
The H^1 norm for conventional VEM and stabilization-free VEM (SFVEM) for $k = 1$.

h	VEM	SFVEM($l = 1$)	SFVEM($l = 2$)	SFVEM($l = 3$)
1.768e−01	1.469e−01	1.469e−01	1.468e−01	1.486e−01
1.250e−01	1.076e−01	1.077e−01	1.076e−01	1.096e−01
8.839e−02	7.678e−02	7.677e−02	7.677e−02	7.746e−02
6.250e−02	5.161e−02	5.161e−02	5.161e−02	5.196e−02
4.419e−02	3.645e−02	3.646e−02	3.645e−02	3.664e−02

Table A.7
The L^2 norm for conventional VEM and stabilization-free VEM (SFVEM) for $k = 2$.

h	VEM	SFVEM($l = 2$)	SFVEM($l = 3$)	SFVEM($l = 4$)
1.768e−01	1.372e−04	1.352e−04	1.365e−04	1.593e−04
1.250e−01	5.048e−05	5.063e−05	5.048e−05	5.689e−05
8.839e−02	1.825e−05	1.815e−05	1.818e−05	1.929e−05
6.250e−02	5.945e−06	6.055e−06	5.945e−06	6.389e−06
4.419e−02	2.102e−06	2.140e−06	2.102e−06	2.232e−06

Table A.8
The H^1 norm for conventional VEM and stabilization-free VEM (SFVEM) for $k = 2$.

h	VEM	SFVEM($l = 2$)	SFVEM($l = 3$)	SFVEM($l = 4$)
1.768e−01	6.433e−03	6.436e−03	6.427e−03	6.737e−03
1.250e−01	3.108e−03	3.113e−03	3.106e−03	3.232e−03
8.839e−02	1.539e−03	1.542e−03	1.537e−03	1.579e−03
6.250e−02	7.633e−04	7.645e−04	7.630e−04	7.831e−04
4.419e−02	3.795e−04	3.801e−04	3.793e−04	3.881e−04

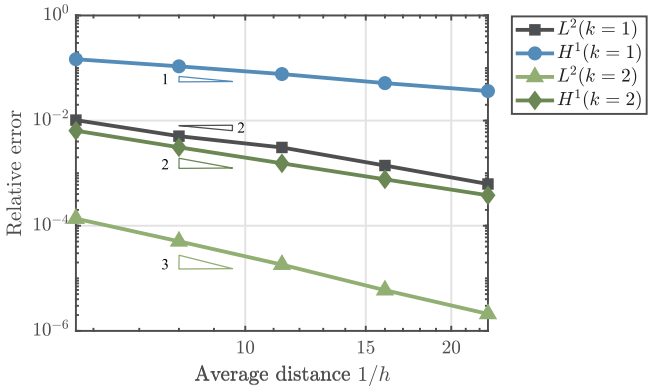

Fig. A.18. Numerical solutions obtained by the stabilization-free VEM with different meshes.

the conventional VEM and the stabilization-free VEM with different parameters l (for $k = 1$, l is selected as $l = 1, 2, 3$ and for $k = 2$, l is selected as $l = 2, 3, 4$ in this example). The L^2 and H^1 norm defined in Eqs. (A.2) and (A.3) are selected with the solutions given in Tables A.5–A.8. The L^2 and H^1 convergence curves of the stabilization-free VEM are depicted in Fig. A.18.

From the above data, it can be observed that the accuracy of the results obtained by SFVEM is very close to the accuracy obtained by traditional VEM (not only the magnitude is the same, but also the convergence rate is basically the same, even if l does not meet the given conditions). This shows that the proposed SFVEM is accurate and leads to similar results as provided by the traditional VEM.

References

[1] 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.

[2] 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.

[3] D.A. Di Pietro, A. Ern, Mathematical Aspects of Discontinuous Galerkin Methods, Springer Berlin, Heidelberg, 2012.

[4] J. Hesthaven, T. Warburton, Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications, vol. 54, 2007.

[5] L. Veiga, K. Lipnikov, G. Manzini, The Mimetic Finite Difference Method for Elliptic Problems, Springer Cham, 2014, http://dx.doi.org/10.1007/978-3-319-02663-3.

[6] F. Brezzi, K. Lipnikov, V. Simoncini, A family of mimetic finite difference methods on polygonal and polyhedral meshes, Math. Models Methods Appl. Sci. 15 (2005) http://dx.doi.org/10.1142/S0218202505000832.

[7] V. Tishkin, A. Samarskii, A. Favorskii, M. Shashkov, Operational finite-difference schemes, Differential Equations 17 (1981) 854–862.

[8] N. Moes, J. Dolbow, T. Belytschko, A finite element method for crack growth without remeshing, Internat. J. Numer. Methods Engrg. 46 (1999) 131–150, http://dx.doi.org/10.1002/(SICI)1097-0207(19990910)46:13.0.CO;2-J.

[9] 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.

[10] 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) http://dx.doi.org/10.1142/S0218202512500492.

[11] 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) http://dx.doi.org/10.1051/m2an/2015067.

[12] 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.

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

[14] 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) http://dx.doi.org/10.1016/j.cma.2014.05.005.

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

[16] 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.

[17] 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) http://dx.doi.org/10.1016/j.cma.2019.02.043.

[18] 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) http://dx.doi.org/10.1016/j.cma.2016.12.020.

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

[20] 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.

[21] 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.

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

[23] 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.

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

[25] 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.

[26] 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.

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

[28] M. Cihan, B. Hudobivnik, F. Aldakheel, P. Wriggers, Virtual element formulation for finite strain elastodynamics, Comput. Model. Eng. Sci. 129 (2021) http://dx.doi.org/10.32604/cmes.2021.016851.

[29] 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.

[30] 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.

[31] O. Čertík, 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) 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) http://dx.doi.org/10.1093/imanum/drac019.

[33] L. Beirão da Veiga, C. Lovadina, A. Russo, Stability analysis for the virtual element method, Math. Models Methods Appl. Sci. 27 (13) (2017) 2557–2594, http://dx.doi.org/10.1142/S021820251750052X.

[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] 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.

[36] 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.

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

[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] 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.
[41] E. Artioli, L. Veiga, F. Dassi, Curvilinear virtual elements for 2D solid mechanics applications, Comput. Methods Appl. Mech. Engrg. 359 (2019) 112667, http://dx.doi.org/10.1016/j.cma.2019.112667.
[42] C. Talischi, G. Paulino, A. Pereira, I. Menezes, PolyMesher: a general-purpose mesh generator for polygonal elements written in MATLAB, Struct. Multidiscip. Optim. 45 (2012) 309–328, http://dx.doi.org/10.1007/s00158-011-0706-z.
[43] R.D. Cook, Improved two-dimensional finite element, J. Struct. Div. 100 (9) (1974) 1851–1863, http://dx.doi.org/10.1061/JSDEAG.0003877.