An engineering perspective to the virtual element method and its interplay with the standard finite element method

Mengolini, Michael; Benedetto, Matías F.; Aragón, Alejandro M.

An engineering perspective to the virtual element method and its interplay with the standard finite element method

Michael Mengolini[a], Matías F. Benedetto[b], Alejandro M. Aragón[a],*

[a] *Faculty of Mechanical, Maritime and Materials Engineering, Delft University of Technology, Mekelweg 2, 2628 CD Delft, The Netherlands*
[b] *INTECIN (UBA - CONICET), Facultad de Ingeniería, Universidad de Buenos Aires, Buenos Aires, Argentina*

Highlights

- A practical exposition of the Virtual Element Method (VEM) for linear elastostatics;
- VEM's advantages and disadvantages are discussed and compared to both h- and p-FEM;
- The method's performance is thoroughly studied for smooth and non-smooth problems;
- VEM's versatility as a helper method to FEM in a hybrid approach is demonstrated;
- Implementation aspects of VEM in displacement-based FEM packages are discussed.

Abstract

In this manuscript we thoroughly study the behavior of the virtual element method (VEM) in the context of two-dimensional linear elasticity problems for an engineering audience versed in standard FEM. Through detailed convergence studies we show the accuracy and the convergence rates recovered by VEM, and we compare them to those obtained by the h- and p-versions of the finite element method (FEM). We also demonstrate the mixability between FEM and VEM; in particular, applications of VEM for coupling non-conforming discretizations and for local refinement are presented, showing higher versatility compared to FEM. Computer implementation aspects in displacement-based finite element codes are thoroughly explained, remarking on the main differences with respect to standard FEM.

1. Introduction

The Virtual Element Method (VEM) is a numerical technique designed for polytopal discretizations (polygons in 2-D or polyhedra in 3-D). Its mathematical foundations have been well established and its benefits have been explored on numerous applications. However, some further insight into the inner-workings of the method is still required for the application-oriented user. The purpose of this paper is to give an engineering perspective to VEM,

* Corresponding author.
E-mail address: a.m.aragon@tudelft.nl (A.M. Aragón).

showing with practical applications not only its features but also the synergic advantages that arise when the method is blended with the standard finite element method (FEM).

Inspired by the ideas put forward by the Mimetic Finite Difference method [1,2], VEM was devised and put into a Galerkin framework [3], ultimately becoming a generalization of standard FEM that allows using arbitrary polytopal meshes. The main idea behind VEM is that the local shape function space in each element is defined implicitly—thus the name *virtual*. Furthermore, shape functions are constructed according to an integer k, which represents the interpolation order of the approximation. At variance with the p-version of the finite element method (p-FEM) [4], however, VEM may contain non-polynomial functions. In spite of its short life span, VEM has already caught the attention of the computational mechanics community; the method has been investigated in the context of a wide number of problems, including small strain [5–7] and finite strain [8] linear elasticity, material inelasticity [9,10], topology optimization [11], plate bending [12], contact [13], geomechanics [14], and fracture mechanics [15]. High-order VEM has been applied to scalar problems and has been developed theoretically for arbitrary polynomial interpolation k [16–18]. In the present context of linear elastostatics, a C++ implementation and description for the lowest order case can be found in [19] and for an arbitrary order formulation see [20].

The method was designed to handle arbitrarily shaped polytopal discretizations, which are particularly convenient for meshing complex geometries. This endows VEM with unmatched flexibility in the pre-processing stage of the analysis. This is by far the most appealing feature of the method, not only because VEM has been shown to be robust to mesh distortion [18,21], but also because non-conforming discretizations in standard FEM—*i.e.*, meshes containing hanging nodes—can be handled in VEM in a straightforward manner; VEM simply interprets a hanging node in a non-conforming mesh as a division of an edge at a 180° angle, effectively increasing by one the number of edges in elements sharing this node. This requires no modification whatsoever to the VEM formulation and the element is treated normally, thereby resulting in a VEM-conforming mesh. Handling non-conforming discretizations gives VEM an edge over standard FEM, since creating conforming discretizations in the latter is a time-consuming process that may even result in inappropriate meshes for the analysis; remeshing may introduce elements with bad aspect ratios that can affect the accuracy of the solution and/or degrade the condition number of the system matrix [22,23].

We believe that the many innovative features of VEM can be buried beneath mathematical formality and bogged down by some of its technical complexity. For this reason, in this manuscript we thoroughly study VEM for linear elastostatics problems in the hopes of making the method more accessible to the FEM practitioner, who is interested in it but is otherwise discouraged by VEM's seemingly steep learning curve. We discuss the method's advantages and disadvantages and compare them to standard FEM. We further compare VEM to p-FEM in the context of high-order interpolations for a *smooth* problem. We provide detailed convergence analyses that show VEM is not only optimally convergent, but also converges faster than p-FEM. We investigate VEM as a procedure to facilitate standard FEM analyses. To this end, a *hybrid* VEM-FEM approach is used to take advantage of VEM's main virtue in dealing with non-conforming discretizations. We finally exploit VEM for an efficient refinement strategy in regions of high stress concentrations, where standard FEM would mandate at best for a complex strategy for handling hanging nodes, or at worst for a complete remeshing.

2. Formulation

2.1. Problem statement

We are interested in solving the standard linear elasticity boundary value problem on a continuous domain $\Omega \subset \mathbb{R}^2$, as shown in Fig. 1a. A body force per unit area $\boldsymbol{f}$ acts on Ω, and its domain boundary $\partial\Omega$ (with unit outward normal $\boldsymbol{n}$) is decomposed into disjoint regions $\partial\Omega_{\mathrm{u}}$ and $\partial\Omega_{\mathrm{t}}$; these are, respectively, regions with prescribed essential (Dirichlet) and natural (Neumann) boundary conditions. For simplicity we assume homogeneous boundary condition $\bar{\boldsymbol{u}} = \boldsymbol{0}$ on $\partial\Omega_{\mathrm{u}}$. An arbitrary traction field $\bar{\boldsymbol{t}}$ is prescribed on $\partial\Omega_{\mathrm{t}}$.

The weak (variational) description of equilibrium is: Find $\boldsymbol{u} \in \mathcal{V}$ such that

$$a\left(\boldsymbol{u}, \boldsymbol{v}\right) = L\left(\boldsymbol{v}\right) \qquad \forall \boldsymbol{v} \in \mathcal{V}, \tag{1}$$

where $\mathcal{V}$ is a vector-valued function space—with components $v_{1,2}$ belonging to the first-order Sobolev space $\mathcal{H}^1(\Omega)$ with null value on Ω_u, *i.e.*, $\mathcal{V} \equiv \left[\mathcal{H}_0^1\left(\Omega\right)\right]^2$—and the bilinear and linear forms are given, respectively, by

$$a\left(\boldsymbol{u}, \boldsymbol{v}\right) = \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{v}) \, \mathrm{d}\Omega, \tag{2}$$
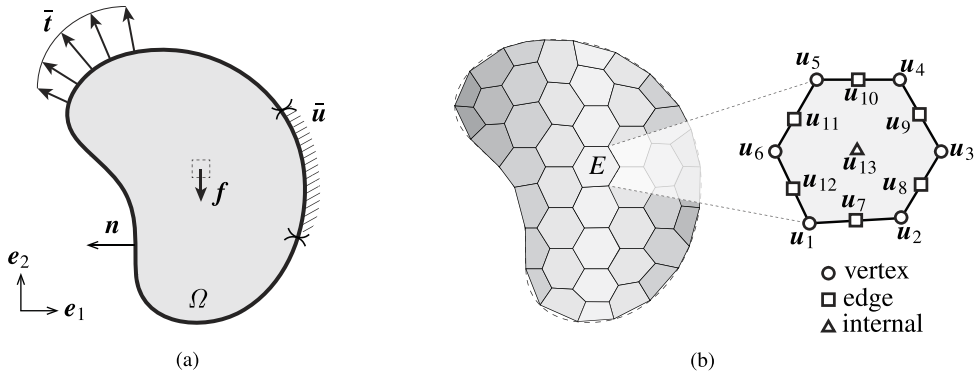
Fig. 1. (a) Schematic of a solid domain Ω with boundary $\partial\Omega$. Essential and natural boundary conditions, $\bar{u}$ and $\bar{t}$ respectively, are prescribed on disjoint portions of the boundary $\partial\Omega_\mathrm{u}$ and $\partial\Omega_\mathrm{t}$. The body force is denoted by f. (b) A polygonal discretization Ω^h showing in the inset an element E with the corresponding degrees of freedom for order $k = 2$.

$$L\left(v\right) = \int_\Omega v \cdot f \, \mathrm{d}\Omega + \int_{\partial\Omega_\mathrm{t}} v \cdot \bar{t} \, \mathrm{d}\partial\Omega. \tag{3}$$

In (2), the Cauchy stress tensor σ follows Hooke's law, *i.e.*, $\sigma = C\epsilon$, where $\epsilon\left(u\right) = \frac{1}{2}\left(\nabla u + \nabla u^\mathsf{T}\right)$ is the linearized strain tensor.

2.2. VEM discrete equations

In VEM, the (Bubnov-)Galerkin or finite-dimensional form of (1) is obtained by choosing an appropriate discrete space $\mathcal{V}^h \subset \mathcal{V}$ for the trial solution u^h and weight function v^h. Unlike in standard FEM, polygonal elements of arbitrary shape can be used—both convex and non-convex—and thus the domain Ω is first partitioned into non-overlapping polygonal regions E such that $\Omega \approx \Omega^h \equiv \mathrm{int}\left(\bigcup_i \overline{E}_i\right)$, as illustrated in Fig. 1b. In order to deal with arbitrarily shaped polygonal elements, VEM uses general finite element spaces that may contain non-polynomial functions. However, as for standard FEM, numerical integration is required for the polynomial component and hence we assume that it is possible to compute their contribution by means of some quadrature procedure. One such technique, although by no means the only possibility, is to subdivide each polygon into triangles and use Gaussian integration. However, an appropriate choice of DOFs could drastically decrease explicit numerical integration, as explained later.

Projection operator. The local projection operator is the most crucial component of VEM. It is designated $\Pi_{E,k} : \mathcal{V}^h(E) \rightarrow \mathcal{P}_k(E)$, where $\mathcal{V}^h(E) \equiv \left[\mathcal{V}^h(E)\right]^2$ indicates the local space of VEM shape functions on E (formally defined in Appendix A.1) and $\mathcal{P}_k(E) \equiv [\mathcal{P}_k(E)]^2$, with $\mathcal{P}_k(E)$ denoting the polynomial space of order less than or equal to k defined on element E. The projector is used to obtain the polynomial projection of VEM shape functions without losing accuracy in the process. This is done because VEM shape functions cannot be used directly—they are not explicitly known in general. Unless stated explicitly, it is implied hereafter that $\Pi \equiv \Pi_{E,k}$ to simplify the notation.

The operator is constructed locally in each polygon so that it satisfies the following orthogonality condition:

$$a_E\left(u^h - \Pi u^h, p\right) = 0, \qquad \forall \, p \in \mathcal{P}_k(E), \tag{4}$$

where integration is performed on E (indicated by the subscript) and zero energy modes (infinitesimal rigid motions) are fixed for uniqueness of the projection. This equation manifests that the error between u^h and Πu^h in E, measured in the energy norm, cannot be captured by the polynomial space $\mathcal{P}_k$ used, *i.e.*, the error in the energy norm is orthogonal to $\mathcal{P}_k$. In VEM lexicon, this property is referred to as *k-consistency*.

Discrete variational form. The discrete bilinear form on an element E for $u^h, v^h \in \mathcal{V}^h$ is given by

$$\hat{a}_E\left(u^h, v^h\right) = a_E\left(\Pi u^h, \Pi v^h\right) + \tau^h S_E\left(u^h, v^h\right), \tag{5}$$

where the first term in the right hand side is known as the consistency part and is no different than what would be obtained by following standard procedures to obtain a finite-dimensional form of (1)—except for the use of the projection operator. The second term, which is scaled by a user-defined parameter τ^h, stabilizes the method. An approximation $\hat{L}_E(v^h) \approx L_E(v^h)$ of the local linear form is needed to compute the contribution of body forces (further explained in Section 2.5). The total contribution to the internal energy and the load term are obtained as

$$\hat{a}\left(u^h, v^h\right) = \sum_{E \in \Omega^h} \hat{a}_E\left(u^h, v^h\right) \quad \text{and} \quad \hat{L}\left(v^h\right) = \sum_{E \in \Omega^h} \hat{L}_E\left(v^h\right), \tag{6}$$

respectively. With these considerations in mind, the discrete form of the problem expressed in abstract form is: Find $u^h \in \mathcal{V}^h$ such that

$$\hat{a}\left(u^h, v^h\right) = \hat{L}\left(v^h\right) \qquad \forall v^h \in \mathcal{V}^h. \tag{7}$$

A stabilization, which computes an approximation of the strain energy not taken into account in the projection, only needs to scale appropriately, *i.e.*, it should be of the same order of magnitude as the consistency term regardless of the mesh. More specifically, the stabilization is required to preserve the coercivity of the problem so that $\hat{a}\left(u^h, v^h\right)$ is bounded below and therefore can be solved (and the solution is unique). In other words, the second term in (5) is needed because functions within each element are only polynomials when dealing with $k = 1$ and simplexes— triangles (tetrahedra) in 2-D (3-D); for $k > 1$, a virtual element has more DOFs for triangles but less or equal for quadrilaterals. In general, for polygonally shaped elements, functions inside them may no longer be polynomials and the stability component controls the strain energy of terms of order greater than k. This, along with the fact the VEM local space contains all polynomials of order up to k, ensures that VEM passes the patch test of order $\leq k - 1$ no matter the shape of elements used. In particular, virtual elements of any order $k > 0$ pass the zeroth order patch test, *i.e.*, the local VEM shape functions are able to represent constant strain states.

2.3. Degrees of freedom (DOFs)

In each polygon of the discretization, a vector-valued test function in the local VEM space, $v^h \in \mathcal{V}^h(E)$, has the following properties:

- components of v^h are continuous functions along the polygon boundary ∂E;
- v^h is a vector with polynomial components of degree k on every edge of e;
- $\Delta v^h\big|_E$ are polynomials of degree $k - 2$ in the interior of E.[1]

Although these properties allow us to recover the behavior of v^h over the boundary ∂E, there is no information about v^h inside E—except for a condition on its Laplacian. In fact, DOFs associated to a polygon are defined in such a way that explicit knowledge about the shape functions is not actually needed—which is emphasized by the *virtual* word in the name of the method. Given a polynomial order k, DOFs associated to a given polygon are (schematically shown in Fig. 1b for $k = 2$):

(*i*) **Vertex DOFs**
  For $k \geq 1$, the value of v^h at the n_v vertices of E (with coordinates $x_1, x_2, \ldots, x_{n_v}$);
(*ii*) **Edge DOFs**
  For $k \geq 2$, the value of v^h at $k - 1$ points along each edge's interior;
(*iii*) **Internal DOFs**
  For $k \geq 2$, the moments up to order $k - 2$ of v^h in E:

$$\frac{1}{|E|} \int_E v^h \cdot p \, dE, \qquad \forall p \in \mathcal{P}_{k-2}(E). \tag{8}$$

For DOFs of type (*i*) and (*ii*), an appropriate choice is the $(k + 1)$-point Gauss-Lobatto quadrature points on each edge as they simplify the computation of integrals. Another choice could be to replace DOFs of type (*ii*) with

---

[1] VEM functions can be thought of as the solution of a PDE inside its corresponding element, by prescribing their value at the element's boundary and solving $\Delta v^h = f$, with f a polynomial. In general, the solution to this problem is not a polynomial and thus the VEM space may contain non-polynomials.
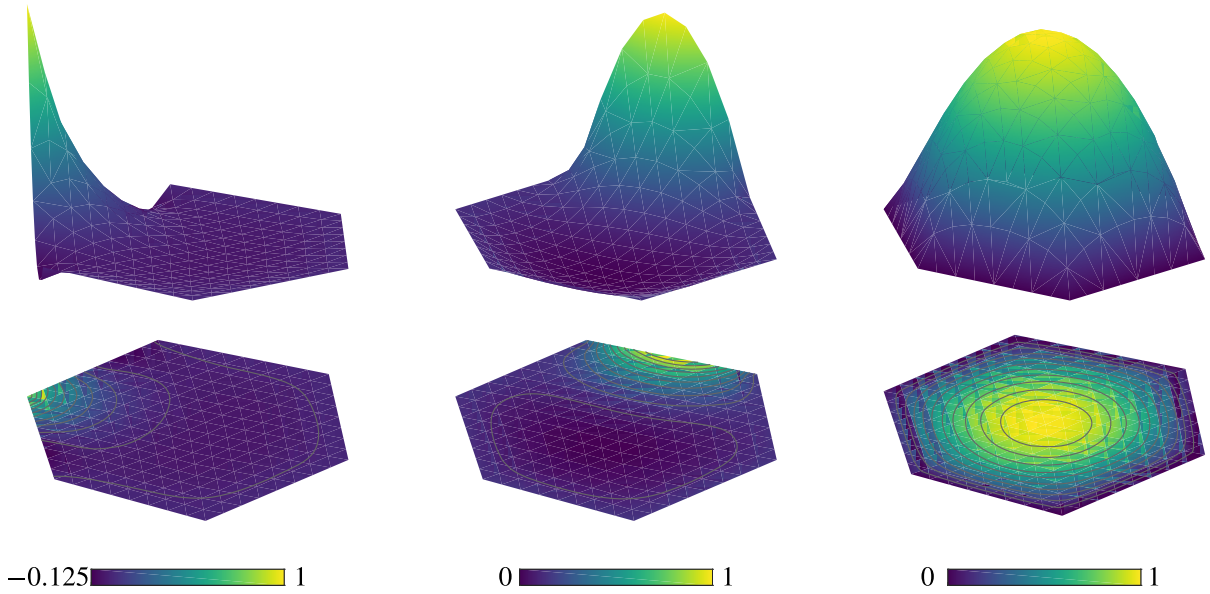
Fig. 2. Visual representation of VEM functions obtained by solving a boundary value problem in a hexagonal domain. Referring back to Fig. 1b, the functions correspond to vertex (a), edge (b), and internal (c) DOFs (either component).

$k - 1$ "moments" on the edge. In fact, any way of univocally determining a polynomial of order k on an edge can be used, although the one presented here closely resembles the choice for standard FEM while the use of moments on an edge is closer to p-FEM. For a given polygon, the total number of DOFs along one spatial dimension is $n_d = \dim(\mathcal{V}^h(E)) = kn_v + k(k - 1)/2$; thus for 2-D elasticity a polygonal element has $2n_d$ DOFs.

At this point it is convenient to define operator dof_i acting on $v_h \in \mathcal{V}^h(E)$, *i.e.*, $\mathrm{dof}_i(v^h)$, $i = 1, \ldots, 2n_d$, as the *value of v^h at the ith degree of freedom*. Basis functions for the local VEM space along one dimension will be denoted by $\{\varphi_i\}_{i=1,\ldots,n_d}$ and are defined so that the Kronecker-δ property holds, *i.e.*, $\mathrm{dof}_i(\varphi_j) = \delta_{ij}$. The vector that collects the shape functions is denoted by $N = [\varphi_1, \ldots, \varphi_{n_d}]$. A graphic representation of functions that associated with each DOF type is presented in Fig. 2, where functions that correspond to edge and vertex DOFs have a null mean value, *i.e.*, their value at the internal DOF is null. The contrary happens to a function associated with an internal DOF, whose mean value is the area of the element and is identically zero on the boundary. These functions are defined implicitly by a local boundary value problem and therefore a numerical approximation was computed for illustration purposes. As we are dealing with problems in 2 dimensions, the bold symbol φ will stand for a vector-valued expression of the basis functions, such that $\varphi_1 = [\varphi_1, 0]$, $\varphi_2 = [0, \varphi_1]$, ..., $\varphi_{2i-1} = [\varphi_i, 0]$, $\varphi_{2i} = [0, \varphi_i]$, ..., $\varphi_{2n_d-1} = [\varphi_{n_d}, 0]$, $\varphi_{2n_d} = [0, \varphi_{n_d}]$. The choice of DOFs univocally defines a polynomial of order k on any edge of the mesh. Therefore, contiguous elements sharing DOFs across a common edge define an inter-element continuous function so that conformity is preserved and the solution is globally C^0-continuous across the domain. For C^1 VEM discretizations see [24,25].

2.4. Local stiffness matrix

In this section we show the construction of the element local stiffness matrix k_E. In standard FEM, it is usually written as

$$k_E = \int_E B^{\mathsf{T}} C B \, \mathrm{d}E,$$ (9)

where C is the material constitutive tensor for either plane stress/strain and $B = \partial N = \begin{bmatrix} \partial \varphi_1 & \partial \varphi_2 & \dots & \partial \varphi_{n_d} \end{bmatrix}$ is the strain–displacement matrix, with ∂ denoting the differential operator (in Voigt notation)

$$\partial = \begin{bmatrix} \partial x_1 & 0 \\ 0 & \partial x_2 \\ \partial x_2 & \partial x_1 \end{bmatrix}.$$

In VEM, because the functions are unknown within the element, constructing the stiffness matrix is possible thanks to the local DOFs and the aforementioned projection operator. The stiffness matrix is obtained by the contributions of *consistency* and *stability* components:

$$k_E = \underbrace{\int_E \left(B\tilde{\Pi} \right)^\mathsf{T} C \left(B\tilde{\Pi} \right) \, \mathrm{d}E}_{k_E^c := \text{consistency}} + \underbrace{k_E^s}_{\text{stability}} \tag{10}$$

where $B\tilde{\Pi} := \partial\, \tilde{\Pi} N = \begin{bmatrix} \partial\tilde{\Pi}\varphi_1 & \partial\tilde{\Pi}\varphi_2 & \dots & \partial\tilde{\Pi}\varphi_{n_d} \end{bmatrix}$, since only an approximation of the shape functions can be recovered using $\tilde{\Pi}$ ($\varphi_i \approx \tilde{\Pi}\varphi_i$). Nevertheless, this approximation will retain optimal convergence properties since the consistency part is exact up to order k, *i.e.*, it can compute the exact strain energy associated with polynomial strains of order up to $k - 1$. While the consistency part has to be computed exactly, its expression differs from that of standard FEM, as explained next.

Consistency term. For a given admissible vector-valued displacement field $v = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^\mathsf{T}$, the (engineering) strains associated with it can be computed using the strain operator:

$$\epsilon(v) = \begin{bmatrix} \partial_{x_1} v_1 \\ \partial_{x_2} v_2 \\ \partial_{x_2} v_1 + \partial_{x_1} v_2 \end{bmatrix}. \tag{11}$$

The projection operator Π, defined by (4), allows the computation of the consistency term of the stiffness matrix by first expressing the projection of a VEM function in the local polynomial base $P_k = \{p_\alpha\}_{\alpha=1,\dots,n_k} \in \mathcal{P}_k(E)$ (see Appendix A.1). To wit,

$$\Pi(\varphi_i) = \sum_{\alpha=1}^{n_k} s_{i,\alpha}\, p_\alpha \in \mathcal{P}_k(E), \qquad s_i \in \mathbb{R}^{n_k}. \tag{12}$$

Next, the component-wise expression can be obtained by using linearity of the strain tensor:

$$
\begin{aligned}
\left(k_E^c \right)_{ij} &= \int_E \epsilon \left(\Pi(\varphi_i) \right)^\mathsf{T} C\, \epsilon \left(\Pi(\varphi_j) \right) \, \mathrm{d}E = \int_E \epsilon \left(\sum_{\alpha=1}^{n_k} s_{i,\alpha} p_\alpha \right)^\mathsf{T} C\, \epsilon \left(\sum_{\beta=1}^{n_k} s_{j,\beta} p_\beta \right) \mathrm{d}E \\
&= \sum_{\alpha=1}^{n_k} \sum_{\beta=1}^{n_k} s_{i,\alpha}\, s_{j,\beta} \int_E \epsilon \left(p_\alpha \right)^\mathsf{T} C\, \epsilon \left(p_\beta \right) \mathrm{d}E = \sum_{\alpha=1}^{n_k} \sum_{\beta=1}^{n_k} s_{i,\alpha}\, s_{j,\beta}\, a_E(p_\alpha, p_\beta) \\
&= \sum_{\alpha=1}^{n_k} \sum_{\beta=1}^{n_k} \tilde{\Pi}_{i\alpha} \tilde{\Pi}_{\beta j} \tilde{G}_{\alpha\beta} = \left[\tilde{\Pi}^\mathsf{T} \tilde{G}\, \tilde{\Pi} \right]_{ij}.
\end{aligned}
\tag{13}
$$

The matrix representation of the projection operator $\tilde{\Pi}$ is given by

$$\tilde{\Pi} = G^{-1}\overline{B} \tag{14}$$

where components of matrices $\overline{B}$ and G are given by

$$\overline{B}_{\beta i} = \int_E \epsilon(p_\beta)^\mathsf{T} C\, \epsilon(\varphi_i) \, \mathrm{d}E, \quad \text{and} \tag{15}$$

$$G_{\beta\alpha} = \int_E \epsilon(p_\beta)^\mathsf{T} C\, \epsilon(p_\alpha) \, \mathrm{d}E, \tag{16}$$

respectively (see Appendices B.1 and B.3). The projection operator given by (14) is crucial to understanding the method; (15) and (16) can both be seen as *internal virtual work* produced on element E by a state of stress $C\epsilon(p_\beta)$

with respect to *virtual strain* fields $\epsilon(\varphi_i)$ and $\epsilon(p_\alpha)$, respectively. Therefore, if we rewrite Eq. (14) as $G\tilde{\Pi} = \overline{B}$, we see that the function of the projection is to ensure the internal virtual work produced by VEM functions is the same as that caused by the polynomial basis chosen.

Finally, the consistency part of the stiffness matrix is computed as

$$k_E^c = \tilde{\Pi}^\mathsf{T} \ \tilde{G} \ \tilde{\Pi}, \tag{17}$$

where $\tilde{G}$ equals G but with the first three rows containing all zeros. In essence, we decompose the VEM shape functions into polynomials, and then use G. To better understand (17), we reformulate it into something more akin to (9) and thus more familiar to standard FEM; replacing $\tilde{G}$ and using the strain operator (11) yields

$$k_E^c = \tilde{\Pi}^\mathsf{T} \int_E \tilde{B}^\mathsf{T} \ C \ \tilde{B} \ \mathrm{d}E \ \tilde{\Pi} = \int_E \left(\tilde{B}\tilde{\Pi} \right)^\mathsf{T} \ C \ \left(\tilde{B}\tilde{\Pi} \right) \ \mathrm{d}E, \tag{18}$$

where $\tilde{B}_{3\times n_k}$ is defined as:

$$\tilde{B} = \begin{bmatrix} 0 & 0 & 0 & \partial p_4 & \partial p_5 & \dots & \partial p_{n_k} \end{bmatrix}. \tag{19}$$

Note that since p_1, p_2 and p_3 are taken as infinitesimal rigid motions (A.3), there is no strain associated with them and thus the first 6 columns of $\tilde{B}$ are zero.

It is important to note that G is computable while $\overline{B}$ is (*a priori*) not, since it involves integration of a VEM shape function inside the element. However, integration by parts allows to compute the internal energy when at least one of the terms is a polynomial, as it is the case here (see Appendix A.2).

Stability term. This term is not present in classical FEM discretizations and requires some preliminaries. The dof operator defined in 2.3 can also be applied to polynomials $p \in \mathcal{P}_k(E)$ since $\mathcal{P}_k(E) \subset \mathcal{V}^h(E)$. The matrix $D_{2n_d \times n_k}$ collects the values of the polynomials p_α at DOFs on E (see (A.1)), and is defined as

$$D_{i\alpha} = \mathrm{dof}_i(p_\alpha) \quad \text{(value of } p_\alpha \text{ at the } i\text{th DOF)}. \tag{20}$$

Bear in mind that the value at the degree of freedom can be a simple evaluation of the function, in the case of point values, or an integration, in cases where a DOF is a moment on a length or an area. Note that for vector valued functions, the DOF operator can refer to different components of the function. The matrix representation of the projection operator acting from $\mathcal{V}^h(E)$ into itself can then be written as

$$\Pi_{ij} = \mathrm{dof}_i\left(\Pi(\varphi_j)\right) = \mathrm{dof}_i\left(\sum_{\alpha=1}^{n_k} s_{i,\alpha} \ p_\alpha\right) = \sum_{\alpha=1}^{n_k} D_{i\alpha}\tilde{\Pi}_{\alpha j} = \left(D\tilde{\Pi}\right)_{ij} \longrightarrow \Pi = D\tilde{\Pi}. \tag{21}$$

The stability term can now be computed as

$$k_E^s = \tau^h \ \mathrm{tr}\left(k_E^c\right)(I - \Pi)^\mathsf{T}(I - \Pi), \tag{22}$$

where, as stated earlier, τ^h is a user-defined parameter that can be taken as $1/2$ for linear elasticity (see [20] for an analysis of this parameter in linear elasticity and [10] for inelastic solids), $\mathrm{tr} \cdot$ denotes the trace operator, and I is the $2n_d \times 2n_d$ identity matrix. An inspection of (22) reveals that the stability term is basically a rough approximation of the internal energy associated with the difference between a VEM shape function and its projection. Namely, the term $(I - \Pi)^\mathsf{T}(I - \Pi)$ takes into account this difference, while $\tau^h \ \mathrm{tr}\left(k_E^c\right)$ *scales* the term with respect to the consistency part, which is the desired property used for proving convergence of the method [3]. Since all polynomials of degree up to order k are included in the local VEM space, these terms in the stabilization are of higher order than the chosen discretization order k. For a similar presentation with explicit matrix construction and arbitrary interpolation order k see [20].

2.5. Local load vector

From (3), the local VEM loading term for a single polygonal element takes the form

$$r_E = \int_E v^h \cdot f \ \mathrm{d}E + \int_{\partial E \cap \Omega_t} v^h \cdot \bar{t} \ \mathrm{d}\partial E + \sum_{i=1} v^h(x_i) \cdot F_i, \tag{23}$$

where in addition to the body force and boundary tractions, we added the contribution of concentrated forces $\boldsymbol{F}_i$ widely used in engineering practice—and which could be mathematically described by means of Dirac delta functions.

The last two terms in (23) do not require any special treatment in VEM and thus follow the standard FEM approach. In particular, tractions on the Neumann boundary can be computed directly as in standard FEM since VEM local shape functions on elements' boundaries are polynomials of degree k and are thus known explicitly. Also, concentrated loads are associated with the displacement DOFs of the location where the load is applied, and since nodal DOFs for VEM and FEM are analogous, so is the handling of concentrated loads. In the case that this load is associated with one of the DOFs of type (iii) in (8), the same procedure as for generalized (non-Lagrangian) DOFs in FEM can be applied. Non-zero prescribed values at the Dirichlet boundary can be dealt with by obtaining the equivalent traction forces that the prescribed displacements generate, as in standard FEM. Alternatively, by using "lift functions" the problem can be solved as homogeneous with an added load term due to the prescribed displacements, which is again standard FEM practice.

The first term in (23) requires special consideration since it involves the VEM shape function in the interior of the element, and therefore the projection operator is needed to express the VEM displacement $\boldsymbol{v}^h$ as a polynomial within E in order to perform the integration. We thus follow a similar procedure to that used for the construction of the stiffness matrix, and we distinguish between three cases based on the accuracy order k. For k > 2, we introduce the (orthogonal) L^2 projection $\Pi_{k-2} : \mathcal{V}_k^h(E) \to \mathcal{P}_{k-2}(E)$ so that the loading term due to body forces can be approximated as

$$\int_E \boldsymbol{v}^h \cdot \boldsymbol{f} \, \mathrm{d}E \approx \int_E \Pi_{k-2}(\boldsymbol{v}^h) \cdot \boldsymbol{f} \, \mathrm{d}E. \tag{24}$$

Note that, although projection is done on polynomials of order $k - 2$, this is enough to retain optimal convergence properties as the approximation of the load vector and that of the discrete solution are of the same order[2]. For $k = 2$ we use the L^2 projection Π_k^0 onto the space of polynomials $\mathcal{P}_k(E)$, so that the loading term becomes

$$\int_E \boldsymbol{v}^h \cdot \boldsymbol{f} \, \mathrm{d}E \approx \int_E \Pi_k^0(\boldsymbol{v}^h) \cdot \boldsymbol{f} \, \mathrm{d}E. \tag{25}$$

Finally, in the case of $k = 1$, the computation is done taking an average of $\boldsymbol{f}$ over the element and using a vertex-based integration rule for $\boldsymbol{v}^h$. More details are given in Appendix C. An alternative procedure can be found in [20], where the projection is taken on the body force $\boldsymbol{f}$.

2.6. Assembly of local arrays

Once that local stiffness matrices and local load vectors are calculated for all elements in the discretization by using (10) and (23), respectively, these are assembled into the global stiffness matrix and load vector as

$$\boldsymbol{K} = \bigwedge_E \boldsymbol{k}_E \qquad \boldsymbol{R} = \bigwedge_E \boldsymbol{r}_E, \tag{26}$$

where $\bigwedge$ denotes the standard finite element assembly operator. As it is commonly done in standard FEM, a system of linear equations $\boldsymbol{KU} = \boldsymbol{R}$ is solved for the global degree of freedom vector $\boldsymbol{U}$ after prescribing boundary conditions. It is worth noting that the condition number of the prescribed stiffness matrix depends on the choice of the polynomial basis used; the monomial basis used here erodes the conditioning as the order k of the interpolation increases. This undesired behavior can be alleviated by choosing an orthogonal polynomial basis [26].

3. Implementation

In this section we discuss, with the aid of pseudo-code, the implementation of VEM in a standard displacement-based FE package. Implementing VEM requires the following modifications or additions:

---

[2] For a discretization of order k, using $\boldsymbol{\Pi}_k$ or $\boldsymbol{\Pi}_{k-1}$ in (24) would yield the same convergence rate as it is limited by the rate of the chosen interpolation order of the virtual elements.

(i) **Input/Output (I/O)** Regarding input, VEM requires discretization formats that are not standard, as meshes can be comprised of polygons of an arbitrary number of vertices—and all polygons need not have the same number of vertices; this implies that the same data structure used for storing elements in standard FEM may not be reused. One way to accommodate for new mesh data is to set a separate data structure for the VEM discretization that uses an associative array (map) where the key is the number of polygon nodes and the value is the corresponding connectivity array (element freedom table). An important remark on polygonal meshes is that, in general, it is not straightforward to produce discretizations whose elements have a specific number of vertices. For instance, meshing procedures based on Voronoi cells usually result in discretizations where most of the elements are hexagons, with some pentagons or quadrangles near the boundaries. Typically, the simplest meshes to produce are those containing elements with 3, 4, or 6 vertices. However, depending on the nature of the problem, elements of different number of vertices may arise and this is where the mesh versatility offered by VEM becomes an asset.

Regarding output, some post-processing is needed to write the results in a format that is commensurate with the use of polygons of arbitrary shape. Many post-processors support the graphical representation of polygonal data by providing them with field data only at vertex locations; inside polygons the field is then interpolated from vertex data. This output type is straightforward to achieve in a VEM implementation: Because in VEM the DOFs associated with vertices correspond to the value of the field, ordering all vertex DOFs first would require minimal post-processing because the solution vector can be simply sliced. This approach may not be sufficient to properly display the field within the polygons, particularly for higher-order interpolations. A better representation of the field can then be obtained by creating an *"output mesh"*, where the field value is not only given at polygon vertices, but also internally by evaluating $\mathbf{\Pi}\boldsymbol{u}^h(\boldsymbol{x})$ at various locations within the polygons. The post-processing here is not trivial and takes some time to produce [27].

(ii) **Local arrays** The other main addition to a standard FEM package lies in the computation of local stiffness matrices and local load vectors. The routine does not only compute the arrays depending on the value of the interpolation order k, but also returns the corresponding DOF array so that the contributions can be readily assembled into the global sparse arrays. Pseudo-code for the assembly routine is given in Algorithm 1.

All other parts of a FEM package, from the data structures used to store the global stiffness matrix and the load vector, to the solver used to obtain the solution vector $\boldsymbol{U}$, need no further modification. However, if a hybrid FEM-VEM code is sought, further modifications are needed. Detecting a non-conforming FEM interface and placing polygons to create a VEM conforming discretization requires a set of geometric operations that could be encapsulated into a *geometric engine*. Standard FEM elements that share nodes along the non-conforming interface could be removed altogether from the discretization, or simply *masked* so they are not taken into consideration in the numerical quadrature of local arrays. The procedure then detects non-conforming nodes in the mesh and embeds them into polygons that replace the removed/masked elements. Noteworthy, no new DOFs are added when dealing with linear interpolations and thus efficiency is not compromised when solving the global system matrix.

It is worth highlighting that there are some additional operations required for computing the stiffness matrix in VEM with respect to FEM. The computation of the projector operator is of course not present in FEM since shape functions are already polynomials; therefore, VEM requires a higher number of computations to obtain the local stiffness matrix. However, many of these added operations are not computationally intensive and numerical quadrature can be significantly reduced with a smart choice of DOFs. For instance, the computation of matrix $\boldsymbol{G}$ can be directly avoided for the computation of the projector. Matrix $\overline{\boldsymbol{B}}$ can be obtained by 1-D integration on element boundaries simplified by the choice of Gauss-Lobatto integration points, and the internal integrations are directly obtained from the DOF definitions. Matrix $\boldsymbol{D}$ requires the evaluation of polynomials at specific locations as well as integration of polynomials over the element, which ultimately requires numerical quadrature (see the appendix for more details). Additionally, the stabilization part involves some matrix operations as well. The load vector also requires the computation of a projector for the body load for $k \geq 2$, while boundary tractions and concentrated loads are handled as in standard FEM. Finally, the assembly process is basically the same, except for the fact that VEM elements of the same order do not necessarily have the same number of DOFs if they have different number of vertices.

Algorithm 1 Computation of local matrices and load vector due to body forces

---

Input: Interpolant order k, polygonal element E with nodal coordinates $X = \begin{bmatrix} x_i & x_j & \ldots \end{bmatrix}^\top$, quadrature point weights γ and coordinates ξ, constitutive matrix C and body force vector f

---

function getLocalArrays($k, E, X, \gamma, \xi, C, f$)

  $\tau^h \leftarrow 1/2$ – *stability parameter*

  $\bar{x}, A, h_E, n_v \leftarrow$ polygonGeometry(X)   – *get centroid, area, diameter and number of vertices*

  $n_d \leftarrow 2 n_v k + k(k-1)$ – *get number of DOFs*

  $n_k \leftarrow (k+1)(k+2)$ – *get dimension of polynomial space*

  $\overline{B}_{\beta i} \leftarrow a_E^h(p_\beta, \varphi_i)$ – *get $\overline{B}_{n_k \times 2n_d}$ matrix* (15) *and Appendix* B.1

  $D_{i\alpha} \leftarrow \text{dof}_i(p_\alpha)$ – *get $D_{2n_d \times n_k}$ matrix* (20) *and Appendix* B.2

  $G \leftarrow \overline{B} D$ – *compute $G_{n_k \times n_k}$ matrix*

  $\tilde{G} \leftarrow \begin{bmatrix} \mathbf{0}_{3 \times n_k} \\ G[3:,:] \end{bmatrix}$ – *compute $\tilde{G}_{n_k \times n_k}$ matrix, equal to G except for kernel part*

  $\tilde{\Pi} \leftarrow G^{-1} \overline{B}$ – *Compute projection operator* (14) *to polynomial space*

  $\Pi \leftarrow D \tilde{\Pi}$ – *Compute projection operator* (21) *to VEM space*

  – *(OPTIONAL) check computation of G*

  $G_{\beta\alpha} \leftarrow a_E^h(p_\beta, p_\alpha)$ – *get $G_{n_k \times n_k}$ matrix* (16) *and (Appendix* B.3)

  verify $G = \overline{B} D$

  – *compute stiffness matrix*

  $k_E^c \leftarrow \tilde{\Pi}^\top \tilde{G} \tilde{\Pi}$ – *Compute consistency part* (17)

  $k_E^s \leftarrow \tau^h \operatorname{tr}(k_E^c)(I_{2n_d \times 2n_d} - \Pi)^\top (I_{2n_d \times 2n_d} - \Pi)$ – *Compute stabilization part* (22)

  $k_E \leftarrow k_E^c + k_E^s$

  – *compute force vector*

  $r_E^f \leftarrow \mathbf{0}_{2n_d \times 1}$ – *Initialize local load vector due to f*

  if $k = 1$ **then**

   | $r_E^f \leftarrow$ computeForce(X, f)   – *compute local load vector due to f* (C.3)

  else if $k = 2$ **then**

   | $H_{\alpha\beta}^0 \leftarrow \int_E p_\alpha \cdot p_\beta \, dE$ – *get $H_{n_k \times n_k}$ matrix* (C.6)

   | $Q_{\alpha i} \leftarrow \int_E p_\alpha \cdot \varphi_i \, dE$ – *get $Q_{n_{k-2} \times 2n_d}$ matrix* (C.19)

   | $Q^0 \leftarrow Q, H^0, G, B$ – *get $Q_{n_k \times 2n_d}^0$ matrix* (C.9)

   | $\Pi_k^0 \leftarrow (H^0)^{-1} Q^0$ – *get Π_k^0 matrix, of size $n_k \times 2n_d$*

   | $r_E^f \leftarrow$ computeForce(X, f, Π_k^0)   – *compute local load vector due to f* (C.10)

  else

   | $H_{\alpha\beta} \leftarrow \int_E p_\alpha \cdot p_\beta \, dE$ – *get $H_{n_{k-2} \times n_{k-2}}$ matrix* (C.13)

   | $Q_{\alpha i} \leftarrow \int_E p_\alpha \cdot \varphi_i \, dE$ – *get $Q_{n_{k-2} \times 2n_d}$ matrix* (C.19)

   | $\Pi_{k-2} \leftarrow (H)^{-1} Q$ – *get Π_{k-2} matrix, of size $n_{k-2} \times 2n_d$*

   | $r_E^f \leftarrow$ computeForce(X, f, Π_{k-2})   – *compute local load vector due to f* (C.17)

  end if

  return k_E, r_E^f

end function

---

4. Results

In this section we showcase VEM with the aid of numerical tests. We study the accuracy of VEM through a series of patch tests, and then investigate VEM both as a standalone method and as a *helper method* to standard FEM for the coupling of non-conforming discretizations.

4.1. Patch tests

Constant strain patch test

We first test whether VEM is able to recover a constant state of strain through a simple patch test, since it is a necessary condition for guaranteeing convergence [28]. The problem, which is schematically shown in Fig. 3, consists of a unit square plate $\Omega = [0, 1] \times [0, 1]$ mm^2 constrained as shown and loaded along its right edge with a traction $\bar{t} = 1$ N/mm e_1. The plate's material has Young modulus $E = 1$ MPa and Poisson ratio $\nu = 0.3$. Plane strain conditions are considered.

Since the exact solution for this problem is a linear displacement field, first order VEM is enough to pass the test. We thus consider $k = 1$ on two discretizations. While in the first all elements are VEM polygons, in the second we further test the mixability between one concave VEM polygon and standard FEM elements—specifically six linear triangular (T3) and four bilinear quadrangular (Q4) elements. The results are shown in Figs. 4a and 4b, respectively, for the discretizations thereof. The figures show, on the deformed configuration, that both discretizations recover
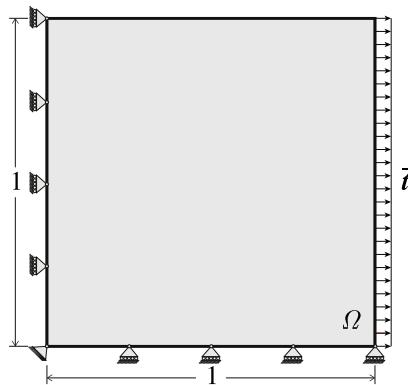

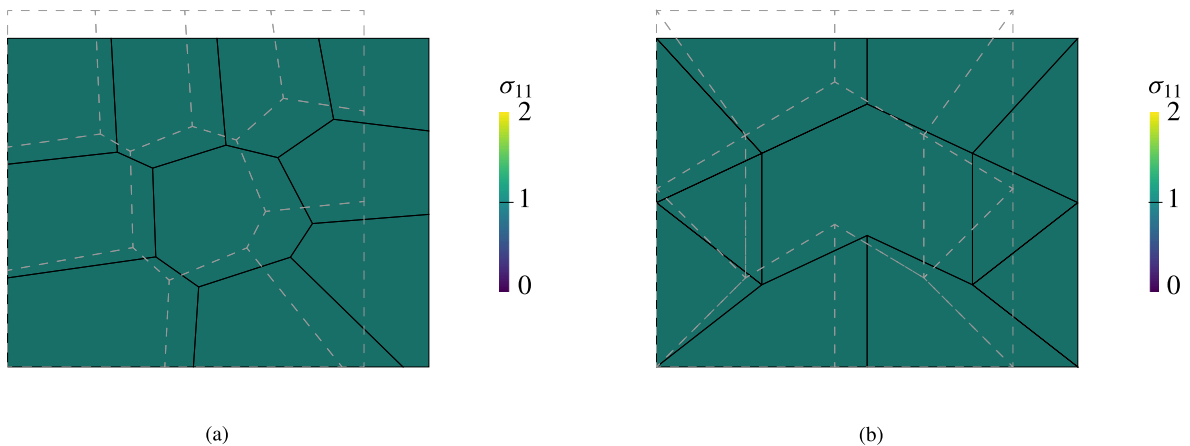**Fig. 3.** Square plate Ω subjected to a lateral traction $\bar{t}$.

Fig. 4. Stress component in the e_1 direction shown over the deformed configuration for VEM (a) and VEM-FEM (b) discretizations, whose original configuration is portrayed with dashed lines.

Fig. 5. Deformed configuration for the high-order patch test, showing in colors the magnitude of the displacement field.

the constant state of stress $\sigma_{11} = 1$ MPa. Consequently, for $k = 1$, VEM polygons and standard FEM elements are perfectly mixable. In fact, T3 elements and triangular VEM elements are equivalent in terms of formulation.

High-order patch test

The traditional patch test studied above aims at recovering a constant state of strain. However, this test does not shed any light into the behavior of VEM for higher-order interpolations. Here we study a high-order patch test, where we manufacture a solution with $k > 1$ and see whether we can capture it with error within numerical precision. In fact, it can be shown that an order k VEM discretization automatically satisfies patch tests of orders $1, 2, \ldots, k - 1$. This is due to the fact that a VEM space of order k contains all polynomials up to degree k. The constant strain patch test was kept for completeness.

Consider the following displacement field:

$$
\mathbf{u} = \begin{bmatrix} x_1 + 3x_1 x_2^2 + x_1 x_2^3 \\ x_1 x_2^3 + \dfrac{x_2^6}{2} \end{bmatrix},
\tag{27}
$$

with a 6th-order polynomial in the u_2 component, whose strain is therefore a 5th-order polynomial. We solve the problem on the square plate of Fig. 3 under plane stress conditions, and material properties $E = 1$ MPa and Poisson's ratio $\nu = 0.2$. A pseudo-body force, obtained as $\mathbf{f} = -\nabla \cdot \boldsymbol{\sigma}(\mathbf{u})$, is also applied.

Fig. 5 shows the numerical solution for this test, where the magnitude of the displacement field is plotted on the deformed configuration. As expected, with $k = 6$ VEM was able to obtain this solution with an error in the L^2 norm $\mathcal{O}(10^{-12})$ and an error in energy norm $\mathcal{O}(10^{-11})$ (these are defined in Section 4.2). Noteworthy, the apparent curvature in the right edge of the square is the result of describing the deformed shape also considering the $k - 1$ internal points for each polygonal element edge in the discretization.

Non-conforming patch test

We now study a patch test where a non-conforming FEM discretization is resolved using VEM polygons. In standard FEM non-conforming discretizations can be resolved by either remeshing locally to create a conforming mesh, or by adding constraint equations to ensure the continuity of the field at hanging nodes. This process is usually not computationally efficient nor robust, since local remeshing could generate distorted elements that may degrade the accuracy of the method. Furthermore, if the mesh sizes at both sides of the non-conforming interface are considerably different, remeshing may no longer be localized to the interface due to the need for transitioning elements on the coarser side.

For this type of applications VEM has a clear edge over standard FEM because hanging nodes can be easily incorporated into VEM polygonal elements; these are only used along the non-conforming interface. As a result, VEM can be seen as a helper procedure for extending the capabilities of standard FEM.

The problem is schematically shown in Fig. 6a, where the domain Ω is discretized by T3 and Q4 elements, respectively, at the left and right of a non-conforming interface Γ. Tractions $\bar{\mathbf{t}} = 1$ N/mm $\mathbf{e}_1$ and $\bar{\mathbf{t}} = 1$ N/mm $\mathbf{e}_2$ are prescribed at the right and top edges of the domain, respectively. We use plane strain conditions, $E = 1$ MPa

(a)   (b)
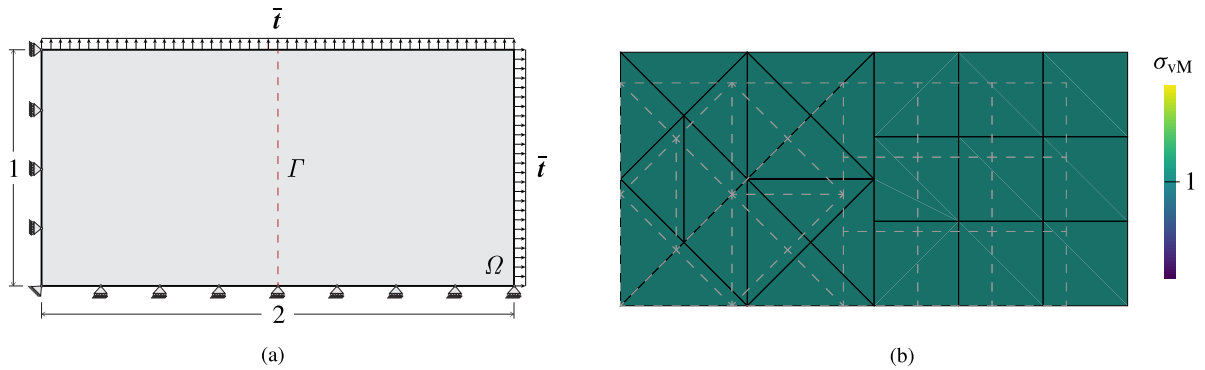
Fig. 6. Non-conforming patch test. (a) Schematic showing a line Γ that is considered as a non-conforming interface in the discretization. von Mises Stress field σ_{vM} (in MPa) plotted on the deformed configuration.

and Poisson's ratio $\nu = 3/8$. The resulting von Mises stress field is shown on the deformed configuration in Fig. 6b, where it can be seen the stress is constant throughout the plate.

4.2. Convergence analysis on a smooth problem

We now focus our attention to the convergence behavior of VEM, *i.e.*, we look at how fast the error with respect to an exact solution decreases as the mesh size $h \to 0$ (*h*-convergence) or as the interpolant order increases for a fixed mesh size h (*p*-convergence). Consequently, for each study, VEM will be compared to the *h*- and *p*-versions of FEM, respectively.

Consider the *smooth* displacement field given by

$$\boldsymbol{u} = \sin(4\pi x_1) \sin(4\pi x_2) (\boldsymbol{e}_1 + \boldsymbol{e}_2), \tag{28}$$

on the unit square domain Ω shown in Fig. 3. Either component of (28) is illustrated in Fig. 7a, and the types of discretizations for VEM and *p*-FEM, respectively, are shown in Figs. 7b–7c. The material constants for this plane stress problem are $E = 1\,\text{MPa}$ and $\nu = 3/8$. Homogeneous essential boundary conditions are prescribed along the entire perimeter of the domain. In addition, a pseudo-body force $\boldsymbol{f} = -\nabla \cdot \boldsymbol{\sigma}(\boldsymbol{u})$ is applied on the entire plate.

The error of our VEM approximations with respect to (28) is quantified through two standard measures, although the projection of the discrete solution is used in the computations (as usually done when working with VEM):

(a)   (b)   (c)

Fig. 7. (a) Either component of the exact solution $\boldsymbol{u}$ given by Eq. (28). (b,c) Discretizations used in *p*-convergence for VEM (b) and *p*-FEM (c).

Fig. 8. L^2 norm (a,b) and energy norm (c,d) of the relative error as a function of total DOFs N_{dof} (a,c) and mesh size h (b,d).

- Relative error in L^2 norm:

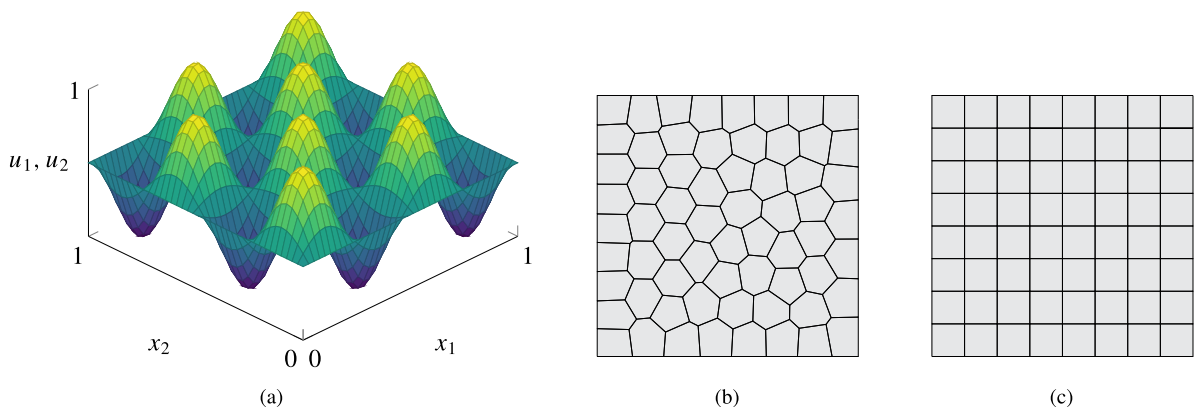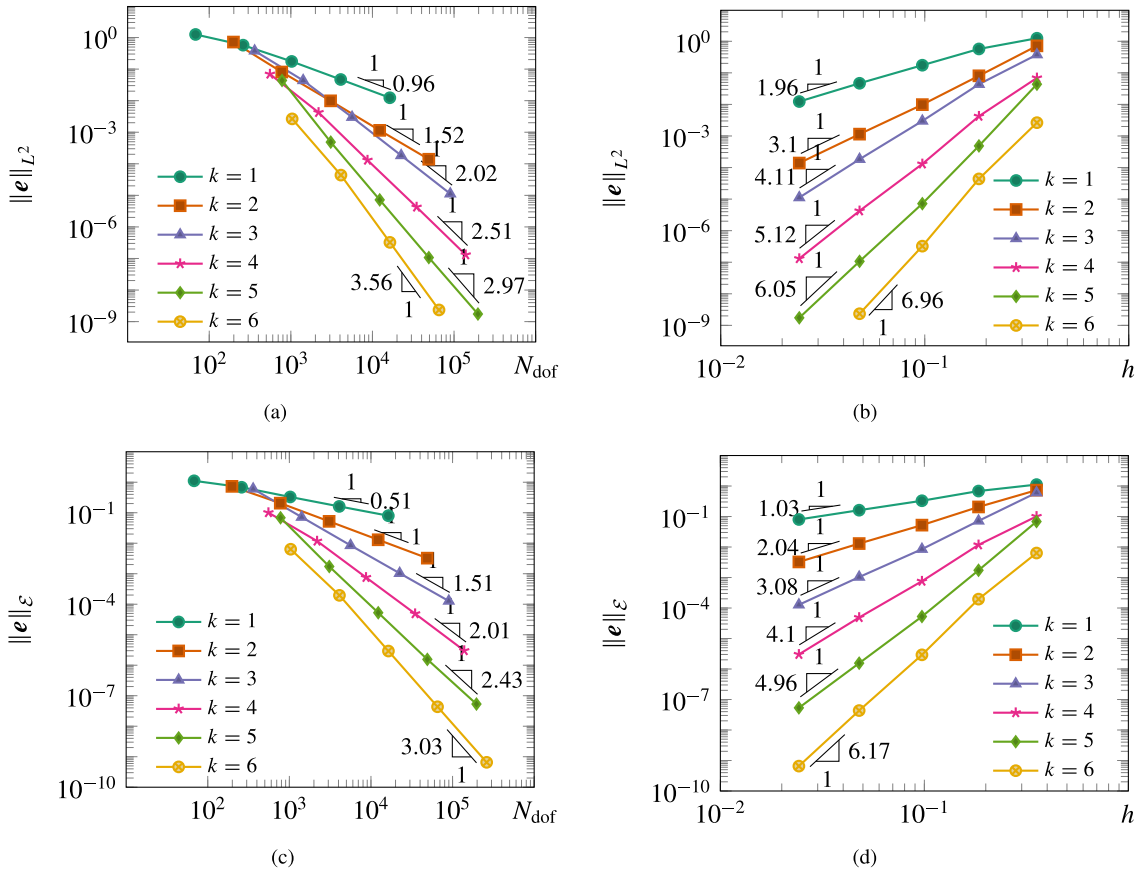$$\|\boldsymbol{e}\|_{L^2} \equiv \frac{\left\|\boldsymbol{u} - \Pi \boldsymbol{u}^h\right\|_{L^2}}{\|\boldsymbol{u}\|_{L^2}} = \sqrt{\frac{\sum_{E \in \Omega^h} \int_E \left(\boldsymbol{u} - \Pi \boldsymbol{u}^h\right)^{\mathsf{T}} \left(\boldsymbol{u} - \Pi \boldsymbol{u}^h\right) \, \mathrm{d}E}{\sum_{E \in \Omega^h} \int_E \|\boldsymbol{u}\|^2 \, \mathrm{d}E}};\tag{29}$$

- Relative error in energy norm:

$$\|\boldsymbol{e}\|_{\mathcal{E}} \equiv \frac{\left\|\boldsymbol{u} - \Pi \boldsymbol{u}^h\right\|_{\mathcal{E}}}{\|\boldsymbol{u}\|_{\mathcal{E}}} = \sqrt{\frac{\sum_{E \in \Omega^h} \int_E \left(\boldsymbol{\epsilon}(\boldsymbol{u}) - \boldsymbol{\epsilon}(\Pi \boldsymbol{u}^h)\right)^{\mathsf{T}} \boldsymbol{C} \left(\boldsymbol{\epsilon}(\boldsymbol{u}) - \boldsymbol{\epsilon}(\Pi \boldsymbol{u}^h)\right) \, \mathrm{d}E}{\sum_{E \in \Omega^h} \int_E \boldsymbol{\epsilon}(\boldsymbol{u})^{\mathsf{T}} \boldsymbol{C} \boldsymbol{\epsilon}(\boldsymbol{u}) \, \mathrm{d}E}}.\tag{30}$$

The h-convergence results, with discretizations composed of 64, 256, 1024, and 4096 polygons, are summarized in Fig. 8; Figs. 8a–8b show the relative error in L^2 norm as per (29) as a function of the total number of DOFs N_{dof} 8(a) and mesh size h 8(b). Similar results are presented in Figs. 8c–8d for the relative error in energy norm given by (30). As apparent from the figures, VEM attains optimal convergence rates for this smooth problem for all interpolation orders. It is worth noting that in order to get these convergence rates, the load vector is computed differently for $k = 1$ and $k = 2$, as explained in Section 2.5.

Figs. 9a–9d show the p-convergence results in L^2 and energy norms; we compare VEM results with those obtained by p-FEM on a structured mesh of Q4 elements. We therefore fix the mesh size h (see Figs. 7b–7c for the discretizations used) and increase the order of the interpolant. The figures show that, similarly to p-FEM, VEM recovers exponential rates of convergence of the form

$$\left\|\boldsymbol{u} - \boldsymbol{u}^h\right\|_{\mathcal{E}} \leq \frac{C}{e^{\gamma(N_{\text{dof}})^{\theta}}} \|\boldsymbol{u}\|_{\mathcal{E}},\tag{31}$$
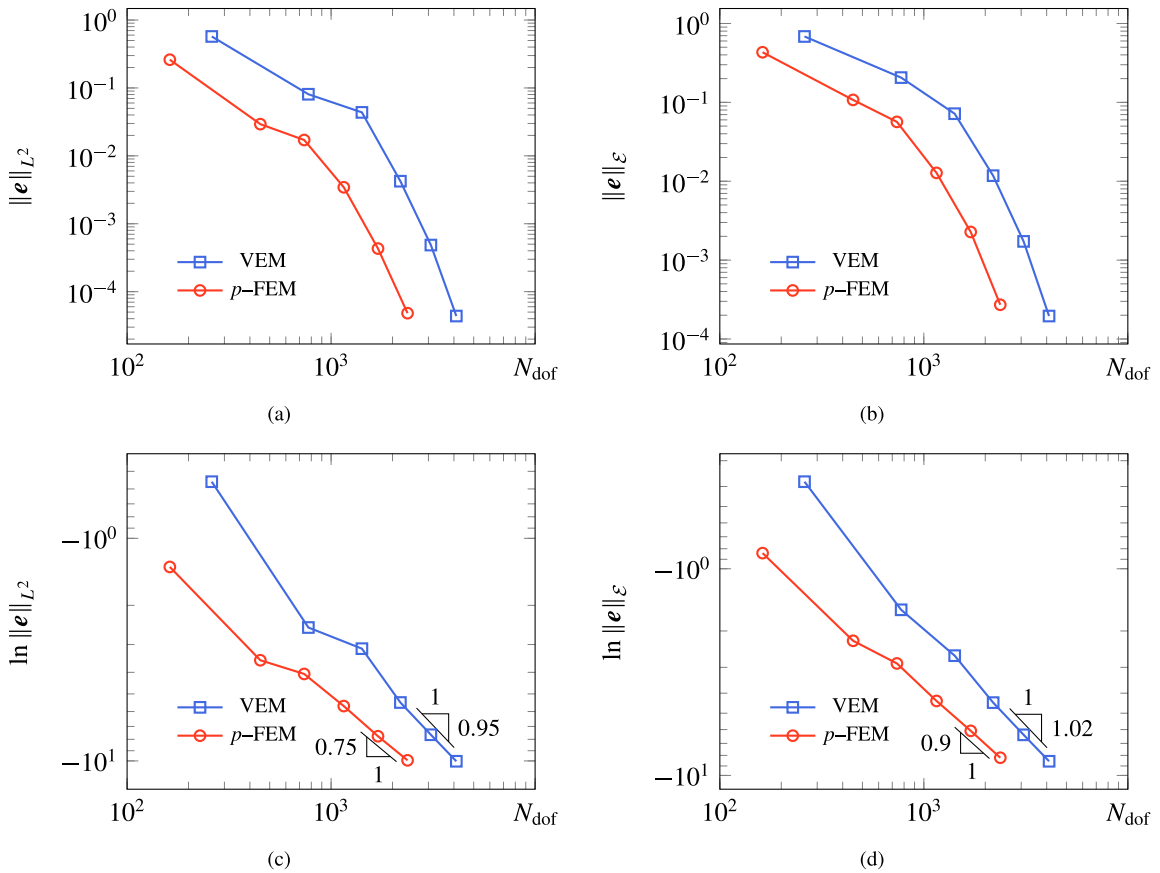
Fig. 9. p-convergence results for the error in L^2 and energy norms, (a) and (b), respectively, as a function of the total number of DOFs N_{dof}. The natural logarithm of the corresponding error is applied again in Figures (c) and (d) to display the exponent θ in Eq. (31).

where C, γ, θ are constants. In Figs. 9c–9d we show the results of applying the natural logarithm twice to (31), resulting in the equation of a line in $\ln N_{\text{dof}} \times \ln(\ln \|e\|)$ space, i.e., $\ln(\ln \|e\|_\mathcal{E}) = C_1 - \theta \ln N_{\text{dof}}$. It is shown that VEM has a steeper slope θ than p-FEM; $\theta = 1.02$ for VEM vs. $\theta = 0.9$ for p-FEM in energy norm (theory predicts $\theta \geq 1/2$).

VEM and p-FEM therefore perform similarly for smooth problems. The difference in accuracy observed is due to the lower accuracy of the polygonal element type used with respect to the number of DOFs. Moreover, the difference in slope reflects the type of approximation spaces used. For instance, VEM on triangles has more DOFs (except for the lowest order), so the slope is expected to be better for p-FEM. On the other hand, on quadrilateral meshes VEM uses less DOFs for $k > 2$ while still retaining the same convergence properties, so the slope in this case should be higher for VEM.

4.3. Stress convergence

In this example we look at the classical problem of an infinite plate with a circular hole of radius $a = 4\,\text{mm}$, as shown in Fig. 10a. The plate is subjected to a uniaxial tension of magnitude $\sigma_0 = 1\,\text{N/mm}$ at infinity. Due to symmetry, we only analyze the portion shown in the inset under plane strain conditions, where $w = 20\,\text{mm}$. The exact traction $\bar{t} = \sigma \cdot n$ is applied to top and right edges. The material of the plate is linearly elastic with $E = 200\,\text{GPa}$ and $\nu = 0.3$.

(a)

(b)   (c)
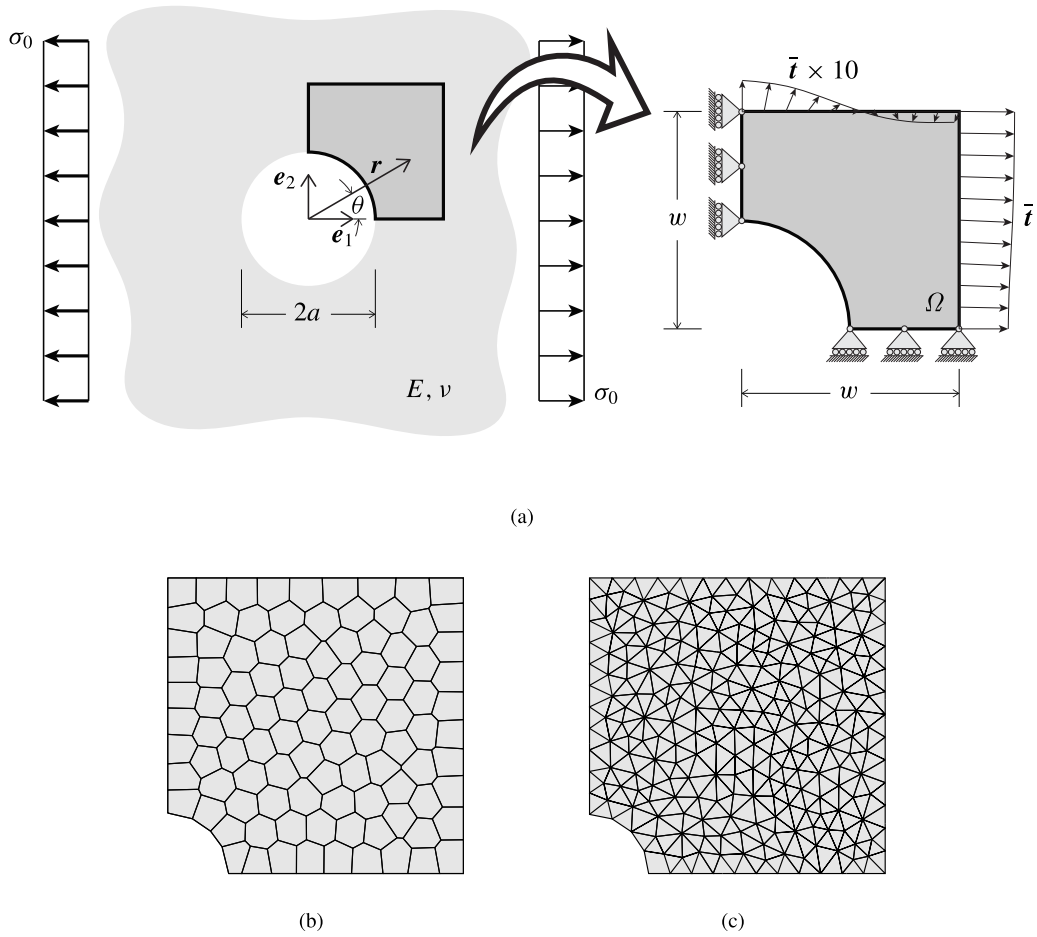
Fig. 10. (a) Schematic for the infinite plate with a circular hole showing in the inset the computational domain chosen due to symmetry; (b,c) Typical VEM and FEM discretizations.

This problem has an analytical solution [29]:

$$
\boldsymbol{u} = \frac{(1+v)a\sigma_0}{2E}
\begin{bmatrix}
2(1-v)(\dfrac{r}{a} + \dfrac{2a}{r})\cos\theta + (\dfrac{a}{r} - \dfrac{a^3}{r^3})\cos 3\theta \\[2mm]
-2(1-2v)\dfrac{a}{r}\sin\theta - 2v\dfrac{r}{a}\sin\theta + (\dfrac{a}{r} - \dfrac{a^3}{r^3})\sin 3\theta
\end{bmatrix},
\tag{32}
$$

with associated stress field

$$
\boldsymbol{\sigma} =
\begin{bmatrix}
\sigma_{11} \\ \sigma_{22} \\ \sigma_{12}
\end{bmatrix}
= \sigma_0
\begin{bmatrix}
1 - \dfrac{a^2}{r^2}(\dfrac{3}{2}\cos 2\theta + \cos 4\theta) + \dfrac{3a^4}{2r^4}\cos 4\theta \\[2mm]
-\dfrac{a^2}{r^2}(\dfrac{1}{2}\cos 2\theta - \cos 4\theta) - \dfrac{3a^4}{2r^4}\cos 4\theta \\[2mm]
-\dfrac{a^2}{r^2}(\dfrac{1}{2}\sin 2\theta + \sin 4\theta) + \dfrac{3a^4}{2r^4}\sin 4\theta
\end{bmatrix}.
\tag{33}
$$

In this problem we study the convergence in stress using the L^2 norm of the stress error:

$$
\|\boldsymbol{e}_\sigma\|_{L^2} \equiv \frac{\|\boldsymbol{\sigma} - \boldsymbol{\sigma}^h\|_{L^2}}{\|\boldsymbol{\sigma}\|_{L^2}} = \sqrt{\frac{\sum_{E\in\Omega^h}\int_E (\boldsymbol{\sigma} - \boldsymbol{\sigma}^h)^\mathsf{T}(\boldsymbol{\sigma} - \boldsymbol{\sigma}^h)\,\mathrm{d}E}{\sum_{E\in\Omega^h}\int_E \|\boldsymbol{\sigma}\|^2\,\mathrm{d}E}},
\tag{34}
$$

Fig. 11. Stress convergence curves for the infinite plate with a hole under uniaxial tension and VEM discretizations with $k = 1$. The error in stresses is measured in L^2 norm by Eq. (34).

which is homologous to the error measure defined in (30), and the stresses are computed from the projection of the discrete solution as $\sigma^h = C\epsilon(\Pi u^h)$. The h-convergence is performed using meshes with $100, 400, 1600$ and 6400 polygonal elements. We also compared the VEM solutions with standard FEM using constant strain triangular elements; uniform meshes composed of $167, 492, 1869$ and 7497 T3 elements were used. Typical VEM and FEM discretizations used in this example are shown in Figs. 10b and 10c, respectively.

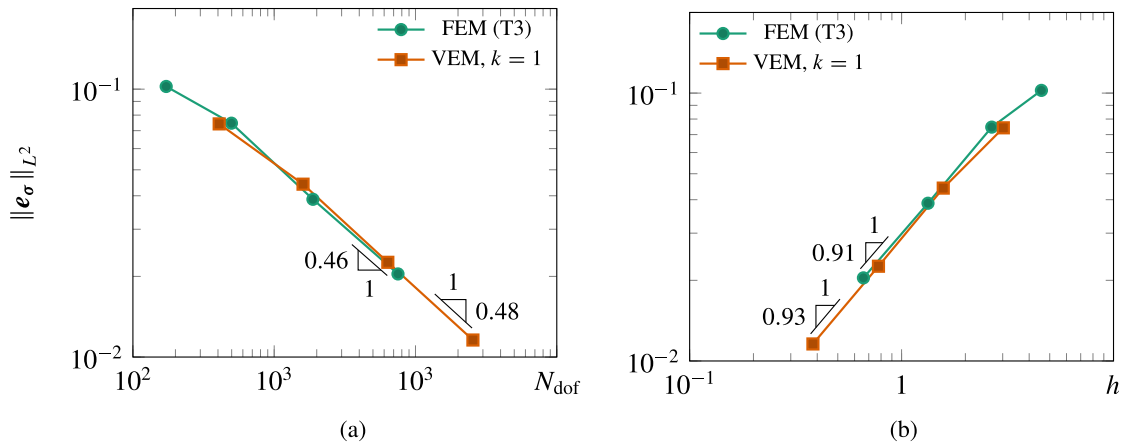The convergence results are summarized in Fig. 11, where we appreciate similar convergence behavior in stress for both VEM and FEM. Because of the stress concentration at the hole, better predictions could be obtained by using *geometric meshes* [30], where the mesh size is reduced in geometric progression towards the stress concentration. Later in this section we will explore an alternative approach where VEM is used to study stresses in regions with singularities.

4.4. Coupling non-matching meshes

Here we build on the non-conforming patch test of Section 4.1 and show how standard FEM can be coupled to VEM for solving a complex non-conforming mesh; as explained earlier, VEM polygons will be constructed only where a non-conforming node is detected.

We revisit the example of Fig. 10a for the 2-D stress analysis of an infinite plate containing a circular hole loaded in uniaxial tension. We choose $x_2 = 0$ to place our non-conforming interface and again we take advantage of symmetry to model half of the problem. The discretization is composed of T3 elements for $x_2 \geq 0$ and Q4 elements for $x_2 \leq 0$. We prescribe the exact value of the traction field at the domain's top, bottom, and right edges.

The results are shown in Fig. 12, where the stress in the e_1 direction is shown on the deformed configuration. A stress concentration factor $K_t = 2.98$ is obtained, which approaches the well known theoretical value $K_t = 3$ obtained by Kirsch [31]. The stress concentration value obtained is, of course, closer to the theoretical prediction in the upper mesh due to the use of finer elements. In the next example we use this fact to improve the approximation properties close to a singularity.

4.5. VEM for localized mesh refinement

This section deals with the solution of the Flamant problem [32,33], a half-space loaded by a downward vertical force, as illustrated in Fig. 13a. Its interest lies in studying stresses in the neighborhood of the applied load considering that there is a singularity at the origin. This problem has applications in foundation design, where stresses in the soil (considered as a semi-infinite elastic medium) are sought. The exact solution for the displacement
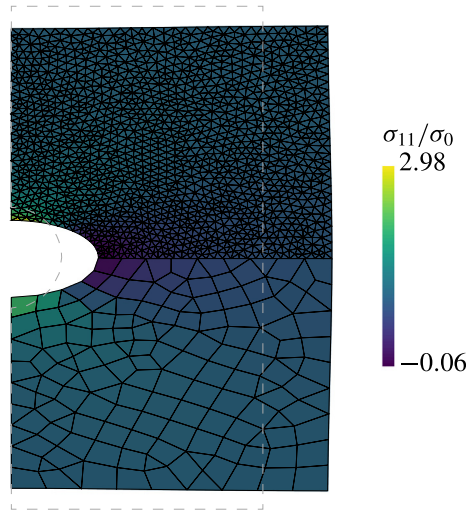
Fig. 12. Stress plot for the plate with a hole under unidirectional tension σ_0 at $x_1 = \infty$. In this example VEM polygons are placed to resolve the non-conforming interface between the triangular and quadrangular meshes. The figure shows the normalized element stress (average value per element) in the $\boldsymbol{e}_1$ direction, directly calculated from the finite element solution. The maximum value 2.98 approximates the well-known value of 3.

(a)   (b)
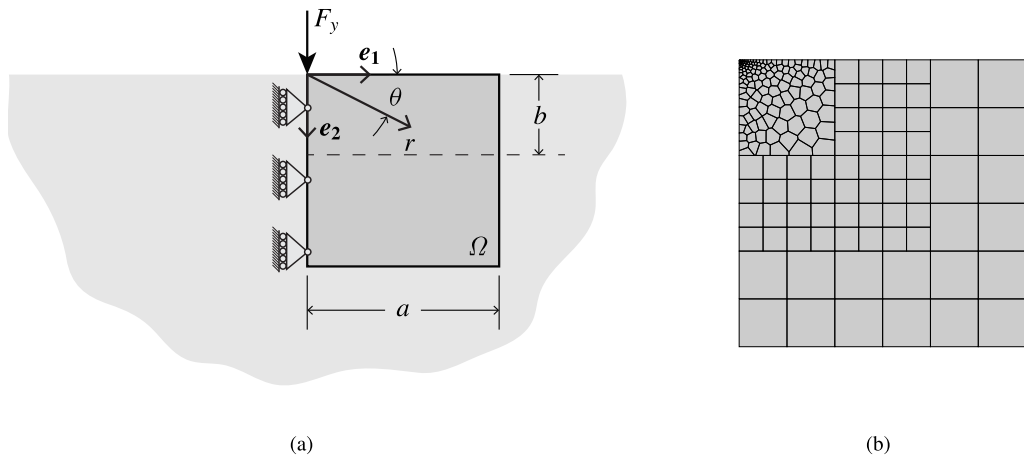
Fig. 13. (a) Schematic for the Flamant problem, a half-space loaded with a point force as shown; (b) VEM discretization.

field in the radial and angular directions is given by:

$$
\boldsymbol{u}\,(r,\theta) = \frac{-F_y}{4\pi G}
\begin{bmatrix}
(\kappa - 1)\,\theta - \dfrac{2x_1 x_2}{r^2} \\[2mm]
(\kappa + 1)\log r - \dfrac{2x_2^2}{r^2}
\end{bmatrix},
\qquad
\kappa =
\begin{cases}
3 - 4\nu & \text{for plane strain,} \\[2mm]
\dfrac{3 - \nu}{1 + \nu} & \text{for plane stress,}
\end{cases}
\tag{35}
$$

where F_y is the magnitude of the applied point load, G is the shear modulus, and $r = \sqrt{x_1^2 + x_2^2}$, $\theta = \arctan(x_2/x_1)$ are polar coordinates with origin located at the load's point of application. The exact stress field is given in polar coordinates by

$$
\boldsymbol{\sigma}\,(r,\theta) = \begin{bmatrix} \sigma_r & \sigma_\theta & \sigma_{r\theta} \end{bmatrix}^{\mathsf{T}} = \begin{bmatrix} -\dfrac{2F_y}{\pi r}\sin\theta & 0 & 0 \end{bmatrix}^{\mathsf{T}},
\tag{36}
$$
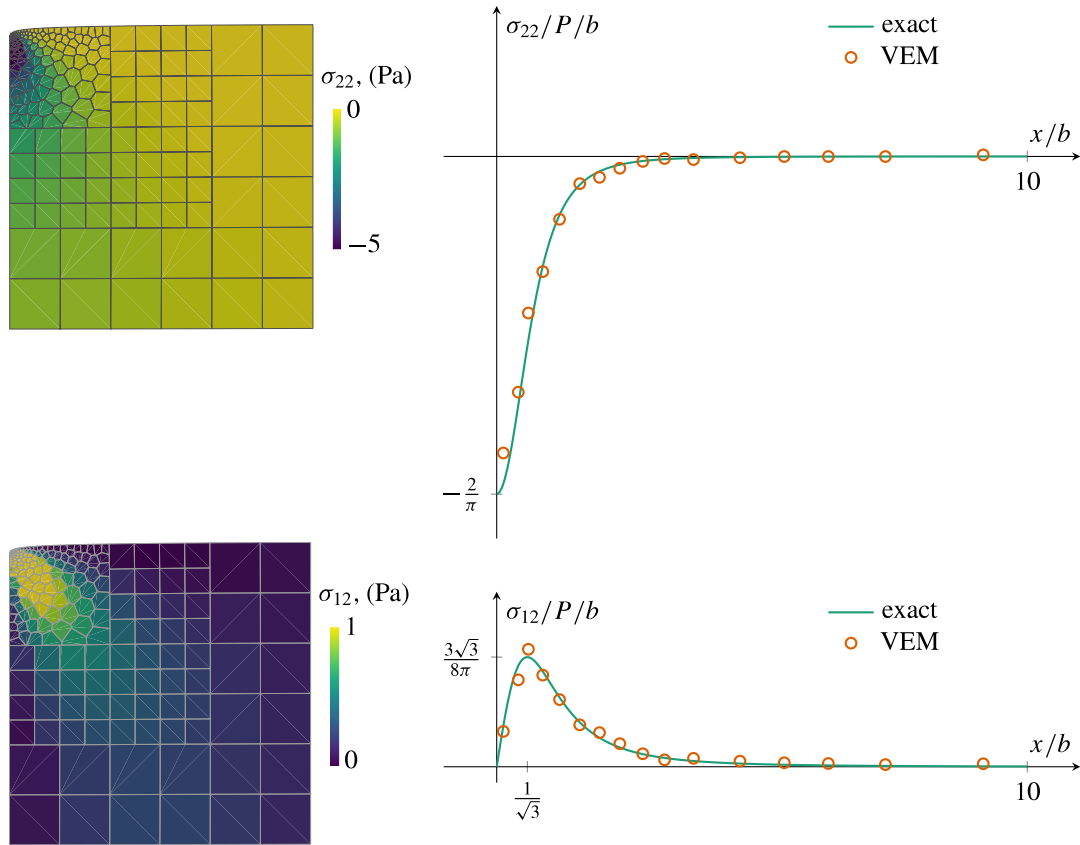
Fig. 14. Comparison between the exact stresses and those resulting from the VEM approximation. The stress components σ_{22} (top) and σ_{12} (bottom) were measured along a horizontal line located at $x_2 = 0.1$ m below the applied load.

or in Cartesian coordinates by

$$\boldsymbol{\sigma}\,(x_1, x_2) = \begin{bmatrix} \sigma_{11} & \sigma_{22} & \sigma_{12} \end{bmatrix}^{\mathsf{T}} = \frac{-2F_y}{\pi\,(x_1^2 + x_2^2)^2}\begin{bmatrix} x_1^2 x_2 & x_2^3 & x_1 x_2^2 \end{bmatrix}^{\mathsf{T}}. \tag{37}$$

Inspecting (35) reveals that, for a fixed θ, displacements are directly proportional to $\log(r)$—and are therefore undefined at $r = 0$—but also that displacements are unbounded for $r \to \infty$. Finite element solutions of this problem show that indeed $\|\boldsymbol{u}\,(\mathbf{0})\|$ becomes increasingly large as $h \to 0$. Furthermore, by simply integrating (36) in a neighborhood of the applied load we can infer that the solution to this problem does not have finite strain energy (the H^0-norm of the stress is analogous to the energy norm); in fact, it can be easily shown that the solution is not in H^1 but in $H^{1-\varepsilon}$, with $\varepsilon > 0$.

Despite all that, the finite element solution for this problem is well behaved if we exclude the element that contains the load, and we will investigate the stresses obtained by our VEM approximation. The computational domain consists of the square region shown in Fig. 13 with $a = 1$ m, so we take advantage of symmetry. At the bottom and right sides of the domain we prescribe the displacement field given by (35). We will then measure stresses at a horizontal line located at $x_2 = 0.1$ m (distance marked b in Fig. 13a). By taking full advantage of VEM's ability to handle arbitrary discretizations, a local refinement scheme can be devised to increase the number of elements only in the area of interest, in this case close to where the load is applied. The discretization we used is shown in Fig. 13b, where increasingly smaller polygons are placed close to the applied load.

The results are then summarized in Fig. 14, where the vertical stress σ_{22} and the shear stress σ_{12} obtained by the VEM approximation are compared to the exact values given by (37). VEM stresses were averaged in each element, as shown by the contour plots in the figure. Because stresses close to the load get very high values, the contour plot for σ_{22} was capped to a minimum value of -5 Pa. Similarly, the stress contour for σ_{12} was capped to a maximum

value of 1 Pa. Along the line $x_2 = b$ these averaged stress values are reported at the center of each element and compared to the exact values. The figures show good agreement between the two, and thus VEM can be used as a powerful technique for creating fine meshes close to points of interest in our domain. It is worth noting that all elements in the discretization chosen are VEM polygons. Nevertheless, as shown earlier, we could have used a standard FEM discretization and use VEM elements only close to the applied load—with appropriate transition VEM polygons.

5. Summary and conclusions

In this manuscript we studied high-order VEM in the context of 2-D elastostatics. We discussed the construction of the method, highlighting the main differences with respect to a standard displacement-based FEM formulation. Noteworthy, VEM defines a *virtual space*, which is composed of trial/test functions whose explicit knowledge is not required. By building a projection operator it is then possible to obtain the necessary information about those functions in the standard polynomial space needed for the computation of the stiffness matrix. The construction of the virtual space is performed element by element and requires *only* mesh geometry, allowing the framework to be used regardless of the type of polygons—including non-convex shapes. However, the computation of the projection operator can be computationally expensive, in particular when the displacement field is approximated with high-order functions.

VEM's accuracy and convergence properties were compared to those of *h*- and *p*-versions of FEM. VEM guarantees the same performance but a higher robustness for arbitrary polygonal discretizations. With iso-parametric formulations, commonly used in standard FEM, a reference element is defined on which all computations are performed and then transformed to the actual position in space. Because the mapping between reference and actual elements involves the Jacobian of the transformation and badly shaped elements result in loss of accuracy, the performance of the method becomes dependent on mesh quality. Thus, for geometrically complicated problems, VEM provides a clear advantage in this regard. The numerical tests conducted showed not only that VEM's accuracy is on par with that of FEM, but also that optimal convergence rates are achieved for high-order approximations of the displacement field. VEM is therefore a viable alternative to FEM in problems with intricate geometries, where creating a matching mesh would require too much effort.

We showed how to exploit the advantages arising from this polygonal-mesh based technique for coupling non-conforming meshes. A fine VEM discretization was also used to improve the approximation properties near regions where the solution is no longer smooth. Therefore, integrating VEM in standard FEM packages results in a powerful framework for handling hanging nodes and local refinements. To that end, we gave detailed instructions for a straightforward computer implementation. At a glance, we have summarized the main differences and similarities of the two methods in Table 1.

VEM possesses features that, on the very least, put it on equal grounds to standard FEM and its variations for any problem, while for some problems it is clearly superior. The latter group includes problems with complex geometries for which a good quality mesh is difficult to obtain, solutions that require very local refinements, and situations for which two independently meshed domains need to be made compatible. These problems highlight VEM's robustness to mesh distortion. Further insight into this topic can be found in [18,21], but in general most works dealing with numerical results for VEM showcase the robustness of the method.

Acknowledgments

Appendix A. Formulation

A.1. Virtual element definitions

Rigorously, the local VEM space of interpolation order k on an element E is defined as

$$\boldsymbol{\mathcal{V}}^h(E) = \left\{ \boldsymbol{v}^h \in \left[\mathcal{H}^1(E) \cap C^0(E) \right]^2 : \left. v_i^h \right|_{\partial E} \in C^0(\partial E), \left. v_i^h \right|_{e \in E} \in \mathcal{P}_k(e), \left. \Delta v_i^h \right|_E \in \mathcal{P}_{k-2}(E) \text{ for } i = 1, 2 \right\}.$$

$$(A.1)$$

Table 1
Comparison between standard FEM and VEM.

Property	FEM	VEM
Meshing	Elements are triangles or quadrilaterals (in 2-D), and tetrahedra, hexahedra, wedges, or pyramids in 3-D • Mesh conformity is necessary, *i.e.*, hanging nodes need to be handled to avoid non-conforming discretizations	Elements can be of arbitrary polyhedral geometry (convex or non-convex) • Hanging nodes are handled straightforwardly by adding vertices
Shape functions	Known polynomials • Quadratic or higher-order interpolations require Lagrange elements with multiple nodes along edges	Defined implicitly (unknown in the interior, known on element boundaries, and can be evaluated through the dof operator) • Local shape functions include all polynomials up to order k and may contain other non-polynomials functions
Local stiffness matrix	Standard computation: usually isoparametric formulation with Gauss quadrature	Computed on the actual geometry (no mapping) • Needs the computation of the projection operator • Stiffness matrix is divided into consistency and stability parts, the latter requires a user-defined stabilization parameter
Local load vector	Standard computation: usually isoparametric formulation with Gauss quadrature	Requires the projection of the body force and it depends on polynomial order k
Convergence	*Algebraic convergence* for smooth problems	Same convergence properties as in standard FEM and *exponential convergence* for smooth problems when increasing polynomial order k (as in p–FEM)
Boundary conditions	Direct enforcement (essential BCs) • Direct computation on boundary (natural BCs)	Same enforcement and same computation since VEM shape functions are explicitly known on element boundaries[a]
Mesh quality	Accuracy is highly dependent on mesh quality[b]	It has been shown experimentally to be very robust to mesh distortion including small edges, small angles, collapsing nodes, *etc.*
Computational cost	Standard computation and assembly	Requires some overhead computations for the projector and the stabilization term • Assembly of global matrices and vectors is standard

[a]This holds for the choice of DOFs presented here because of the Kronecker-δ property. For a different choice of edge DOFs, a local problem for explicitly reconstructing the same function on the boundary is required.
[b]See [34] for a detailed survey on what constitutes a good finite element, including assessments of mesh quality.

The features of this space, as well as the corresponding DOFs and their total count were given in Section 2.3. It has been shown that these DOFs univocally define functions on $\mathcal{V}^h(E)$ [6]. As mentioned in the main text, a basis for this space will be denoted by $\{\boldsymbol{\varphi}_i\}_{i=1,\dots,2n_d} \subset \mathcal{V}^h(E)$, where each basis function takes the value 1 at its corresponding DOF and 0 on all others. The global VEM space is defined analogously to standard FEM by

$$\mathcal{V}^h = \left\{ \boldsymbol{v}^h \in \left[\mathcal{H}^1(\Omega) \cap C^0(\overline{\Omega}) \right]^2 : \ \boldsymbol{v}^h\big|_E \in \mathcal{V}^h(E), \ \text{for all } E \in \Omega^h \right\}. \tag{A.2}$$

Since shape functions shall be projected onto the space $\mathcal{P}_k(E)$, we now introduce the basis $\boldsymbol{P}_k = \{\boldsymbol{p}_\alpha\}_{\alpha=1,\dots,n_k} \subset \mathcal{P}_k(E)$, where $n_k = (k+1)(k+2)$ is the cardinality of $\boldsymbol{P}_k(E)$ since they are polynomials in two variables for both spatial directions.

Denoting with $\bar{\boldsymbol{x}} = (\bar{x}_1, \bar{x}_2)$ and h_E the centroid and diameter of element E, respectively, let the scaled monomials ξ and η be defined as

$$\xi = \left(\frac{x_1 - \bar{x}_1}{h_E} \right), \qquad \eta = \left(\frac{x_2 - \bar{x}_2}{h_E} \right). \tag{A.3}$$

The cardinality of the basis used depends on the interpolation order k, such that all the monomials of Pascal's triangle of order less then or equal to k are included. For instance,

- $k = 0$:

$$P_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}; \tag{A.4}$$

- $k = 1$:

$$P_1 = P_0, \begin{pmatrix} -\eta \\ \xi \end{pmatrix}, \begin{pmatrix} \eta \\ \xi \end{pmatrix}, \begin{pmatrix} \xi \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \eta \end{pmatrix}; \tag{A.5}$$

- $k = 2$:

$$P_2 = P_1, \begin{pmatrix} \xi^2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \xi^2 \end{pmatrix}, \begin{pmatrix} \xi\eta \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \xi\eta \end{pmatrix}, \begin{pmatrix} \eta^2 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \eta^2 \end{pmatrix}; \tag{A.6}$$

- Arbitrary k:

$$P_k = P_{k-1}, \begin{pmatrix} \xi^k \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \xi^k \end{pmatrix}, \begin{pmatrix} \xi^{k-1}\,\eta \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \xi^{k-1}\,\eta \end{pmatrix}, \dots, \begin{pmatrix} \eta^k \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \eta^k \end{pmatrix}. \tag{A.7}$$

The first three monomials ($\alpha = 1, 2, 3$) in (A.5), for which the infinitesimal strain $\epsilon(p_\alpha) = 0$, represent infinitesimal rigid body motions.

A.2. Projection operator Π

The definition of the local projection operator $\Pi \equiv \Pi_{E,k} : \mathcal{V}^h(E) \to \mathcal{P}_k(E)$, stated in (4), can be rewritten as: Given $v^h \in \mathcal{V}^h(E)$, its projection $\Pi v^h \in \mathcal{P}_k(E)$ satisfies

$$a_\varepsilon \left(v^h, p \right) = a_\varepsilon \left(\Pi v^h, p \right) \qquad \forall \; p \in \mathcal{P}_k(E). \tag{A.8}$$

This condition establishes that the internal elastic energy of the VEM function is computed exactly when tested against strains derived from polynomial displacement fields, such as those used in patch tests. In its matrix form, the projector $\tilde{\Pi}$ can be thought of as a matrix that maps the coefficients of a decomposition of a shape function φ_i in the base of $\mathcal{V}^h(E)$ into the coefficients of a linear combination of polygons in the space $\mathcal{P}_k(E)$. To compute $\tilde{\Pi}$ we first have to express v^h and p with respect to the bases $\{\varphi_i\}$ and P_k, respectively. By expressing $\Pi\varphi_i$ as a linear combination of p_β

$$\Pi(\varphi_i) = \sum_{\beta=1}^{n_k} s_{i,\beta}\; p_\beta\;, \qquad i = 1, \dots, 2n_d, \tag{A.9}$$

we recall (A.8) and use the linearity of a_E^h to obtain

$$\int_E \epsilon(\varphi_i)^\mathsf{T}\; C\epsilon(p_\alpha)\; \mathrm{d}E = \sum_{\beta=1}^{n_k} s_{i,\beta} \int_E \epsilon(p_\beta)^\mathsf{T}\; C\epsilon(p_\alpha)\; \mathrm{d}E \qquad p_\alpha \in P_k. \tag{A.10}$$

Taking $\alpha = 1, \dots, n_k$ results in a linear system of n_k equations with the n_k unknowns $s_{i,\beta}$ that can be written in compact matrix form as

$$b_i = Gs_i, \tag{A.11}$$

with

$$b_i = \begin{bmatrix} a_E^h(p_1, \varphi_i) \\ a_E^h(p_2, \varphi_i) \\ \vdots \\ a_E^h(p_{n_k}, \varphi_i) \end{bmatrix} \quad \text{and} \quad s_i = \begin{bmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,n_k} \end{bmatrix}. \tag{A.12}$$

Repeating for all basis functions φ_i, we obtain the matrix $\overline{B}$ of vectors b_i, and the matrix representation $\tilde{\Pi}$ of the projection operator:

$$\overline{B} = \begin{bmatrix} b_1 & b_2 & \dots & b_{2n_d} \end{bmatrix}, \tag{A.13}$$

$$\tilde{\Pi} = \begin{bmatrix} s_1 & s_2 & \dots & s_{2n_d} \end{bmatrix}. \tag{A.14}$$

The system (A.11) then becomes

$$\overline{B} = G\tilde{\Pi} \longrightarrow \tilde{\Pi} = G^{-1}\overline{B}. \tag{A.15}$$

The definitions and computations of $\overline{B}$ and G are further explained in Appendices B.1 and B.3, respectively. Looking at the system above we can already point out that, for the monomials that represent rigid body motion (the first three), Eq. (A.10) becomes $0 = 0$. These monomials constitute the kernel of the projector Π, so that for them we define a new projection operator, which can be thought of as the following Euclidean scalar product on the $2n_v$ vertex DOFs:

$$\frac{1}{n_v} \sum_{i=1}^{2n_v} \operatorname*{dof}_i \left(v^h \right) \operatorname*{dof}_i \left(p_\alpha \right) = \frac{1}{n_v} \sum_{i=1}^{2n_v} \operatorname*{dof}_i \left(\Pi v^h \right) \operatorname*{dof}_i \left(p_\alpha \right) \qquad \alpha = 1, 2, 3. \tag{A.16}$$

More intuitively, the conditions arising from (A.16) are nothing more than equality of average displacement in both spatial directions (p_1, p_2) and equality of (infinitesimal) rotation (p_3) between v^h and Πv^h. Finally, decomposing a VEM function v^h into a combination of basis functions φ_i and using Eq. (A.9), we can solve system (A.15) for any VEM function, where the first three rows of $\overline{B}$ are taken from (A.16).

Appendix B. Local matrices

B.1. Matrix $\overline{B}$

The computation of matrix $\overline{B}_{n_k \times 2n_d}$ in (A.13) is probably the most conceptually important one, since it involves integrating VEM shape functions inside an element, which is *a priori* not possible. This matrix is defined as

$$\overline{B} = \begin{bmatrix} a_E(p_1, \varphi_1) & a_E(p_1, \varphi_2) & \dots & a_E(p_1, \varphi_{2n_d}) \\ a_E(p_2, \varphi_1) & a_E(p_2, \varphi_2) & \dots & a_E(p_2, \varphi_{2n_d}) \\ \vdots & \vdots & \ddots & \vdots \\ a_E(p_{n_k}, \varphi_1) & a_E(p_{n_k}, \varphi_2) & \dots & a_E(p_{n_k}, \varphi_{2n_d}) \end{bmatrix}. \tag{B.1}$$

- Kernel part:
  The first three rows are computed with (A.16) such that:

$$\overline{B}_{\alpha I} = \frac{1}{n_v} \sum_{i=1}^{2n_v} \operatorname*{dof}_i \left(\varphi_I \right) \operatorname*{dof}_i \left(p_\alpha \right) \qquad \alpha = 1, 2, 3. \tag{B.2}$$

  Example:
  E is a triangle with vertices x_1, x_2, x_3 and order of accuracy $k = 1$, the kernel part of $\overline{B}$ is written as:

$$\operatorname{Ker}(\overline{B}) = \frac{1}{n_v} \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ -\eta|_{x_1} & \xi|_{x_1} & -\eta|_{x_2} & \xi|_{x_2} & -\eta|_{x_3} & \xi|_{x_3} \end{bmatrix}. \tag{B.3}$$

  When the interpolation order increases, the additional columns related to edge and internal DOFs are set to zero; this is because Eq. (A.16) uses only the values for the vertex DOFs.
- Non-kernel part:
  We can use integration by parts to transform the original computation into two separate integrals:

$$a_E(\varphi_i, p_\alpha) = \int_E \epsilon(\varphi_i)^\mathsf{T} \, \sigma(p_\alpha) \, \mathrm{d}E = -\int_E \varphi_i \cdot \left[\nabla \cdot (C \, \epsilon(p_\alpha)) \right] \mathrm{d}E + \int_{\partial E} \varphi_i \cdot \sigma(p_\alpha) \, \hat{n}_e \, \mathrm{d}e, \tag{B.4}$$

  where $\hat{n}_e$ is the outward normal vector to the boundary. The first term on the right computes the product of a VEM function against a polynomial of degree $k - 2$ in the element's interior and the second one is taken on its boundary, where the VEM functions are explicitly known. Thus, both terms can be obtained only from the knowledge of the VEM function on the DOFs defined in Section 2.3. We examine them separately:

(*i*) The first term is related to internal DOFs. Expressing the strain $\epsilon(\boldsymbol{p}_\alpha)$ in Voigt notation, and $\boldsymbol{C}$ as the usual 3×3 constitutive matrix in either plane stress/strain, the term $\nabla \cdot (\boldsymbol{C}\epsilon(\boldsymbol{p}_\alpha))$ can be written as a linear combination of the monomial basis $\boldsymbol{p}_\beta$ of the space $\mathcal{P}_{k-2}(E)$ (since derivatives are taken twice):

$$\nabla \cdot (\boldsymbol{C}\,\epsilon(\boldsymbol{p}_\alpha)) = \sum_{\beta=1}^{n_{k-2}} d_{\alpha,\beta}\,\boldsymbol{p}_\beta. \tag{B.5}$$

Substituting in (B.4), the first term becomes:

$$-\int_E \boldsymbol{\varphi}_i\,\nabla \cdot (\boldsymbol{C}\,\epsilon(\boldsymbol{p}_\alpha))\,\mathrm{d}E = -\sum_{\beta=1}^{n_{k-2}} d_{\alpha,\beta} \int_E \boldsymbol{\varphi}_i\,\cdot\boldsymbol{p}_\beta\,\mathrm{d}E. \tag{B.6}$$

Denoting by $|E|$ the area of the element, and recalling the definition of the internal DOFs (8) as the moments up to order $k-2$, the equation above can be expressed as:

$$-\sum_{\beta=1}^{n_{k-2}} d_{\alpha,\beta} \int_E \boldsymbol{\varphi}_i\,\cdot\boldsymbol{p}_\beta\,\mathrm{d}E = -|E|\sum_{\beta=1}^{n_{k-2}} d_{\alpha,\beta}\,\underset{(2kn_v+\beta)}{\mathrm{dof}}(\boldsymbol{\varphi}_i), \tag{B.7}$$

due to the property of the shape functions that $\mathrm{dof}_{(2kn_v+\beta)}(\boldsymbol{\varphi}_i) = 1$ when $i = 2kn_v + \beta$ and 0 otherwise. We are thus able to compute this integral without any kind of numerical integration by just using the definition of internal DOFs.

Example:

Consider an order of accuracy $k = 2$. The basis of the space $\mathcal{P}_{k-2}(E)$ is $\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$, which means that each element E has 2 internal DOFs:

$$\begin{aligned}
\underset{(2kn_v+1)}{\mathrm{dof}}(\boldsymbol{v}^h) &= \frac{1}{|E|} \int_E \boldsymbol{v}^h\,\cdot\begin{pmatrix} 1 \\ 0 \end{pmatrix}\,\mathrm{d}E = \frac{1}{|E|} \int_E v_1^h \mathrm{d}E, \\
\underset{(2kn_v+2)}{\mathrm{dof}}(\boldsymbol{v}^h) &= \frac{1}{|E|} \int_E \boldsymbol{v}^h\,\cdot\begin{pmatrix} 0 \\ 1 \end{pmatrix}\,\mathrm{d}E = \frac{1}{|E|} \int_E v_2^h \mathrm{d}E.
\end{aligned} \tag{B.8}$$

Considering the monomial $\boldsymbol{p}_7 = \begin{pmatrix} \eta^2 \\ 0 \end{pmatrix}$, strain and divergence of the stress are, respectively,

$$\epsilon(\boldsymbol{p}_7) = \begin{bmatrix} 0 \\ 0 \\ 2\eta \end{bmatrix}, \qquad \nabla \cdot (\boldsymbol{C}\epsilon(\boldsymbol{p}_7)) = d_{7,1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = d_{7,1}\,\boldsymbol{p}_1, \tag{B.9}$$

where $d_{7,1}$ is the first coefficient of $\boldsymbol{p}_7$, *i.e.*, the constant part (B.5). This leads to:

$$\begin{aligned}
-\int_E \boldsymbol{\varphi}_i\,\nabla \cdot (\boldsymbol{C}\,\epsilon(\boldsymbol{p}_7))\,\mathrm{d}E &= -d_{7,1} \int_E \boldsymbol{\varphi}_i\,\cdot\boldsymbol{p}_1\,\mathrm{d}E \\
&= -d_{7,1}\,|E|\,\underset{(2kn_v+1)}{\mathrm{dof}}(\boldsymbol{\varphi}_i) = \begin{cases} -d_{7,1}\,|E| & \text{if } 2kn_v+1 = i, \\ 0 & \text{otherwise.} \end{cases}
\end{aligned} \tag{B.10}$$

(*ii*) The second term of (B.4) is related to boundary DOFs. We can split the boundary integral as a sum of edge contributions, which are computed using a quadrature rule:

$$\int_{\partial E} \boldsymbol{\varphi}_i \cdot \boldsymbol{\sigma}(\boldsymbol{p}_\alpha)\hat{\boldsymbol{n}}_e\,\mathrm{d}e = \sum_{j=1}^{n_v} \int_{e_j} \boldsymbol{\varphi}_i \cdot \boldsymbol{\sigma}(\boldsymbol{p}_\alpha)\hat{\boldsymbol{n}}_{e_j}\,\mathrm{d}e = \sum_{j=1}^{n_v} \frac{|e_j|}{2} \sum_{r=1}^{k+1} w_r\,\boldsymbol{\varphi}_i\Big|_{\xi_r} \cdot \Big(\boldsymbol{\sigma}(\boldsymbol{p}_\alpha)\Big|_{\xi_r} \cdot \hat{\boldsymbol{n}}_{e_j}\Big), \tag{B.11}$$

where e_j refers to the jth edge of ∂E and $\hat{\boldsymbol{n}}_{e_j}$ its corresponding outward unit normal vector. By choosing Gauss-Lobatto points ξ_r (with associated weights w_r) for the definition of the edge DOFs, numerical integration over the boundary is simplified as the values of $\boldsymbol{v}^h$ at these integration points are precisely the DOFs of $\boldsymbol{v}^h$. The same is true for $\boldsymbol{\varphi}_i$ in particular. We point out that the choice of taking $k + 1$ Gauss-Lobatto points on each edge (including the vertices) guarantees an optimal choice; this is because the integrand of (B.11) is a polynomial of order $2k - 1$, which is the order of integration of the $k + 1$ Gauss-Lobatto rule. For basis functions associated with internal DOFs, this term is of course zero.

Example:

Let us consider the generic VEM shape function $\boldsymbol{\varphi}_{2i} = \begin{pmatrix} 0 & \varphi_i \end{pmatrix}^{\mathsf{T}}$ such that its value at the node ξ_r is $\boldsymbol{\varphi}_{2i}\big|_{\xi_r} = \begin{bmatrix} 0 & 1 \end{bmatrix}^{\mathsf{T}}$ and $\begin{bmatrix} 0 & 0 \end{bmatrix}^{\mathsf{T}}$ for all other nodes. Then

$$\int_{\partial E} \boldsymbol{\varphi}_{2i} \cdot \left(\boldsymbol{\sigma}\left(\boldsymbol{p}_\alpha \right) \cdot \hat{\boldsymbol{n}}_e \right) \, \mathrm{d}e = w_r \begin{pmatrix} 0 \\ 1 \end{pmatrix} \cdot \left(\boldsymbol{\sigma}(\boldsymbol{p}_\alpha)\big|_{\xi_r} \cdot \hat{\boldsymbol{n}}_{e_j} \right).$$

since ξ_r is the only integration point where $\boldsymbol{\varphi}_{2i}$ is nonzero.

B.2. Matrix $\boldsymbol{D}$

This $2n_d \times n_k$ matrix is used to express the projection of a VEM function as a linear combination of the VEM functions themselves, unlike before where the projection was expressed as a linear combination of polynomials. Therefore, through this matrix the projection can be applied from the VEM space onto itself. It is obtained by evaluating the polynomials of degree $\leq k$ at the DOFs defined for $\mathcal{V}^h(E)$:

$$\boldsymbol{D} = \begin{bmatrix} \mathrm{dof}_1(\boldsymbol{p}_1) & \mathrm{dof}_1(\boldsymbol{p}_2) & \ldots & \mathrm{dof}_1(\boldsymbol{p}_{n_k}) \\ \mathrm{dof}_2(\boldsymbol{p}_1) & \mathrm{dof}_2(\boldsymbol{p}_2) & \ldots & \mathrm{dof}_2(\boldsymbol{p}_{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{dof}_{2n_d}(\boldsymbol{p}_1) & \mathrm{dof}_{2n_d}(\boldsymbol{p}_2) & \ldots & \mathrm{dof}_{2n_d}(\boldsymbol{p}_{n_k}). \end{bmatrix} \tag{B.12}$$

(*i*) Boundary DOFs: $1 \leq i \leq 2\,k\,n_v$, so $\mathrm{dof}_i(\boldsymbol{p}_\alpha)$ is the value of $\boldsymbol{p}_\alpha$ in the node associated with the ith DOF.

(*ii*) Internal DOFs: $2kn_v + 1 \leq i \leq 2n_d$. Applying definition (8) to $\boldsymbol{p}_\alpha$ and performing numerical integration on the product of known polynomials, we are able to compute those terms as:

$$\mathop{\mathrm{dof}}_{(2kn_v+\beta)} \boldsymbol{p}_\alpha = \frac{1}{|E|} \int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{p}_\beta \, \mathrm{d}E \qquad \text{with } \boldsymbol{p}_\beta \in \mathcal{P}_{k-2}(E). \tag{B.13}$$

Example:

For $k \geq 2$ the first two DOFs of $\boldsymbol{p}_4 = \begin{pmatrix} \eta \\ \xi \end{pmatrix}$ are evaluations of its horizontal and vertical component at $\boldsymbol{x}_1$, with $\boldsymbol{x}_1$ being the first vertex of the polygon: $\mathrm{dof}_1(\boldsymbol{p}_4) = \eta(\boldsymbol{x}_1)$, $\mathrm{dof}_2(\boldsymbol{p}_4) = \xi(\boldsymbol{x}_1)$. The first internal DOF in this case would be

$$\mathop{\mathrm{dof}}_{2kn_v+1}(\boldsymbol{p}_4) = \frac{1}{|E|} \int_E \boldsymbol{p}_4 \cdot \boldsymbol{p}_1 \, \mathrm{d}E = \frac{1}{|E|} \int_E \boldsymbol{p}_4 \cdot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \, \mathrm{d}E = \frac{1}{|E|} \int_E \eta \, \mathrm{d}E.$$

B.3. Matrix $\boldsymbol{G}$

This $n_k \times n_k$ matrix, appearing in (A.15), is defined as:

$$\boldsymbol{G} = \begin{bmatrix} a_E(\boldsymbol{p}_1, \boldsymbol{p}_1) & a_E(\boldsymbol{p}_1, \boldsymbol{p}_2) & \cdots & a_E(\boldsymbol{p}_1, \boldsymbol{p}_{n_k}) \\ a_E(\boldsymbol{p}_2, \boldsymbol{p}_1) & a_E(\boldsymbol{p}_2, \boldsymbol{p}_2) & \cdots & a_E(\boldsymbol{p}_2, \boldsymbol{p}_{n_k}) \\ \vdots & \vdots & \ddots & \vdots \\ a_E(\boldsymbol{p}_{n_k}, \boldsymbol{p}_1) & a_E(\boldsymbol{p}_{n_k}, \boldsymbol{p}_2) & \cdots & a_E(\boldsymbol{p}_{n_k}, \boldsymbol{p}_{n_k}) \end{bmatrix}. \tag{B.14}$$

Its computation is straightforward as it consists on an integration of a product of known polynomials over a polygonal domain. However, it can also be expressed in terms of the already presented matrices $\overline{\boldsymbol{B}}$ and $\boldsymbol{D}$ as $\boldsymbol{G} = \overline{\boldsymbol{B}} \boldsymbol{D}$. Therefore, it is not necessary to compute $\boldsymbol{G}$ if $\overline{\boldsymbol{B}}$ and $\boldsymbol{D}$ were already obtained—although it is a useful debugging tool to verify that both procedures lead to the same result within numerical precision.

Appendix C. Load term

In this section we show how to compute the VEM loading term r_E^f due to body forces $\boldsymbol{f} = (f_1, f_2)$ as detailed in Section 2.3. As for the construction of the stiffness matrix, a projection operator is used to express the VEM

displacement $\boldsymbol{v}^h$ as a polynomial. We introduce the classical orthogonal L^2 projection $\Pi_{k-2} : \mathcal{V}^h(E) \to \mathcal{P}_{k-2}(E)$ such that the loading term can be approximated. Π_{k-2} is defined with the following orthogonality condition:

$$\int_E \boldsymbol{p}_\alpha \, \boldsymbol{v}^h \, \mathrm{d}E = \int_E \boldsymbol{p}_\alpha \, \Pi_{k-2}(\boldsymbol{v}^h) \, \mathrm{d}E \qquad \forall \, \boldsymbol{p}_\alpha \in \mathcal{P}_{k-2}(E). \tag{C.1}$$

The computation of the local load vector due to body forces can be discriminated among different cases, depending on the interpolation order k, as follows:

(*i*) Case $k = 1$: For this particular case the space $\mathcal{P}_{k-2}(E)$ is not defined, so we compute the loading term with an integration rule based on the value of $\boldsymbol{v}^h$ at the vertices of the polygon:

$$\left(\boldsymbol{f}, \boldsymbol{v}^h \right)_E \approx \frac{1}{|E|} \int_E \boldsymbol{f} \, \mathrm{d}E \cdot \frac{1}{n_v} \sum_{j=1}^{n_v} \boldsymbol{v}^h \left(\boldsymbol{x}_j \right), \tag{C.2}$$

where $\boldsymbol{x}_j$ is the spatial location of the jth vertex. Considering now the basis functions $\boldsymbol{\varphi}_i$ to obtain the discrete load vector associated with the ith DOF, Eq. (C.2) becomes

$$\left(\boldsymbol{f}, \boldsymbol{\varphi}_i \right)_E \approx \frac{1}{|E|} \int_E \boldsymbol{f} \, \mathrm{d}E \cdot \frac{1}{n_v} \sum_{j=1}^{n_v} \varphi_i \left(\boldsymbol{x}_j \right), \quad i = 1, \ldots, 2n_v. \tag{C.3}$$

(*ii*) Case $k = 2$: In this case, a better choice can be made in order to retain optimal convergence rates [35]. We use the L^2 projection Π_k^0 onto the space of polynomials $\mathcal{P}_k(E)$, defined as:

$$\int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{\varphi}_i \, \mathrm{d}E = \int_E \boldsymbol{p}_\alpha \cdot \Pi_k^0(\boldsymbol{\varphi}_i) \, \mathrm{d}E \qquad \forall \, \boldsymbol{p}_\alpha \in \mathcal{P}_k(E), \tag{C.4}$$

where

$$\Pi_k^0(\boldsymbol{\varphi}_i) = \sum_{\beta=1}^{n_k} s_{i,\beta} \, \boldsymbol{p}_\beta. \tag{C.5}$$

After substituting we obtain the following system of equations:

$$\int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{\varphi}_i \, \mathrm{d}E = \sum_{\beta=1}^{n_k} s_{i,\beta} \int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{p}_\beta \, \mathrm{d}E \qquad \forall \, \boldsymbol{p}_\alpha \in \mathcal{P}_k(E), \tag{C.6}$$

which is expressed in matrix form as

$$\boldsymbol{q}_i^0 = \boldsymbol{H}^0 \boldsymbol{s}_i^0. \tag{C.7}$$

Extending column-wise to all shape functions, the system can be rewritten as:

$$\boldsymbol{Q}^0 = \boldsymbol{H}^0 \, \boldsymbol{\Pi}_k^0 \longrightarrow \boldsymbol{\Pi}_k^0 = \left(\boldsymbol{H}^0 \right)^{-1} \boldsymbol{Q}^0. \tag{C.8}$$

While the matrix $\boldsymbol{H}_{n_k \times n_k}^0$ can be computed numerically since the integrand is a product of known polynomials, the matrix $\boldsymbol{Q}_{n_k \times 2n_d}^0$ is computed as follows:

$$\boldsymbol{Q}_{\alpha i}^0 = \begin{cases} \boldsymbol{Q}_{\alpha i} & \alpha \le n_{k-2}, \\ (\boldsymbol{H}^0 \boldsymbol{G}^{-1} \boldsymbol{B})_{\alpha,i} & n_{k-2} < \alpha \le n_k, \end{cases} \tag{C.9}$$

where $\boldsymbol{Q}_{n_{k-2} \times 2n_d}$ is computed as in (C.19). Once $\boldsymbol{\Pi}_k^0$ is obtained through (C.8), it can be substituted in (25) using Π_k^0 instead of Π_{k-2} to construct the local loading term associated to shape function $\boldsymbol{\varphi}_i$ as

$$\left(\boldsymbol{f}, \boldsymbol{\varphi}_i \right)_E \approx \int_E \boldsymbol{f} \cdot \Pi_k^0(\boldsymbol{\varphi}_i) \, \mathrm{d}E = \int_E \boldsymbol{f} \cdot \left(\sum_{\beta=1}^{n_k} s_{i,\beta} \, \boldsymbol{p}_\beta \right) \mathrm{d}E = \int_E \boldsymbol{f} \cdot \left(\sum_{\beta=1}^{n_k} (\boldsymbol{\Pi}_k^0)_{\beta i} \, \boldsymbol{p}_\beta \right) \mathrm{d}E. \tag{C.10}$$

(*iii*) Case $k > 2$: The projection operator Π_{k-2} can be computed using internal DOFs. First we write (C.1) in terms of $\boldsymbol{\varphi}_i$:

$$\int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{\varphi}_i \, \mathrm{d}E = \int_E \boldsymbol{p}_\alpha \cdot \Pi_{k-2}(\boldsymbol{\varphi}_i) \, \mathrm{d}E \qquad \forall \, \boldsymbol{p}_\alpha \in \mathcal{P}_{k-2}(E), \tag{C.11}$$

and then we express $\Pi_{k-2}(\boldsymbol{\varphi}_i)$ as a linear combination of the basis $\boldsymbol{p}_\beta \in \mathcal{P}_{k-2}(E)$:

$$\Pi_{k-2}(\boldsymbol{\varphi}_i) = \sum_{\beta=1}^{n_{k-2}} r_{i,\beta}\ \boldsymbol{p}_\beta. \tag{C.12}$$

This leads to the system of equations

$$\int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{\varphi}_i\ \mathrm{d}E = \sum_{\beta=1}^{n_{k-2}} r_{i,\beta} \int_E \boldsymbol{p}_\alpha \cdot \boldsymbol{p}_\beta\ \mathrm{d}E \qquad \forall\ \boldsymbol{p}_\alpha \in \mathcal{P}_{k-2}(E), \tag{C.13}$$

whose matrix form is

$$\boldsymbol{q}_i = \boldsymbol{H}\boldsymbol{r}_i. \tag{C.14}$$

Extending column-wise to all shape functions, the system can be written as

$$\boldsymbol{Q} = \boldsymbol{H}\ \boldsymbol{\Pi}_{k-2} \longrightarrow \boldsymbol{\Pi}_{k-2} = \boldsymbol{H}^{-1}\boldsymbol{Q}, \tag{C.15}$$

where $\boldsymbol{\Pi}_{k-2}$ is the matrix representation of the projection operator:

$$\boldsymbol{\Pi}_{k-2} = \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \dots & \boldsymbol{r}_{2n_d} \end{bmatrix}. \tag{C.16}$$

Matrix $\boldsymbol{Q}_{n_{k-2} \times 2n_d}$ is computable thanks to the internal DOFs, as shown in Appendix C.1, while matrix $\boldsymbol{H}_{n_{k-2} \times n_{k-2}}$ can be computed numerically since the integrand is a product of known polynomials. Once $\boldsymbol{\Pi}_{k-2}$ is obtained through (C.15), it can be substituted in (24) to construct the local loading term associated to shape function $\boldsymbol{\varphi}_i$ as:

$$(\boldsymbol{f}, \boldsymbol{\varphi}_i)_E \approx \int_E \boldsymbol{f}\ \cdot \Pi_{k-2}(\boldsymbol{\varphi}_i)\ \mathrm{d}E = \int_E \boldsymbol{f}\ \cdot \left(\sum_{\beta=1}^{n_{k-2}} r_{i,\beta}\ \boldsymbol{p}_\beta\right)\ \mathrm{d}E = \int_E \boldsymbol{f}\ \cdot \left(\sum_{\beta=1}^{n_{k-2}} (\boldsymbol{\Pi}_{k-2})_{\beta i}\ \boldsymbol{p}_\beta\right)\ \mathrm{d}E. \tag{C.17}$$

For all cases presented, the complete local load vector due to body forces is obtained by collecting all base functions' contributions:

$$r_E^f = \begin{pmatrix} (\boldsymbol{f}, \boldsymbol{\varphi}_1)_E \\ (\boldsymbol{f}, \boldsymbol{\varphi}_2)_E \\ \vdots \\ (\boldsymbol{f}, \boldsymbol{\varphi}_{2n_d})_E \end{pmatrix}. \tag{C.18}$$

C.1. Matrix $\boldsymbol{Q}$

This $n_{k-2} \times 2n_d$ matrix is obtained as

$$\boldsymbol{Q} = \begin{bmatrix} \int_E \boldsymbol{p}_1\ \cdot \boldsymbol{\varphi}_1\ \mathrm{d}E & \int_E \boldsymbol{p}_1\ \cdot \boldsymbol{\varphi}_2\ \mathrm{d}E & \dots & \int_E \boldsymbol{p}_1\ \cdot \boldsymbol{\varphi}_{2n_d}\ \mathrm{d}E \\ \int_E \boldsymbol{p}_2\ \cdot \boldsymbol{\varphi}_1\ \mathrm{d}E & \int_E \boldsymbol{p}_2\ \cdot \boldsymbol{\varphi}_2\ \mathrm{d}E & \dots & \int_E \boldsymbol{p}_2\ \cdot \boldsymbol{\varphi}_{2n_d}\ \mathrm{d}E \\ \vdots & \vdots & \ddots & \vdots \\ \int_E \boldsymbol{p}_{n_{k-2}}\ \cdot \boldsymbol{\varphi}_1\ \mathrm{d}E & \int_E \boldsymbol{p}_{n_{k-2}}\ \cdot \boldsymbol{\varphi}_2\ \mathrm{d}E & \dots & \int_E \boldsymbol{p}_{n_{k-2}}\ \cdot \boldsymbol{\varphi}_{2n_d}\ \mathrm{d}E \end{bmatrix}. \tag{C.19}$$

Recalling (8), we can express the internal DOFs of $\boldsymbol{\varphi}_i$ as:

$$\operatorname*{dof}_{(2kn_v+\alpha)}(\boldsymbol{\varphi}_i) = \frac{1}{|E|} \int_E \boldsymbol{\varphi}_i\ \cdot \boldsymbol{p}_\alpha\ \mathrm{d}E \qquad \forall\ \boldsymbol{p}_\alpha \in \mathcal{P}_{k-2}(E). \tag{C.20}$$

So the generic terms of $\boldsymbol{Q}$ can be written as $\int_E \boldsymbol{p}_\alpha\ \cdot \boldsymbol{\varphi}_i\ \mathrm{d}E = |E|\ \mathrm{dof}_{(2kn_v+\alpha)}(\boldsymbol{\varphi}_i)$, which due to the Kronecker-δ property of the basis functions, is equal to $|E|$ if $2kn_v + \alpha = i$ and 0 otherwise.

Appendix D. Strains and stresses

These are computed element-wise once the displacement field is obtained. Once again, we remark that we do not have the knowledge of v^h in the interior of the element, and therefore the discrete solution is approximated by its projection on the polynomial space as

$$v^h \approx \Pi(v^h) = \mathbf{\Pi}(N)u^\varepsilon = \begin{bmatrix} \Pi(\varphi_1) & \Pi(\varphi_2) & \dots & \Pi(\varphi_{2n_d}) \end{bmatrix} u^\varepsilon, \tag{D.1}$$

where $u^\varepsilon = \begin{pmatrix} u_1^1 & u_2^1 & \dots & u_1^{n_d} & u_2^{n_d} \end{pmatrix}^\top$ is the vector of the local values of the DOFs for the discrete solution. Recalling (A.9), the projection of the basis functions can be written as:

$$\Pi(N) = \begin{bmatrix} p_1 & p_2 & \dots & p_{n_k} \end{bmatrix} \tilde{\mathbf{\Pi}}, \tag{D.2}$$

where $\tilde{\mathbf{\Pi}}_{n_k \times 2n_d}$ is the projection matrix defined in (A.15) and it is used here to obtain the coefficients of the combination of polynomials that is the projection. Substituting in (D.1) we obtain the projection of the discrete displacement solution v^h:

$$\Pi(v^h) = \begin{bmatrix} p_1 & p_2 & \dots & p_{n_k} \end{bmatrix} \tilde{\mathbf{\Pi}} \, u^\varepsilon \tag{D.3}$$

For the strain computation, as in standard FEM, we need the matrix of the shape functions derivatives, denoted as ∂N:

$$\partial N \approx \partial \left(\Pi(N) \right) = \partial \begin{bmatrix} p_1 & p_2 & \dots & p_{n_k} \end{bmatrix} \tilde{\mathbf{\Pi}}, \tag{D.4}$$

where

$$\partial \begin{bmatrix} p_1 & p_2 & \dots & p_{n_k} \end{bmatrix} = \begin{bmatrix} \partial_{x_1} p_{1,1} & \partial_{x_1} p_{2,1} & \dots & \partial_{x_1} p_{n_k,1} \\ \partial_{x_2} p_{1,2} & \partial_{x_2} p_{2,2} & \dots & \partial_{x_2} p_{n_k,2} \\ \partial_{x_2} p_{1,1} + \partial_{x_1} p_{1,2} & \partial_{x_2} p_{2,1} + \partial_{x_1} p_{2,2} & \dots & \partial_{x_2} p_{n_k,1} + \partial_{x_1} p_{n_k,2} \end{bmatrix}. \tag{D.5}$$

Substituting we finally obtain an expression for strains:

$$\epsilon(v^h) = \partial N \, \tilde{\mathbf{\Pi}} \, u^\varepsilon. \tag{D.6}$$

Once the strains are known, stresses can be directly computed by multiplication with the material constitutive matrix C, *i.e.*,

$$\sigma(v^h) = C \epsilon(v^h). \tag{D.7}$$

References

[1] F. Brezzi, K. Lipnikov, V. Simoncini, A family of mimetic finite difference methods on polygonal and polyhedral meshes, Math. Models Methods Appl. Sci. 15 (10) (2005) 1533–1551, http://dx.doi.org/10.1142/S0218202505000832, arXiv:http://www.worldscientific.com/doi/pdf/10.1142/S0218202505000832, URL http://www.worldscientific.com/doi/abs/10.1142/S0218202505000832.

[2] L. Beirão da Veiga, G. Manzini, The Mimetic Finite Element Method and the Virtual Element Method for Elliptic Problems with Arbitrary Regularity, Los Alamos National Laboratory. Technical Report LA-UR-12-22977.

[3] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Methods Appl. Sci. 23 (01) (2013) 199–214, http://dx.doi.org/10.1142/S0218202512500492, arXiv:http://www.worldscientific.com/doi/pdf/10.1142/S0218202512500492, URL http://www.worldscientific.com/doi/abs/10.1142/S0218202512500492.

[4] I. Babuška, B. Szabó, I. Katz, The *p*-version of the finite element method, SIAM J. Numer. Anal. 18 (3) (1981) 515–545, http://dx.doi.org/10.1137/0718033, arXiv:https://doi.org/10.1137/0718033.

[5] A.L. Gain, C. Talischi, G.H. Paulino, On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes, Comput. Methods Appl. Mech. Engrg. 282 (2014) 132–160, http://dx.doi.org/10.1016/j.cma.2014.05.005, URL http://www.sciencedirect.com/science/article/pii/S0045782514001509.

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

[7] E. Artioli, S. de Miranda, C. Lovadina, L. Patruno, A stress/displacement virtual element method for plane elasticity problems, Comput. Methods Appl. Mech. Engrg. 325 (2017) 155–174, http://dx.doi.org/10.1016/j.cma.2017.06.036, URL http://www.sciencedirect.com/science/article/pii/S0045782517302475.

[8] H. Chi, L. Beirão da Veiga, G. Paulino, Some basic formulations of the virtual element method (vem) for finite deformations, Comput. Methods Appl. Mech. Engrg. 318 (2017) 148–192, http://dx.doi.org/10.1016/j.cma.2016.12.020, URL http://www.sciencedirect.com/science/article/pii/S0045782516309094.

[9] E. Artioli, L. Beirão da Veiga, C. Lovadina, E. Sacco, Arbitrary order 2D virtual elements for polygonal meshes: part II, inelastic problem, Comput. Mech. 60 (4) (2017) 643–657, http://dx.doi.org/10.1007/s00466-017-1429-9.

[10] R.L. Taylor, E. Artioli, VEM For inelastic solids, in: E. Oñate, D. Peric, E. de Souza Neto, M. Chiumenti (Eds.), Advances in Computational Plasticity: A Book in Honour of D. Roger J. Owen, Springer International Publishing, Cham, 2018, pp. 381–394, http://dx.doi.org/10.1007/978-3-319-60885-3_18.

[11] P. Antonietti, M. Bruggi, S. Scacchi, M. Verani, On the virtual element method for topology optimization on polygonal meshes, Comput. Math. Appl. 74 (5) (2017) 1091–1109, http://dx.doi.org/10.1016/j.camwa.2017.05.025.

[12] F. Brezzi, L.D. Marini, Virtual element methods for plate bending problems, Comput. Methods Appl. Mech. Engrg. 253 (2013) 455–462, http://dx.doi.org/10.1016/j.cma.2012.09.012, URL http://www.sciencedirect.com/science/article/pii/S0045782512002940.

[13] P. Wriggers, W.T. Rust, B.D. Reddy, A virtual element method for contact, Comput. Mech. 1039–1050 http://dx.doi.org/10.1007/s00466-016-1331-x.

[14] O. Andersen, H.M. Nilsen, X. Raynaud, Virtual element method for geomechanical simulations of reservoir models, Comput. Geosci. 21 (5) (2017) 877–893, http://dx.doi.org/10.1007/s10596-017-9636-1.

[15] M.F. Benedetto, A. Caggiano, G. Etse, Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials, Comput. Methods Appl. Mech. Engrg. 338 (2018) 41–67, http://dx.doi.org/10.1016/j.cma.2018.04.001, URL http://www.sciencedirect.com/science/article/pii/S0045782518301713.

[16] L. Beirão da Veiga, A. Chernov, L. Mascotto, A. Russo, Exponential convergence of the hp virtual element method in presence of corner singularities, Numer. Math. 138 (3) (2018) 581–613, http://dx.doi.org/10.1007/s00211-017-0921-7.

[17] L. Beirão da Veiga, A. Chernov, L. Mascotto, A. Russo, Basic principles of hp virtual elements on quasiuniform meshes, Math. Models Methods Appl. Sci. 26 (08) (2016) 1567–1598, http://dx.doi.org/10.1142/S021820251650038X, arXiv:http://www.worldscientific.com/doi/pdf/10.1142/S021820251650038X, URL http://www.worldscientific.com/doi/abs/10.1142/S021820251650038X.

[18] L. Beirão 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, URL http://www.sciencedirect.com/science/article/pii/S0898122117301839.

[19] A. Ortiz-Bernardin, C. Alvarez, N. Hitschfeld, A. Russo, R. Silva-Valenzuela, E. Olate-Sanzana, Veamy: an extensible object-oriented c++ library for the virtual element method, Numer. Algorithms (2018) http://dx.doi.org/10.1007/s11075-018-00651-0.

[20] E. Artioli, L. Beirão da Veiga, C. Lovadina, E. Sacco, Arbitrary order 2D virtual elements for polygonal meshes: part I, elastic problem, Comput. Mech. 60 (3) (2017) 355–377, http://dx.doi.org/10.1007/s00466-017-1404-5.

[21] L. Beirão da Veiga, F. Brezzi, F. Dassi, L.D. Marini, A. Russo, Serendipity virtual elements for general elliptic equations in three dimensions, Chinese Ann. Math. Ser. B 39 (2) (2018) 315–334.

[22] I. Babuška, A. Aziz, On the angle condition in the finite element method, SIAM J. Numer. Anal. 13 (2) (1976) 214–226, http://dx.doi.org/10.1137/0713021, arXiv:https://doi.org/10.1137/0713021.

[23] J.R. Shewchuk, What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures, Preprint, University of California at Berkeley, 2002.

[24] V. Gyrya, H.M. Mourad, C1-continuous Virtual Element Method for Poisson-Kirchhoff plate problem, Tech. rep., Los Alamos National Laboratory, 2016, http://dx.doi.org/10.2172/1327994.

[25] P.F. Antonietti, L. Beirão da Veiga, S. Scacchi, M. Verani, A C^1 virtual element method for the Cahn–Hilliard Equation with Polygonal Meshes, SIAM J. Numer. Anal. 54 (1) (2016) 34–56, http://dx.doi.org/10.1137/15M1008117, arXiv:https://doi.org/10.1137/15M1008117.

[26] S. Berrone, A. Borio, Orthogonal polynomials in badly shaped polygonal elements for the virtual element method, Finite Elem. Anal. Des. 129 (2017) 14–31, http://dx.doi.org/10.1016/j.finel.2017.01.006, URL http://www.sciencedirect.com/science/article/pii/S0168874X16302852.

[27] E. Artioli, S. de Miranda, C. Lovadina, L. Patruno, Virtual element method: an equilibrium-based stress recovery procedure, Internat. J. Numer. Methods Engrg. 117 (8) (2019) 885–900, URL https://doi.org/10.1002/nme.5983.

[28] R.L. Taylor, J.C. Simo, O.C. Zienkiewicz, A.C.H. Chan, The patch test—a condition for assessing fem convergence, Internat. J. Numer. Methods Engrg. 22 (1) (1986) 39–62, http://dx.doi.org/10.1002/nme.1620220105, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.1620220105, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.1620220105.

[29] F. Daneshmand, M.J. Kazemzadeh-Parsi, Static and dynamic analysis of 2d and 3d elastic solids using the modified fgfem, Finite Elem. Anal. Des. 45 (11) (2009) 755–765, http://dx.doi.org/10.1016/j.finel.2009.06.003, URL http://www.sciencedirect.com/science/article/pii/S0168874X09000730.

[30] B.A. Szabó, I. Babuška, Finite Element Analysis, John Wiley and Sons, New York, 1991.

[31] E. Kirsch, Die theorie der Elastizität und die Bedürfnisse der Festigkeitslehre, Z. Vereines deutscher Ingenieure 42 (1898) 797–807.

[32] A.-A. Flamant, Sur la répartition des pressions dans un solide rectangulaire chargé transversalement, C. R.Acad. Sci. 114 (1892) 1465–1468.

[33] P. Podio-Guidugli, A. Favata, The flamant problem, in: Elasticity for Geotechnicians: A Modern Exposition of Kelvin, Boussinesq, Flamant, Cerruti, Melan, and Mindlin Problems, Springer International Publishing, Cham, 2014, pp. 55–77, http://dx.doi.org/10.1007/978-3-319-01258-2_4.

[34] J. Shewchuk, What is a good linear finite element? interpolation, conditioning, anisotropy, and quality measures (preprint), Univ. Calif. Berkeley 73 (2002) 137.

[35] L. Beirão 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, arXiv:http://www.worldscientific.com/doi/pdf/10.1142/S021820251440003X, URL http://www.worldscientific.com/doi/abs/10.1142/S021820251440003X.