Structural and Multidisciplinary Optimization (2020) 62:1089–1114
https://doi.org/10.1007/s00158-019-02268-w

RESEARCH PAPER

Virtual element method (VEM)-based topology optimization: an integrated framework

Heng Chi[1] · Anderson Pereira[2] · Ivan F. M. Menezes[2] · Glaucio H. Paulino[1]

Received: 27 July 2018 / Revised: 29 January 2019 / Accepted: 21 March 2019 / Published online: 24 June 2019

Abstract

We present a virtual element method (VEM)-based topology optimization framework using polyhedral elements, which allows for convenient handling of non-Cartesian design domains in three dimensions. We take full advantage of the VEM properties by creating a unified approach in which the VEM is employed in both the structural and the optimization phases. In the structural problem, the VEM is adopted to solve the three-dimensional elasticity equation. Compared to the finite element method, the VEM does not require numerical integration (when linear elements are used) and is less sensitive to degenerated elements (e.g., ones with skinny faces or small edges). In the optimization problem, we introduce a continuous approximation of material densities using the VEM basis functions. When compared to the standard element-wise constant approximation, the continuous approximation enriches the geometrical representation of structural topologies. Through two numerical examples with exact solutions, we verify the convergence and accuracy of both the VEM approximations of the displacement and material density fields. We also present several design examples involving non-Cartesian domains, demonstrating the main features of the proposed VEM-based topology optimization framework. The source code for a MATLAB implementation of the proposed work, named `PolyTop3D`, is available in the (electronic) Supplementary Material accompanying this publication.

Keywords Topology optimization · Polyhedral meshes · Virtual element method · MATLAB software

Responsible Editor: Byeng D Youn

Dedicated to the memory of Prof. Luiz Eloy Vaz (January 24, 1947-April 15, 2014).

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s00158-019-02268-w) contains supplementary material, which is available to authorized users.

✉ Glaucio H. Paulino
paulino@gatech.edu

Heng Chi
hchi6@gatech.edu

Anderson Pereira
anderson@puc-rio.br

Ivan F. M. Menezes
ivan@puc-rio.br

[1] School of Civil and Environmental Engineering, Georgia Institute of Technology, 790 Atlantic Drive, Atlanta, GA, 30332, USA

[2] Pontifical Catholic University of Rio de Janeiro (PUC-Rio), Rua Marquês de São Vicente, 225, Rio de Janeiro, R.J. 22451-900, Brazil

1 Introduction

Topology optimization is a powerful computational tool to design optimal structures under given loads and boundary conditions. Since the seminal work of Bendsøe and Kikuchi (1988), the field of topology optimization has experienced tremendous growth and had a major impact on several areas of engineering, science and technology. For an overview of this field, we refer the interested readers to available textbooks (Christensen and Klarbring 2009; Haftka and Gürdal 2012; Bendsoe and Sigmund 2013) and a review paper (Rozvany 2009). Among various topology optimization approaches, the density-based approach is commonly adopted, in which the geometry is parametrized by a material density function and the displacement field is discretized by, for example, by finite elements. Because of its simplicity and efficiency, the choice of piece-wise constant density parametrization, where each finite element is assigned with a constant density, is typically employed in conjunction with a lower-order Lagrangian-type displacement approximation. However, this choice of density parametrization suffers from numerical instabilities, such as checkerboard patterns and one node connections (Diaz

and Sigmund 1995; Sigmund et al. 1998). In a general setting, Jog and Haber (1996) formulated topology optimization as a mixed variational problem and demonstrated how different choices of displacement and density interpolations affect the appearance of numerical instabilities. Along this direction, different choices of displacement and density approximations were investigated. One popular choice is to approximate material density fields with nodal design variables. For instance, Matsui and Terada (2004) and Rahmatalla and Swan (2004) employed continuous approximations of the density fields interpolated by Lagrangian finite element basis functions. This approach, sometimes known as continuous approximation of material distribution (CAMD), is effective in preventing the checkerboard patterns but may lead to other forms of numerical instabilities such as "islanding" and "layering" with linear finite elements (Rahmatalla and Swan 2004). Guest et al. (2004) introduced nodal design variables and employed a projection map with an embedded length scale to construct an element-wise constant density field. Belytschko et al. (2003) proposed a formulation in which implicit functions are used to describe topologies of the designs. The implicit functions are descirbed by their nodal values and interpolated by C^0 finite element shape functions. Alternatively, Kang and Wang (2011) proposed to decouple the approximations of density and displacement fields into two independent discretizations. In their approach, the density field is interpolated from nodal design variables using non-local Shepard interpolants, and the displacement field is approximated using standard finite elements. This feature helps the method to better handle problems with complex design domains and allows for flexible adaptive representation of the structural topologies (Wang et al. 2013). Other related choices also include the use of higher-order displacement approximations (Diaz and Sigmund 1995; Matsui and Terada 2004; Groen et al. 2017), non-conforming elements (Jang et al. 2003; Jang et al. 2005), and approximating density and displacement fields on different discretizations (Paulino and Le 2009; Guest and Smith 2010). The multi-resolution topology optimization approaches (see, e.g., Nguyen et al. (2010, 2012); Filipov et al. 2016; Groen et al. 2017) belong to the last family. In addition to various types of numerical instabilities, topology optimization is typically performed on structured finite element meshes (e.g., uniform grids), which may lead to mesh-dependent solutions (Antonietti et al. 2017) and limited ability to discretize complex design domains.

In recent studies, polygonal finite elements have been shown to be effective in suppressing checkerboard patterns and reducing mesh dependency in the solutions of topology optimization (Talischi et al. 2009, 2010 2012a; Gain et al. 2015; Antonietti et al. 2017). Moreover, as compared to standard finite element method (FEM) on uniform grids, polygonal elements are more versatile in discretizing complex domains. To this effect, a robust mesh generator named `PolyMesher`, able to discretize arbitary 2D domains with polygonal elements, has been developed (Talischi et al. 2012b). Other efforts in developing polygonal and polyhedral meshers include the works by Abdelkader et al. (2018) and Pouderoux et al. (2017). In addition, the geometrical flexibility of polygonal finite elements also makes them attractive for mesh adaptation in topology optimization (see, e.g., Nguyen-Xuan (2017) and Hoshina et al. (2018)). However, most of the aforementioned investigations are in 2D and efficient extensions to 3D problems poses several challenges.

The first challenge comes from the difficulties of polyhedral FEM (Hormann and Sukumar 2018). For a major difficulty is associated with obtaining the shape functions and their gradients. Although several shape functions exist in the literature with closed-form expressions, most of them are limited to certain classes of element geometry. For example, the Wachspress shape functions only work with strictly convex and simple polyhedra (meaning the collection of faces that include each vertex consists of exactly three faces) (Floater et al. 2014), and the mean value coordinates are mainly applicable to polyhedra with simplicial faces (Floater et al. 2005). Those limitations in the element geometry could potentially affect the accuracy of the polyhedral FEM when dealing with degenerated elements, such as the ones with skinny faces or small edges. Other types of shape functions, such as harmonic (Martin et al. 2008; Bishop 2014) and max-entropy (Arroyo and Ortiz 2006; Hormann and Sukumar 2008), allow for more general polyhedra (e.g., concave ones); however, their values and gradients at integration points can only be computed numerically, which could be undesirable from a computational perspective, especially for large-scale problems. Another difficulty of the polyhedral FEM is associated with numerical integration (Talischi and Paulino 2014; Manzini et al. 2014; Bishop 2014). Because there is no iso-parametric mapping for polyhedral finite elements, numerical integration needs to be performed in the physical domain. Due to the non-polynomial nature of their shape functions, efficient yet consistent numerical integration rules are difficult to construct on general polyhedral finite elements. In general, a prohibitively large number of integration points in each element is required to ensure convergence of the numerical solution (see Talischi et al. (2015) and Chi et al. (2016, 2015) for some recent research attempting to overcome this difficulty).

In this work, we identify another major challenge to use polyhedral meshes in 3D topology optimization, which is related to computational efficiency. As mentioned in the preceding paragraphs, typically topology optimization adopts an element-wise constant density approximation and a lower-order displacement approximation with the degrees

of freedom (DOFs) located at the vertices of the mesh. As a polyhedral mesh (e.g., centroidal Voronoi tessellation (CVT)) usually contains significantly more vertices than elements, this typical choice of density and displacement approximations becomes considerably less computationally economical on polyhedral meshes. In particular, when compared to uniform grids with a similar number of elements (thus similar number of densities and design variables), we need to solve a much larger structural system in each optimization step if a polyhedral mesh is considered.

The virtual element method (VEM) (Beirão da Veiga et al. 2013) is a recently proposed approach that has the potential to overcome the difficulties of the polyhedral FEM. *The VEM can be viewed as a generalization of the FEM that is able to effectively handle arbitrary polygonal and polyhedral meshes.* One main feature of the VEM is that its shape functions are defined implicitly through a suitable set of partial differential equations (PDEs). Instead of solving the PDEs for the values of shape functions and their gradients at the integration points, the VEM constructs a set of projection operators which project the shape functions and their gradients onto polynomial functions of suitable orders (Beirão da Veiga et al. 2013, 2014). By construction, these projections can be exactly computed using only the DOFs of the unknown fields. Another major feature of the VEM is that it decomposes the weak form into consistency and stability terms, both of which can be directly formed using the projections of the shape functions and their gradients as well as the DOFs of the unknown fields (Beirão da Veiga et al. 2013). As a result, for any element geometry, the VEM only needs to integrate polynomials (and not non-polynomials as in the polyhedral FEM). *For lower-order VEM, which is the focus of this paper, no numerical integration is needed.* Because of such attractive features, the VEM has gained significant visibility in the computational mechanics community. For instance, the VEM has been developed for linear elasticity (Beirão da Veiga et al. 2013; Gain et al. 2014; Artioli et al. 2017), small deformation non-linear elasticity and inelasticity (Beirão da Veiga et al. 2015; Artioli et al. 2017; Taylor and Artioli 2018), finite deformation elasticity and elasto-plasticity (Chi et al. 2017; Wriggers et al. 2017, 2018; Wriggers and Hudobivnik 2017), plate bending (Brezzi and Marini 2013; Antonietti et al. 2018; Mora et al. 2018; Zhao et al. 2016), and damage and fracture problems (De Bellis et al. 2018; Benedetto et al. 2018), to name a few. We also remark that the VEM has been adopted to solve the state equations in topology optimization on unstructured polygonal (Antonietti et al. 2017) and polyhedral (Gain et al. 2015) meshes as well as on Escher-based tessellations (Paulino and Gain 2015).

In this work, we propose a VEM-based topology optimization framework on general polyhedral discretizations. To address the aforementioned challenges of efficiently formulating topology optimization on polyhedral meshes, we adopt the VEM in both the structural and the optimization problems. Similarly to Gain et al. (2015), we use the VEM to solve the elasticity equation in the structural problem. The capability of the VEM in handling arbitrary element geometry allows us to solve the structural problem more efficiently (i.e., no numerical integration is needed) and robustly (i.e., with respect to degenerated elements) on polyhedral meshes. Exploiting the flexibility of VEM in defining local spaces, this work also introduces an enriched continuous approximation of material densities using nodal VEM basis functions. As compared to the standard element-wise constant density approximation, the continuous approximation contains a greater number of DOFs for any given polyhedral mesh and can thus improve the quality of structural topology parameterizations. Moreover, this work explores various approaches of discretizing complex domains in 3D, including regular polyhedra-dominated and unstructured polyhedral meshes, and investigates their influences on the quality and the numerical stability of solutions in the topology optimization. Several design examples are presented on non-Cartesian domains to demonstrate the main features of the VEM-based topology optimization framework. To complement the library of educational codes (e.g., Sigmund (2001), Andreassen et al. (2011), Talischi et al. (2012a), Liu and Tovar (2014), Pereira et al. (2016), Wei et al. (2018), and Sutton (2017)) in the topology optimization and VEM literature, the source code for a MATLAB implementation, named `PolyTop3D`, is provided as (electronic) Supplementary Material.

The ideation of this paper is motivated by the pioneering work of Prof. Luiz Eloy Vaz, who contributed to advance the fields of numerical analysis and optimization (Argyris et al. 1978; Vaz and Hinton 1995). The remainder of this paper is organized as follows. Section 2 provides an overview of the theory and implementation of VEM for 3D linear elasticity problems. Section 3 introduces the proposed VEM-based topology optimization together with a simple numerical example comparing the performance of the proposed continuous density parametrization with the standard element-wise constant one. In Section 4, we present a set of design examples featuring non-Cartesian domains to highlight the main features of the proposed VEM-based topology optimization framework. Section 5 contains concluding remarks and future research directions. In the Appendix, the implementation of the `PolyTop3D` is presented and the computational efficiency of the computer code is demonstrated.

2 VEM basics: theory and implementation

We consider an elastic solid $\Omega \in \mathbb{R}^3$ with its boundary denoted by $\partial\Omega$. The solid is subjected to a prescribed displacement $\mathbf{u}^0$ on one portion of the boundary $\Gamma^{\mathbf{u}}$ and a traction $\mathbf{t}$ on the other portion $\Gamma^{\mathbf{t}}$, such that $\Gamma^{\mathbf{u}} \cup \Gamma^{\mathbf{t}} = \partial\Omega$ and $\Gamma^{\mathbf{u}} \cap \Gamma^{\mathbf{t}} = \emptyset$. In the interior of Ω, the solid is subjected to a body force $\mathbf{f}$. For a given displacement field $\mathbf{u}$, the linearized strain tensor $\boldsymbol{\varepsilon}(\mathbf{u})$ is obtained as $\boldsymbol{\varepsilon}(\mathbf{u}) = 1/2[\nabla\mathbf{u} + (\nabla\mathbf{u})^T]$, where ∇ stands for the gradient operator. The stress tensor is given by means of the constitutive relationship:

$$\boldsymbol{\sigma} = \mathbf{C}\boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{C}$ is the usual elasticity tensor.

The weak form of the elasticity problem consists of finding the displacement $\mathbf{u}$ in the space $\mathcal{K}$ of kinematically admissible displacements:

$$a(\mathbf{u}, \mathbf{v}) = \ell(\mathbf{v}), \quad \forall \mathbf{v} \in \mathcal{K}^0, \tag{2}$$

where

$$a(\mathbf{u}, \mathbf{v}) = \int_\Omega [\mathbf{C}\boldsymbol{\varepsilon}(\mathbf{u})] : \boldsymbol{\varepsilon}(\mathbf{v}) \mathrm{d}\mathbf{x} \quad \ell(\mathbf{v}) = \int_\Omega \mathbf{f}\cdot\mathbf{v}\mathrm{d}\mathbf{x} + \int_{\Gamma^{\mathbf{t}}} \mathbf{t}\cdot\mathbf{v}\mathrm{d}s, \tag{3}$$

and $\mathcal{K}^0$ stands for the space of kinematically admissible displacements that vanish on $\Gamma^{\mathbf{u}}$.

2.1 Virtual spaces on polygonal and polyhedral elements

For a general polyhedral element E consisting of planar faces, this subsection describes the construction of the local virtual space $\mathcal{V}(E)$ following the technique introduced in Beirão da Veiga et al. (2018) and Ahmad et al. (2013). Accordingly, the construction of the virtual space for E depends on the virtual spaces on the faces of E. Therefore, before introducing the construction of the virtual space on E, we will first describe the construction of virtual spaces on faces of E. We also note that, in the literature, there exist different approaches to construct local virtual spaces on general polyhedral elements. The interested readers are referred to Ahmad et al. (2013), Beirão da Veiga et al. (2017), Gain et al. (2014), and Chi et al. (2017) for further information.

2.1.1 Virtual space on polygonal faces

Here, we introduce the definition of the virtual space $\mathcal{V}(f)$ on a generic face f of E, which is assumed to be a planar polygon. The basic idea is to first introduce a preliminary space denoted by $\widetilde{\mathcal{V}}(f)$ as:

$$\widetilde{\mathcal{V}}(f) = \left\{ v \in \mathcal{H}^1(f) : v|_e \in \mathcal{P}_1(e) \ \forall e \in f, \ \Delta v \in \mathcal{P}_0(f) \right\}, \tag{4}$$

where e denotes a generic edge of face f and $\mathcal{P}_k(\cdot)$ is the polynomial space of order k. For the preliminary space, $\widetilde{\mathcal{V}}(f)$, a set of DOFs consists of:

- the values of v at vertices of f, (5)
- the mean value of v over f, i.e. $\dfrac{1}{|f|}\displaystyle\int_f v\mathrm{d}\mathbf{x}$, (6)

where $|f|$ stands for the area of face f. By the following identity:

$$2\int_f v\mathrm{d}\mathbf{x} = \int_f v\mathrm{div}\,\mathbf{x}^f\mathrm{d}\mathbf{x} = -\int_f \nabla v\cdot\mathbf{x}^f\mathrm{d}\mathbf{x} + \int_{\partial f} v\,\mathbf{x}^f\cdot\mathbf{n}\mathrm{d}s, \tag{7}$$

we can, equivalently, replace the DOF (6) by the following integral (Beirão da Veiga et al. 2017):

$$\int_f \nabla v\cdot\mathbf{x}^f\mathrm{d}\mathbf{x}, \tag{8}$$

where $\mathbf{x}^f \doteq \mathbf{x}–\mathbf{x}_c^f$ with $\mathbf{x}_c^f$ being the centroid of the face f. In fact, once we know the above integral, we can compute the mean value of v over f using the DOFs (5) and the identity (7).

Having defined the preliminary virtual space $\widetilde{\mathcal{V}}(f)$, we can define the formal virtual space $\mathcal{V}(f) \subset \widetilde{\mathcal{V}}(f)$ on face f such that:

$$\mathcal{V}(f) = \left\{ v \in \mathcal{H}^1(f) : v|_e \in \mathcal{P}_1(e) \ \forall e \in f, \ \Delta v \in \mathcal{P}_0(f), \right.$$
$$\left. \text{and} \int_f \nabla v\cdot\mathbf{x}^f\mathrm{d}\mathbf{x} = 0 \right\}. \tag{9}$$

By definition, we can show that $\mathcal{P}_1 \subseteq \mathcal{V}(f)$ and that (5) constitutes a complete set of DOFs of $\mathcal{V}(f)$. Using this set of DOFs, we can exactly compute the moment of v on f according to the identity (7) as:

$$\int_f v\mathrm{d}\mathbf{x} = \frac{1}{2}\int_{\partial f} v\mathbf{x}^f\cdot\mathbf{n}\mathrm{d}s = \frac{1}{2}\sum_{e\in\partial f}\int_e v\,\mathbf{x}^f\cdot\mathbf{n}^e\mathrm{d}s, \tag{10}$$

where $\mathbf{n}^e$ denotes the outward normal vector of edge e. Noticing that $\mathbf{x}^f\cdot\mathbf{n}^e = (\mathbf{x}-\mathbf{x}_c^f)\cdot\mathbf{n}^e$ takes constant value for any points on the edge e (which is assumed to be straight), we can simply evaluate it at any point $\mathbf{a}^e$ on e, i.e., $\mathbf{x}^f\cdot\mathbf{n}^e = (\mathbf{a}^e - \mathbf{x}_c^f)\cdot\mathbf{n}^e$, $\forall\mathbf{a}^e \in e$.

2.1.2 Virtual space on polyhedra

Once we know the virtual space on each face f, we are ready to define the virtual element space $\mathcal{V}(E)$ on E. Following the same concept above, we define the virtual space $\mathcal{V}(E)$ as:

$$\mathcal{V}(E) = \left\{ v \in \mathcal{H}^1(E) : v|_f \in \mathcal{V}(f) \ \forall f \in \partial E, \right.$$
$$\left. \Delta v \in \mathcal{P}_0(E), \int_E \nabla v\cdot\mathbf{x}^E\mathrm{d}\mathbf{x} = 0 \right\}, \tag{11}$$

where $\mathbf{x}^E \doteq \mathbf{x} - \mathbf{x}_c^E$ with $\mathbf{x}_c^E$ being the centroid of E. Similarly to $\mathcal{V}(f)$, we can define the set of DOFs of $\mathcal{V}(E)$ as the

values of its functions on the vertices of E. Because $\mathcal{V}(E)$ includes the polynomial space $\mathcal{P}_1(E)$, i.e., $\mathcal{P}_1(E) \subseteq \mathcal{V}(E)$, we can define a projection operator $\Pi_E^\nabla : \mathcal{V}(E) \to \mathcal{P}_1(E)$ such that for any $v \in \mathcal{V}(E)$:

$$\int_E \nabla\left(\Pi_E^\nabla v\right) \cdot \nabla p_1 \mathrm{d}\mathbf{x} = \int_E \nabla v \cdot \nabla p_1 \mathrm{d}\mathbf{x} \quad \forall p_1 \in \mathcal{P}_1(E)$$
$$\text{and} \quad \sum_{\mathbf{x}_v \in E} v(\mathbf{x}_v) = \sum_{\mathbf{x}_v \in E} \Pi_E^\nabla v(\mathbf{x}_v), \tag{12}$$

where we use $\mathbf{x}_v$ to denote the position vector of a generic vertex in E. Because both $\nabla\Pi_E^\nabla v$ and ∇p_1 are constant vectors, the first condition in (12) can be simplified as:

$$\nabla(\Pi_E^\nabla v) = \frac{1}{|E|}\int_E \nabla v \mathrm{d}\mathbf{x} = \frac{1}{|E|}\sum_{f\in\partial E}\int_f v\,\mathbf{n}^f \mathrm{d}f, \tag{13}$$

where $|E|$ denotes the volume of element E and $\mathbf{n}^f$ is the (constant) outward unit normal vector of face f. This condition ensures that the gradient of the projection $\Pi_E^\nabla v$ equals the average gradient of v over E. Recall from the definition of $\mathcal{V}(f)$ that, given the value of v at the vertices of f, we can compute the moment of v over f (see (10)), and consequently, we can explicitly compute the boundary integral on the right hand side of (13) using the DOFs of v and geometric information of E. On the other hand, the second condition in (12) determines the constant component of the projection by ensuring that, when evaluated at the vertices of E, the average value of $\Pi_E^\nabla v$ is equal to the average value of v. Furthermore, making use of the following identity:

$$3\int_E v\mathrm{d}\mathbf{x} = \int_E v\,\mathrm{div}\,\mathbf{x}^E \mathrm{d}\mathbf{x} = -\int_E \nabla v\cdot\mathbf{x}^E \mathrm{d}\mathbf{x} + \int_{\partial E} v\,\mathbf{x}^E\cdot\mathbf{n}\mathrm{d}s, \tag{14}$$

and the definition of $\mathcal{V}(E)$, we can express the moment of v over E as:

$$\int_E v\mathrm{d}\mathbf{x} = \frac{1}{3}\sum_{f\in\partial E}\int_f v\mathbf{x}^E\cdot\mathbf{n}^f \mathrm{d}f. \tag{15}$$

Realizing that $\mathbf{x}^E \cdot \mathbf{n}^f = \left(\mathbf{x} - \mathbf{x}_c^E\right)\cdot\mathbf{n}^f$ is constant for any points $\mathbf{x}$ on planar f, we can simply evaluate this quantity at any point $\mathbf{a}^f$ on face f, i.e. $\mathbf{x}^E = (\mathbf{a}^f - \mathbf{x}_c^E)\cdot\mathbf{n}^f$, $\forall \mathbf{a}^f \in f$. Thus, we can compute the moment of v over E exactly as:

$$\int_E v\mathrm{d}\mathbf{x} = \frac{1}{3}\sum_{f\in\partial E}\left(\mathbf{x}^E\cdot\mathbf{n}^f\right)\int_f v\mathrm{d}f, \tag{16}$$

using only the DOFs of v (recall (10)).

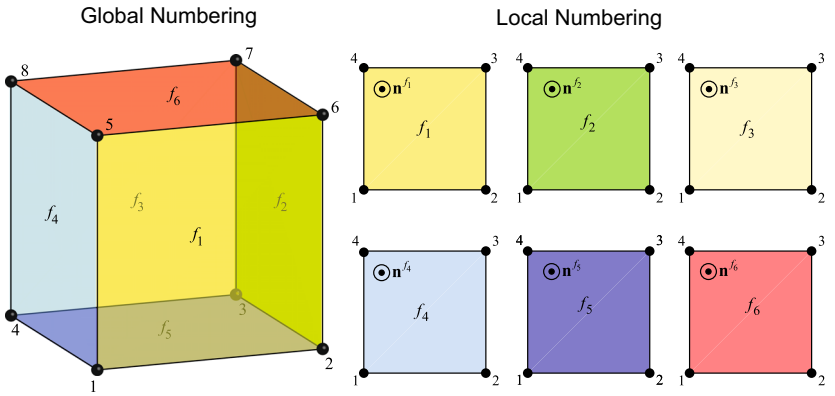
2.1.3 Some implementation details

Consider a polyhedron E consisting of m vertices numbered as $\mathbf{x}_1, ..., \mathbf{x}_m$ and denote $\mathcal{F}_i$ as the set of faces that are connected to the ith vertex. Suppose that a face $f \in \mathcal{F}_i \in \mathbb{R}^3$ has m^f vertices $\mathbf{x}_j^f$, $j = 1, ..., m^f$, we locally renumber those vertices in a counterclockwise fashion with respect to the outward normal $\mathbf{n}^f$ which points out of the element. We also utilize a map $\mathcal{G}_f$ to denote the relation between the global numbering and local numbering of the vertices on the face f. If the ith vertex of E (under global numbering) is numbered as the jth vertex of f (under local numbering), we write $j = \mathcal{G}_f(i)$. Figure 1 shows an illustration of the above notation. Following this notation, this subsection provides the implementation details to construct the virtual space $\mathcal{V}(E)$. In particular, we focus on the procedures of calculating the projection $\Pi_E^\nabla\varphi_i$ and the moment $\int_E \varphi_i$, where φ_i is the basis function of $\mathcal{V}(E)$ associated with the ith vertex. For vertex i, φ_i is defined to be a function which belongs to $\mathcal{V}(E)$ and takes 1 at the ith vertex and 0 at the other vertices of E.

Based on the definition of the projection operator Π_E^∇ in (12), we can express $\Pi_E^\nabla\varphi_i$ as:

$$\Pi_E^\nabla\varphi_i = \left(\nabla\Pi_E^\nabla\varphi_i\right)\cdot\left(\mathbf{x} - \widehat{\mathbf{x}}^E\right) + \frac{1}{m}, \tag{17}$$

where $\widehat{\mathbf{x}}^E \doteq 1/m\sum_{j=1}^m \mathbf{x}_i$ is the algorithmic mean of the position vectors of the vertices of E. To compute this



Local Numbering \ Global Numbering	i	1	2	3	4	5	6	7	8
	$\mathcal{G}_{f_1}(i)$	1	2			4	3		
	$\mathcal{G}_{f_2}(i)$		1	2			4	3	
	$\mathcal{G}_{f_3}(i)$			1	2			4	3
	$\mathcal{G}_{f_4}(i)$	1			4	2			3
	$\mathcal{G}_{f_5}(i)$	1	4	3	2				
	$\mathcal{G}_{f_6}(i)$					1	2	3	4

Fig. 1 Illustration of the global and local numbering convention and the mapping, $\mathcal{G}_f(i)$, on a regular hexahedral element

projection, we first need to know the moment of φ_i on f, $\forall f \in \mathcal{F}_i$. Using relation (10) and realizing that φ_i varies linear on ∂f, we can compute the moment of φ_i on f as:

$$\begin{aligned}\int_f \varphi_i \mathrm{d}f &= \frac{1}{2}\sum_{e\in\partial f}(\mathbf{x}^f\cdot\mathbf{n})^e \int_e \varphi_i \mathrm{d}s \\ &= \frac{1}{2}\sum_{e\in\partial f}\left(\mathbf{x}_i-\mathbf{x}_c^f\right)\cdot\left(\mathbf{n}^e\int_e\varphi_i\mathrm{d}s\right) \\ &\qquad \text{(evaluate } \mathbf{x}^f=\mathbf{x}-\mathbf{x}_c^f \text{ at vertex } \mathbf{x}_i) \\ &= \frac{1}{4}(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1})\wedge\mathbf{n}^f\cdot\left(\mathbf{x}_i-\mathbf{x}_c^f\right), \end{aligned} \tag{18}$$

in which $\wedge$ stands for the cross product, where the following convention is used: $\mathcal{G}_f(\cdot)+1=1$ whenever $\mathcal{G}_f(\cdot)=m^f$, and $\mathcal{G}_f(\cdot)-1=m^f$ whenever $\mathcal{G}_f(\cdot)=1$. Substituting the above relation into (13), we can show that:

$$\begin{aligned}\nabla\Pi_E^\nabla\varphi_i &= \frac{1}{|E|}\sum_{f\in\mathcal{F}_i}\int_f\varphi_i\,\mathbf{n}^f\mathrm{d}f = \frac{1}{4|E|}\sum_{f\in\mathcal{F}_i}\left(\mathbf{x}_i-\mathbf{x}_c^f\right) \\ &\quad\wedge(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1})\end{aligned} \tag{19}$$

and, by (17), we finally arrive at the following expression:

$$\begin{aligned}\Pi_E^\nabla\varphi_i &= \frac{1}{4|E|}\sum_{f\in\mathcal{F}_i}(\mathbf{x}_i-\mathbf{x}_c^f)\wedge(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1}) \\ &\quad\cdot(\mathbf{x}-\widehat{\mathbf{x}}^E)+\frac{1}{m}.\end{aligned} \tag{20}$$

Moreover, since $\Pi_E^\nabla\varphi_i \in \mathcal{P}_1(E)\subseteq\mathcal{V}(E)$, the set of basis function φ_i can interpolate any linear function exactly, namely, $1=\sum_{i=1}^m\varphi_i$ and $\mathbf{x}=\sum_{i=1}^m\varphi_i\mathbf{x}_i$. Therefore, we can rewrite the relation (20) as a linear combination of the set of basis functions $\varphi_1,...,\varphi_m$ as:

$$\begin{aligned}\Pi_E^\nabla\varphi_i &= \frac{1}{4|E|}\sum_{f\in\mathcal{F}_i}(\mathbf{x}_i-\mathbf{x}_c^f)\wedge(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1}) \\ &\quad\cdot\left(\sum_{j=1}^m\varphi_j\mathbf{x}_j-\sum_{j=1}^m\varphi_j\widehat{\mathbf{x}}^E\right)+\frac{\sum_{j=1}^m\varphi_j}{m} \\ &= \sum_{j=1}^m\left\{\frac{1}{4|E|}\sum_{f\in\mathcal{F}_i}\left(\mathbf{x}_i-\mathbf{x}_c^f\right)\wedge(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1})\right. \\ &\quad\left.\cdot(\mathbf{x}_j-\widehat{\mathbf{x}}^E)+\frac{1}{m}\right\}\varphi_j.\end{aligned} \tag{21}$$

The above expression can be further simplified in matrix form as:

$$\Pi_E^\nabla\varphi_i=\sum_{j=1}^m\mathbf{P}_{(ij)}^E\varphi_j, \tag{22}$$

where $\mathbf{P}_{(ij)}^E$ is the (i,j)th components of the matrix $\mathbf{P}^E\in\mathbb{R}^{m\times m}$ of the form:

$$\begin{aligned}\mathbf{P}_{(ij)}^E &= \frac{1}{4|E|}\sum_{f\in\mathcal{F}_i}(\mathbf{x}_i-\mathbf{x}_c^f)\wedge(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1}) \\ &\quad\cdot(\mathbf{x}_j-\widehat{\mathbf{x}}^E)+\frac{1}{m}.\end{aligned} \tag{23}$$

In terms of implementation, we form $\mathbf{P}^E$ using matrix multiplication as follows. We first define $\mathbf{G}^E\in\mathbb{R}^{m\times 3}$ as a matrix collecting the information of $\nabla\Pi_E^\nabla\varphi_i$, $i=1,...,m$, i.e.:

$$\mathbf{G}^E=\begin{bmatrix}\frac{\partial\Pi_E^\nabla\varphi_1}{\partial x} & \frac{\partial\Pi_E^\nabla\varphi_1}{\partial y} & \frac{\partial\Pi_E^\nabla\varphi_1}{\partial z}\\ \vdots & \vdots & \vdots \\ \frac{\partial\Pi_E^\nabla\varphi_m}{\partial x} & \frac{\partial\Pi_E^\nabla\varphi_m}{\partial y} & \frac{\partial\Pi_E^\nabla\varphi_m}{\partial z}\end{bmatrix}, \tag{24}$$

and compute $\mathbf{P}^E$ as

$$\begin{aligned}\mathbf{P}^E &= \begin{bmatrix}x_1-\frac{1}{m}\sum_{i=1}^m x_i & y_1-\frac{1}{m}\sum_{i=1}^m y_i & z_1-\frac{1}{m}\sum_{i=1}^m z_i & 1\\ \vdots & \vdots & \vdots & \vdots \\ x_m-\frac{1}{m}\sum_{i=1}^m x_i & y_m-\frac{1}{m}\sum_{i=1}^m y_i & z_m-\frac{1}{m}\sum_{i=1}^m z_i & 1\end{bmatrix} \\ &\quad\times\left[\mathbf{G}^E\ \ \frac{1}{m}\mathbf{1}\right]^T,\end{aligned} \tag{25}$$

where $\mathbf{1}\in\mathbb{R}^{m\times 1}$ is a column vector with all components equal to 1.

For later use, we also provide the expression to compute the moment of φ_i over E, which, according to expressions (16) and (18), takes the form:

$$\begin{aligned}\int_E\varphi_i\mathrm{d}\mathbf{x} &= \frac{1}{3}\sum_{f\in\mathcal{F}_i}(\mathbf{x}^E\cdot\mathbf{n}^f)\int_f\varphi_i\mathrm{d}f \\ &= \frac{1}{3}\sum_{f\in\mathcal{F}_i}\left[(\mathbf{x}_i-\mathbf{x}_c^E)\cdot\mathbf{n}^f\right]\int_f\varphi_i\mathrm{d}f \\ &\qquad \text{(evaluate } \mathbf{x}^E=\mathbf{x}-\mathbf{x}_c^E \text{ at vertex } \mathbf{x}_i) \\ &= \frac{1}{12}\sum_{f\in\mathcal{F}_i}\left[(\mathbf{x}_i-\mathbf{x}_c^E)\cdot\mathbf{n}^f\right]\left[(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1})\wedge\mathbf{n}^f\right. \\ &\quad\left.\cdot(\mathbf{x}_i-\mathbf{x}_c^f)\right] \quad \text{(using (18))} \\ &= \frac{1}{12}\sum_{f\in\mathcal{F}_i}(\mathbf{x}_{\mathcal{G}_f(i)+1}-\mathbf{x}_{\mathcal{G}_f(i)-1})\wedge(\mathbf{x}_i-\mathbf{x}_c^E)\cdot(\mathbf{x}_i-\mathbf{x}_c^f).\end{aligned} \tag{26}$$

2.2 VEM approximations for 3D linear elasticity

We consider a discretization, denoted by Ω_h, of the solid Ω into non-overlapping polyhedra consisting of planar faces, where h denotes the average element size. We denote $\Gamma_h^{\mathbf{t}}$ and $\Gamma_h^{\mathbf{u}}$ as the portions of the mesh boundary where the traction and displacement boundary conditions are applied, respectively.

2.2.1 Theoretical background

We define the discrete global displacement space $\mathcal{K}_h \subset \mathcal{K}$ as:

$$\mathcal{K}_h \doteq \left\{ \mathbf{v}_h \in \mathcal{K} : \mathbf{v}_h|_E \in [\mathcal{V}(E)]^3 \right\}. \tag{27}$$

Over each element E, the local displacement $\mathbf{v} = [v_x, v_y, v_z]^T$ belongs to the vectorial space $[\mathcal{V}(E)]^3$, having three displacement DOFs for each vertex of E. In the following discussion, we define the projection operator for vector fields $\mathbf{\Pi}_E^\nabla : [\mathcal{V}(E)]^3 \to [\mathcal{P}_1(E)]^3$ as the action of Π_E^∇ on every component of the vector field, e.g., $\mathbf{\Pi}_E^\nabla \mathbf{v} = [\Pi_E^\nabla v_x, \Pi_E^\nabla v_y, \Pi_E^\nabla v_z]^T$.

Having introduced the discretization, we can decompose the continuous bilinear form (i.e. bilinear form with exact integration) $a(\mathbf{u}_h, \mathbf{v}_h)$ into the summation of element-level contributions:

$$a(\mathbf{u}_h, \mathbf{v}_h) = \sum_E a^E(\mathbf{u}_h, \mathbf{v}_h) = \sum_E \int_E \boldsymbol{\varepsilon}(\mathbf{u}_h) : [\mathbf{C}\boldsymbol{\varepsilon}(\mathbf{v}_h)] \mathrm{d}\mathbf{x}. \tag{28}$$

By exploiting the property that the projection $\mathbf{\Pi}_E^\nabla \mathbf{v}$ satisfies the following orthogonality condition:

$$a^E\left(\mathbf{p}_1, \mathbf{v} - \Pi_E^\nabla \mathbf{v}\right) = 0 \quad \forall \mathbf{p}_1 \in [\mathcal{P}_1(E)]^3 \text{ and } \mathbf{v} \in [\mathcal{V}(E)]^3, \tag{29}$$

then we can decompose $a^E(\mathbf{u}_h, \mathbf{v}_h)$ as:

$$\begin{aligned} a^E(\mathbf{u}_h, \mathbf{v}_h) &= a^E\left(\mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{\Pi}_E^\nabla \mathbf{v}_h\right) \\ &\quad + a^E\left(\mathbf{u}_h - \mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{v}_h - \mathbf{\Pi}_E^\nabla \mathbf{v}_h\right). \end{aligned} \tag{30}$$

Because both of its arguments are linear functions, we can evaluate the first term exactly as:

$$a^E(\mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{\Pi}_E^\nabla \mathbf{v}_h) = |E| \boldsymbol{\varepsilon}(\mathbf{\Pi}_E^\nabla \mathbf{u}_h) : \left[\mathbf{C}\boldsymbol{\varepsilon}(\mathbf{\Pi}_E^\nabla \mathbf{v}_h)\right]. \tag{31}$$

The second term, on the other hand, involves higher order displacement components and is typically evaluated numerically. In the VEM framework, this term is approximated by:

$$\begin{aligned} &a^E\left(\mathbf{u}_h - \mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{v}_h - \mathbf{\Pi}_E^\nabla \mathbf{v}_h\right) \\ &\approx \alpha^E S^E\left(\mathbf{u}_h - \mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{v}_h - \mathbf{\Pi}_E^\nabla \mathbf{v}_h\right), \end{aligned} \tag{32}$$

where $S^E(\cdot,\cdot)$ is a bilinear form that is inexpensive to compute and satisfies the coercivity condition; and α^E is a scaling parameter that scales $S^E(\cdot,\cdot)$ to the same order of magnitude as $a^E(\cdot,\cdot)$. Typical choices of $S^E(\cdot,\cdot)$ and α^E take the form:

$$\begin{aligned} &S^E(\mathbf{u}_h, \mathbf{v}_h) = h_E \sum_{\mathbf{x}_v \in E} \mathbf{u}_h(\mathbf{x}_v) \cdot \mathbf{v}_h(\mathbf{x}_v) \text{ and} \\ &\alpha^E = \frac{\operatorname{trace} \mathbf{C}}{9} = \frac{C_{ijij}}{9} \text{ (in indical notation),} \end{aligned} \tag{33}$$

where $h_E \doteq |E|^{1/3}$ represents the size of element E and $\mathbf{x}_v$ stands for the vertices that belong to E. This gives the final form of the element-level discrete bilinear form as:

$$\begin{aligned} a_h^E(\mathbf{u}_h, \mathbf{v}_h) &= |E| \boldsymbol{\varepsilon}\left(\mathbf{\Pi}_E^\nabla \mathbf{u}_h\right) : \left[\mathbf{C}\boldsymbol{\varepsilon}\left(\mathbf{\Pi}_E^\nabla \mathbf{v}_h\right)\right] \\ &\quad + \alpha^E S^E\left(\mathbf{u}_h - \mathbf{\Pi}_E^\nabla \mathbf{u}_h, \mathbf{v}_h - \mathbf{\Pi}_E^\nabla \mathbf{v}_h\right). \end{aligned} \tag{34}$$

The first and second terms of $a_h^E(\mathbf{u}_h, \mathbf{v}_h)$ are respectively known as the consistency and stability terms, and they are responsible for the satisfaction of the two key conditions, namely consistency and stability, respectively, to ensure the convergence of the VEM approximation (Beirão da Veiga et al. 2013).

On the other hand, we approximate the continuous loading term (i.e., the loading term with exact integration) $\ell(\mathbf{v}_h)$ as (Chi et al. 2017):

$$\begin{aligned} \ell_h(\mathbf{v}_h) &= \sum_{f \in \Gamma_h^{\mathbf{t}}} |f| \mathbf{t}\left(\mathbf{x}_c^f\right) \cdot \left(\mathbf{\Pi}_f^\nabla \mathbf{v}_h\right)\left(\mathbf{x}_c^f\right) \\ &\quad + \sum_{E \in \Omega_h} \mathbf{f}\left(\mathbf{x}_c^E\right) \cdot \left(\mathbf{\Pi}_E^\nabla \mathbf{v}_h\right)\left(\mathbf{x}_c^E\right), \end{aligned} \tag{35}$$

where $\mathbf{\Pi}_f^\nabla \mathbf{v}_h$ is the projection of $\mathbf{v}_h|_f$ onto $[\mathcal{P}_1(f)]^3$ defined in the same way as $\mathbf{\Pi}_E^\nabla \mathbf{v}_h$; and $\mathbf{t}\left(\mathbf{x}_c^f\right)$ and $\mathbf{f}\left(\mathbf{x}_c^E\right)$ are the values of traction and body forces evaluated at $\mathbf{x} = \mathbf{x}_c^f$ and $\mathbf{x} = \mathbf{x}_c^E$, respectively (we recall that $\mathbf{x}_c^f$ and $\mathbf{x}_c^E$ are the centroids of face f and element E, respectively). The above approximation essentially utilizes one-point rules on face f and element E, both of which are exact for integrating any linear function.

We are now ready to state the final form of the VEM approximation for 3D linear elasticity problems, which consists of finding $\mathbf{u}_h \in \mathcal{K}_h$ such that:

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = \sum_E a_h^E(\mathbf{u}_h, \mathbf{v}_h) = \ell_h(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathcal{K}_h^0, \tag{36}$$

where $\mathcal{K}_h^0$ is a subspace of $\mathcal{K}_h$ with functions that vanish on $\Gamma_h^{\mathbf{u}}$.

2.2.2 A few implementation details

For a given element E, we consider a set of basis functions, $\boldsymbol{\varphi}_1, ..., \boldsymbol{\varphi}_{3m}$, for the local displacement space $[\mathcal{V}(E)]^3$ of the form

$$\begin{aligned} \boldsymbol{\varphi}_{3i-2} &= [\varphi_i, 0, 0]^T, \quad \boldsymbol{\varphi}_{3i-1} = [0, \varphi_i, 0]^T, \\ \boldsymbol{\varphi}_{3i} &= [0, 0, \varphi_i]^T, \quad i = 1, ..., m \end{aligned} \tag{37}$$

where we recall that $\varphi_1, ..., \varphi_m$ is the set of basis functions for $\mathcal{V}(E)$. With the basis functions, any displacement field

$\mathbf{v} = [v_x, v_y, v_z]^T \in [\mathcal{V}(E)]^3$ can be interpolated using its DOFs (the values at the vertices) as:

$$\mathbf{v} = \sum_{i=1}^{m} \left[\boldsymbol{\varphi}_{3i-2} v_x(\mathbf{x}_i) + \boldsymbol{\varphi}_{3i-1} v_y(\mathbf{x}_i) + \boldsymbol{\varphi}_{3i} v_z(\mathbf{x}_i) \right]. \quad (38)$$

Moreover, by definition, the projection $\boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_j$, $j = 1, ..., 3m$ is given by:

$$\begin{aligned} \boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_{3i-2} &= [\Pi_E^{\nabla} \varphi_i, 0, 0]^T, \quad \boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_{3i-1} = [0, \Pi_E^{\nabla} \varphi_i, 0]^T, \\ \boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_{3i} &= [0, 0, \Pi_E^{\nabla} \varphi_i]^T, \quad i = 1, ..., m. \end{aligned} \quad (39)$$

According to (34), we evaluate the (j, k)th component of the element stiffness matrix $\mathbf{k}^E \in \mathbb{R}^{3m \times 3m}$ as:

$$\begin{aligned} \mathbf{k}^E_{(jk)} &= a_h^E(\boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k) = |E| \boldsymbol{\varepsilon}\left(\boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_j\right) : \left[\mathbf{C} \boldsymbol{\varepsilon}\left(\boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_k\right) \right] \\ &\quad + \alpha^E S^E \left(\boldsymbol{\varphi}_j - \boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k - \boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_k \right). \end{aligned} \quad (40)$$

More specifically, we can define matrice $\mathbf{B}^E \in \mathbb{R}^{6 \times 3m}$ and $\mathbf{D} \in \mathbb{R}^{6 \times 6}$ of the form

$$\mathbf{B}^E = \begin{bmatrix} \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial x} & 0 & 0 & \dots & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial x} & 0 & 0 \\ 0 & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial y} & 0 & \dots & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial y} & 0 \\ 0 & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial z} & \dots & 0 & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial z} \\ \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial y} & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial x} & 0 & \dots & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial y} & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial x} & 0 \\ 0 & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial z} & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial y} & \dots & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial z} & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial y} \\ \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial z} & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_1}{\partial x} & \dots & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial z} & 0 & \frac{\partial \Pi_E^{\nabla} \varphi_m}{\partial x} \end{bmatrix} \quad (41)$$

$$\mathbf{D} = \begin{bmatrix} C_{1111} & C_{1122} & C_{1133} & C_{1112} & C_{1123} & C_{1113} \\ & C_{2222} & C_{2233} & C_{2212} & C_{2223} & C_{2213} \\ & & C_{3333} & C_{3312} & C_{3323} & C_{3313} \\ & & & C_{1212} & C_{1223} & C_{1213} \\ & \text{symm.} & & & C_{2323} & C_{2313} \\ & & & & & C_{1313} \end{bmatrix}, \quad (42)$$

and rewrite the first term of (40) in matrix form as:

$$|E| \boldsymbol{\varepsilon}\left(\boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_j\right) : \left[\mathbf{C} \boldsymbol{\varepsilon}\left(\boldsymbol{\Pi}_E^{\nabla} \boldsymbol{\varphi}_k\right) \right] = |E| (\mathbf{B}^E)^T \mathbf{D} \mathbf{B}^E. \quad (43)$$

For the second term of (40), we express it in matrix notation as:

$$\alpha^E S^E \left(\boldsymbol{\varphi}_j - \Pi_E^{\nabla} \boldsymbol{\varphi}_j, \boldsymbol{\varphi}_k - \Pi_E^{\nabla} \boldsymbol{\varphi}_k \right) = \alpha^E \left(\mathbf{I} - \mathbf{S}^E \right)^T \left(\mathbf{I} - \mathbf{S}^E \right), \quad (44)$$

where $\mathbf{S}^E \in \mathbb{R}^{3m \times 3m}$ is given by

$$\mathbf{S}^E = \begin{bmatrix} \mathbf{P}^E_{(11)} & 0 & 0 & \dots & \mathbf{P}^E_{(1m)} & 0 & 0 \\ 0 & \mathbf{P}^E_{(11)} & 0 & \dots & 0 & \mathbf{P}^E_{(1m)} & 0 \\ 0 & 0 & \mathbf{P}^E_{(11)} & \dots & 0 & 0 & \mathbf{P}^E_{(1m)} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{P}^E_{(m1)} & 0 & 0 & \dots & \mathbf{P}^E_{(mm)} & 0 & 0 \\ 0 & \mathbf{P}^E_{(m1)} & 0 & \dots & 0 & \mathbf{P}^E_{(mm)} & 0 \\ 0 & 0 & \mathbf{P}^E_{(m1)} & \dots & 0 & 0 & \mathbf{P}^E_{(mm)} \end{bmatrix} \quad (45)$$

with $\mathbf{P}^E$ defined in (25).

Having computed the local stiffness matrix for each element, we can obtain the global stiffness matrix $\mathbf{K} = \sum_E \mathbf{k}^E$ through the standard assembly procedure and solve the linear system of equations:

$$\mathbf{KU} = \mathbf{F} \quad (46)$$

for the nodal displacement vector $\mathbf{U}$, where $\mathbf{F}$ is the external force vector. For a given discretization, the external force vector $\mathbf{F}$ contains the nodal loads computed from applied traction $\mathbf{t}$ and body force $\mathbf{f}$ based on (35).

2.3 Assessment of the VEM approximation for linear elasticity

Here, the performance of the VEM approximations on convergence and accuracy is assessed through a benchmark problem. In the assessment, we evaluate two error measures of the displacement and stress solutions. The measure of error in the displacement solution is defined as:

$$\epsilon_{\mathbf{u}} = \sqrt{\sum_{E \in \Omega_h} \int_E (\widetilde{\mathbf{u}}_h - \mathbf{u}) \cdot (\widetilde{\mathbf{u}}_h - \mathbf{u}) \mathrm{d}\mathbf{x}}, \quad (47)$$

where $\mathbf{u}$ is the exact displacement solution and $\widetilde{\mathbf{u}}_h$ denotes the displacement field obtained by interpolating the VEM DOFs using the 3D Wachspress shape functions (Floater et al. 2014). On the other hand, we also define the $\mathcal{L}^2$ norm of the stress error as:

$$\epsilon_{\boldsymbol{\sigma}} = \sqrt{\sum_{E \in \Omega_h} \int_E (\boldsymbol{\sigma}_h - \boldsymbol{\sigma}) \cdot (\boldsymbol{\sigma}_h - \boldsymbol{\sigma}) \mathrm{d}\mathbf{x}}, \quad (48)$$

where $\boldsymbol{\sigma}$ is the exact stress solution and $\boldsymbol{\sigma}_h$ is a piecewise constant stress field defined such that:

$$\boldsymbol{\sigma}_h|_E = \mathbf{C} \boldsymbol{\varepsilon}\left(\boldsymbol{\Pi}_E^{\nabla} \mathbf{u}_h\right). \quad (49)$$

For both error measures, the integrals are evaluated using a fourth-order integration rule on each tetrahedral subdivision of E.

We consider a boundary value problem in which a cantilever beam is loaded by end shear. As described in Fig. 2 (a), the beam occupies the domain $\Omega = (-1, 1) \times (-1, 1) \times (0, 10)$ and is subjected to a constant traction $\mathbf{t} = [0, -\tau, 0]^T$ on its top surface. According to Barber (2010), the stress solution of such a problem is given by the following expressions:

$$\begin{aligned} \sigma_{xx} &= \sigma_{yy} = \sigma_{xy} = 0, \qquad \sigma_{zz} = \frac{3\tau}{4} yz \\ \sigma_{xz} &= \frac{3\tau \nu}{2\pi^2 (1 + \nu)} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2 \cosh(n\pi)} \sin(n\pi x) \sinh(n\pi y) \end{aligned}$$

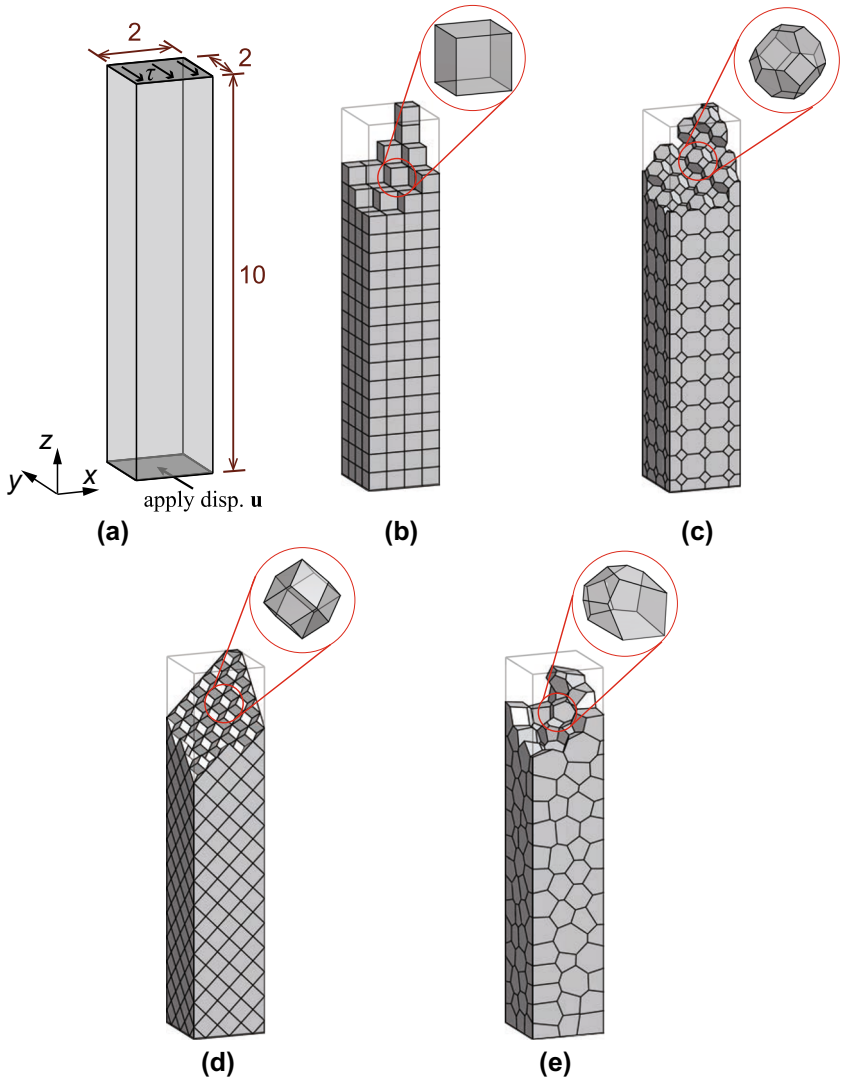


Fig. 2 **a** Problem description of a cantilever beam loaded by end shear. **b** An example of the mesh consisting of regular hexahedra. **c** An example of the mesh consisting of truncated octahedra. **d** An example of the mesh consisting of rhombic dodecahedra. **e** An example of the standard CVT mesh

$$\sigma_{yz} = \frac{3\tau(1-y^2)}{8} + \frac{\tau\nu(3x^2-1)}{8(1+\nu)} - \frac{3\tau\,\nu}{2\pi^2(1+\nu)} \sum_{n=1}^{\infty} \frac{(-1)^n}{n^2 \cosh(n\pi)} \cos(n\pi x) \cosh(n\pi y), \quad (50)$$

where E_Y and ν are Young's modulus and Poisson's ratio, respectively. For the stress solution, we verify that the beam is traction-free on its four lateral surfaces. Additionally, the displacement solution that corresponds to the above stress distributions, up to the addition of a rigid body motion, is given by:

$$\begin{aligned} u_x &= -\frac{3\tau\nu}{4E_Y} xyz \\ u_y &= \frac{\tau}{8E_Y}(3\nu z(x^2 - y^2) - z^3) \\ u_z &= \frac{\tau}{8E_Y}\left[3yz^2 + \nu y(y^2 - 3x^2)\right] + \frac{2(1+\nu)}{E_Y} w(\mathbf{x}), \end{aligned} \quad (51)$$

where $w(\mathbf{x})$ is the anti-derivative of σ_{yz} with respect to y. In our numerical study, the material properties of the solid are taken to be $E_Y = 25$ and $\nu = 0.3$, and the magnitude of the shear load is chosen to be $\tau = -0.1$. We apply the analytical displacement field given in (51) on the bottom surface and analytical traction (calculated from (50)) on the top surface. Three families of structured meshes made up of regular space-filling polyhedra (i.e. regular hexahedra, truncated octahedra and rhombic dodecahedra), and an additional family of CVT meshes are considered, as shown in Fig. 2 b–e.

The convergence of both displacement error, $\epsilon_{\mathbf{u}}$, and stress error, $\epsilon_{\boldsymbol{\sigma}}$, as functions of the average mesh size h, is depicted in Fig. 3 a and b, respectively. The results confirm that optimal convergence rates are obtained for both displacement and stress errors – the rate of convergence for $\epsilon_{\mathbf{u}}$ is 2 and that for $\epsilon_{\boldsymbol{\sigma}}$ is 1 (as expected).

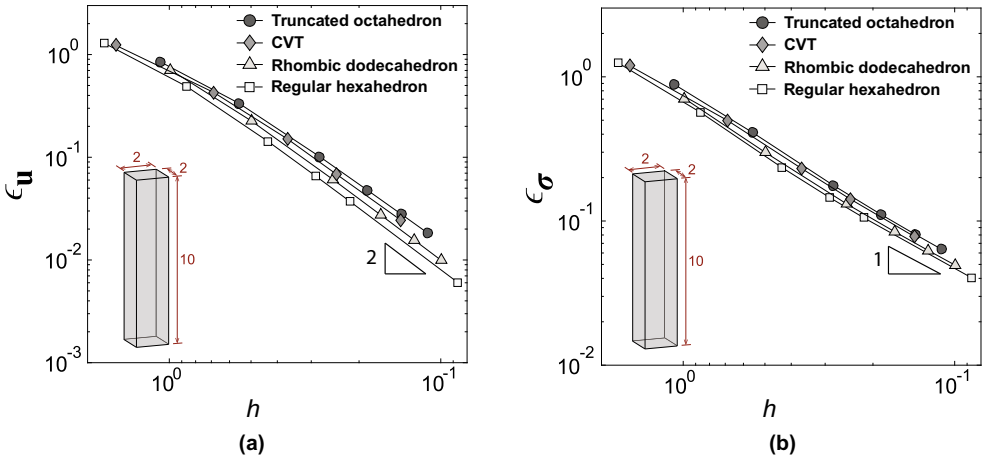


Fig. 3 Convergence of the $\mathcal{L}^2$ norms of the **c** displacement error $\epsilon_{\mathbf{u}}$ and **d** stress error ϵ_{σ} when traction is applied on the top boundary and displacement is applied on the bottom boundary of the domain

3 VEM-based topology optimization using Polytopes

This section introduces a VEM-based topology optimization framework on polyhedral meshes, which features continuous design and material density functions. We shall focus on the classic compliance minimization problems and remark that the proposed framework is readily applicable to other optimization formations (although further investigation is warranted). We also note that, from now on, the body force **f** is neglected.

For a given discretization Ω_h consisting of non-overlapping polyhedra, the topology optimization formulation for the minimum compliance problems is stated as:

$$\begin{aligned} &\inf_{\rho_h \in \mathcal{A}_h} \int_{\Gamma_h^{\mathbf{t}}} \mathbf{t} \cdot \mathbf{u}_h \\ &\text{s.t.} \quad \frac{1}{|\Omega_h|} \int_{\Omega_h} \rho_h - \overline{V} \leq 0 \\ &\text{with} \quad a_h^{\rho}(\mathbf{u}_h, \mathbf{v}_h) = \ell_h(\mathbf{v}_h) \quad \forall \mathbf{v}_h \in \mathcal{K}_h^0, \end{aligned} \tag{52}$$

where ρ_h is a material density function, $\mathcal{A}_h$ is the space of admissible designs and $\overline{V}$ is the allowable volume fraction. To regularize the formulation, the material density function ρ_h is defined as the image of a design function η_h under a map $\mathcal{F}_h$ (e.g., the density filter) and the DOFs of the design function η_h are the design variables (DVs). Moreover, a material interpolation function $m_S(\rho_h)$ is employed to relate the material stiffness to the value of ρ_h at any given point. For instance, if the SIMP model is used (Bendsøe (1989), Rozvany et al. (1992), and Bendsøe and Sigmund (1999)), we have $m_S(\rho_h) = \varepsilon + (1-\varepsilon)\rho_h^p$, where ε is the Ersatz parameter and p is the penalization parameter. Incorporating the spatially varying stiffness $m_S(\rho_h)$, $a_h^{\rho}(\mathbf{u}_h, \mathbf{v}_h)$ is the discrete bilinear form constructed using VEM.

As indicated in the introduction, the topology optimization formulation (52) can be viewed as a two-field mixed approximation problem involving a discrete displacement space $\mathcal{K}_h$ and a discrete design space $\mathcal{A}_h$ (Jog and Haber 1996). The standard (i.e. FEM-based) density-based topology optimization framework in the literature typically employs a continuous displacement field, whose DOFs are the displacements at the vertices of the mesh, and a piecewise-constant design function, whose value in each element is the associated DV. The material density within

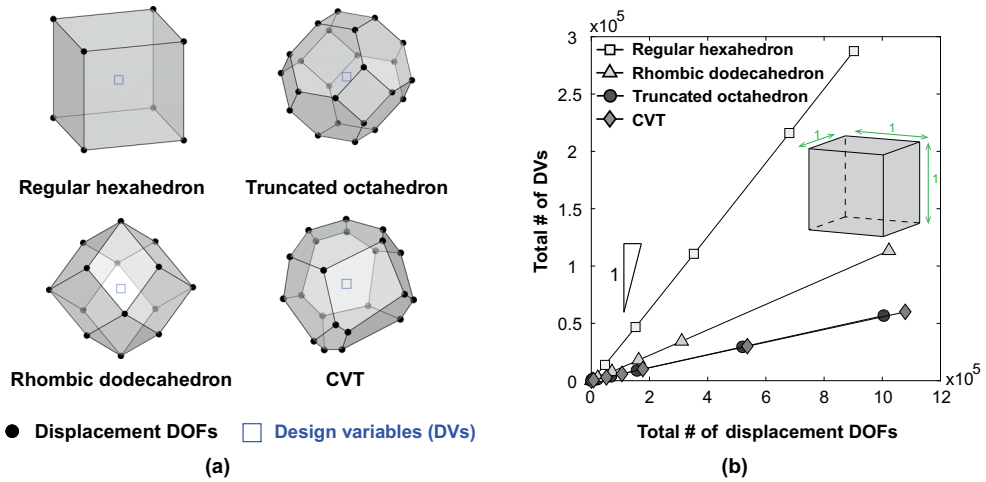


Fig. 4 **a** Illustration of the "mixed" elements adopted in the standard topology optimization framework. The displacement DOFs are located at the vertices and each element contains one DV representing the constant design field. **b** The total number of DVs as function of the total number of displacement DOFs for various discretizations of a unit cube

each element takes a constant value as well. Several examples of "mixed" elements of this type are shown in Fig. 4a. Roughly speaking, for a given discretization, the number of DVs governs the "resolution" of the topology whereas the number of the displacement DOFs (the size of the state equation) dictates the computational cost. If we consider a unit cube and discretize it with the "mixed" elements shown in Fig. 4a, we can then plot in Fig. 4b the total numbers of DVs as function of the total number of displacement DOFs when those meshes are refined. We observe that those functions are close to linear and their slopes can be used to quantify the computational efficiency of the topology optimization framework on various discretizations: the larger the slope is, the more computationally efficient the discretization is. As shown from the slopes in Fig. 4b, the "mixed" approximation in the standard topology optimization framework leads to considerably more displacement DOFs than DVs on various discretizations, which is undesirable from a computational efficiency perspective. Moreover, Fig. 4b also suggests that polyhedral discretizations yield smaller slopes as compared to the hexahedral ones. This observation indicates that, although polyhedral discretizations exhibit several geometric advantages in topology optimization (Gain et al. 2015), they are less computationally efficient than hexahedral ones in the standard topology optimization framework.

Motivated by the above observations, we propose an approximation of the design function (as well as the material density function) for topology optimization on general polyhedral meshes. The basic idea is to consider a more enriched local design space with continuous design functions in each element. *In terms of DVs, they are placed at the vertices as well as the mid-edge nodes of the meshes.* On the other hand, the displacement approximation is kept the same as in the standard case. This leads to a new "mixed" approximation for topology optimization on general polyhedral meshes. We note that the idea of enriching the design space is conceptually similar to the multi-resolution methodologies (Nguyen et al. 2010) in topology optimization. An illustration of several new "mixed" elements of this type is shown in Fig. 5a. Again, considering a unit cube discretized with those "mixed" elements, we plot in Fig. 5b the total numbers of DVs as functions of the total numbers of displacement DOFs with mesh refinement. By comparing the slopes of those functions in Figs. 4b and 5b, we conclude that the proposed "mixed" approximation improves computational efficiency as compared to the standard topology optimization framework. More specifically, for a given discretization with a fixed number of displacement DOFs (thus, with a roughly fixed computational cost), the proposed "mixed" approximation yields an enriched design field and, thus, produces topologies with potentially improved resolutions.

3.1 VEM-based topology optimization framework

This subsection formalizes the proposed topology optimization framework. In this framework, both the discrete displacement and design fields are constructed using the VEM. As we can see, because of its flexibility in handling any element geometry as well as local spaces, the VEM provides an efficient platform to formulate the proposed topology optimization framework.

To introduce the new topology optimization framework, we define the space $\mathcal{A}_h$ for admissible density function ρ_h as:

$$\mathcal{A}_h = \{\rho_h = \mathcal{F}_h(\eta_h) : 0 \le \rho_h \le 1, \quad \rho_h|_E \in \mathcal{V}(E), \\ \text{and } 0 \le \eta_h \le 1, \quad \eta_h|_E \in \mathcal{V}(E), \ \forall E \in \Omega_h\}. \quad (53)$$

In the above definition, we assume that both density function ρ_h and design function η_h are continuous functions with their DOFs being their values at vertices as well as mid-edge nodes of the mesh. The density function ρ_h is defined as the map of a design function η_h by the density filter $\mathcal{F}_h(\cdot)$.

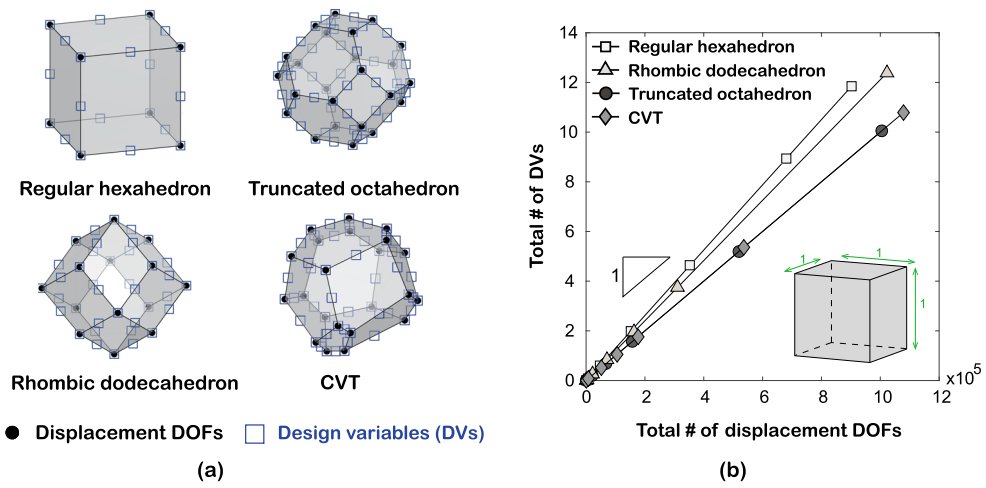


Fig. 5 **a** Illustrations of new "mixed" elements proposed in this work. The displacement DOFs are located at the vertices and the DVs are assigned to the vertices as well as mid-edge nodes of each element. **b** The total number of DVs as function of the total number of displacement DOFs for various discretizations of a unit cube

For a discretization Ω_h consisting of M elements and N_D DVs, we henceforth introduce two vectors, $\mathbf{z} = [z_1, z_2, ..., z_{N_D}]^T$ and $\mathbf{y} = [y_1, y_2, ..., y_{N_D}]^T$, where z_i and y_i are values of η_h and ρ_h at the ith DOF of η_h and ρ_h, respectively.

Since both functions ρ_h and η_h can be characterized by their DOFs, we adopt the following approach to construct the density filter map $\mathcal{F}_h(\cdot)$ between ρ_h and η_h, which is based on their DOFs and the associated position vectors. If we denote by $S(i)$ the set of the indices of DOFs whose positions fall within a sphere of prescribed radius R centered at $\mathbf{x}_i$ (the position vector associated with DOF i), y_i (the ith DOF of the density field ρ_h) is computed as:

$$y_i = \frac{\sum_{j\in S(i)} z_j (1 - ||\mathbf{x}_i - \mathbf{x}_j||/R)^q}{\sum_{k\in S(i)} (1 - ||\mathbf{x}_i - \mathbf{x}_k||/R)^q}, \tag{54}$$

where q is the order of the density filter (Bourdin 2001; Zegard and Paulino 2016). Figure 6 illustrates this density filter on a 2D mesh patch. For easier implementation, we express the density filter mapping between ρ_h and η_h in matrix form as:

$$\mathbf{y} = \mathbf{P}^{\mathcal{F}} \mathbf{z}, \tag{55}$$

where $\mathbf{P}^{\mathcal{F}}$ is a constant and sparse matrix with its (i, j)th component given by

$$\mathbf{P}^{\mathcal{F}}_{(ij)} = \frac{\max(0, (1 - ||\mathbf{x}_i - \mathbf{x}_j||/R)^q)}{\sum_{k\in S(i)} (1 - ||\mathbf{x}_i - \mathbf{x}_k||/R)^q}. \tag{56}$$

Moreover, within each element E (assuming E has m vertices), the density function ρ_h belongs to the VEM space $\mathcal{V}(E)$ defined in (11). Likewise, if we use y_i^E to denote the ith DOFs of ρ_h in E, we can express $\rho_h|_E$ in terms of the set of basis functions for $\mathcal{V}(E)$, $\varphi_1, ..., \varphi_m$, as:

$$\rho_h|_E = \sum_{i=1}^{m} \varphi_i y_i^E. \tag{57}$$

The volume constraint function in (52) can then be recast as:

$$\frac{\int_{\Omega_h} \rho_h}{|\Omega_h|} - \overline{V} = \frac{\sum_{E\in\Omega_h} \sum_{i=1}^{m} (\int_E \varphi_i) y_i^E}{\sum_{E\in\Omega_h} |E|} - \overline{V}, \tag{58}$$

where we recall from the (26) that $\int_E \varphi_i$, $i = 1, ..., m$ can be exactly computed by definition of $\mathcal{V}(E)$. To assist with implementation, we also express the volume constraint function in matrix notation as:

$$\frac{\int_{\Omega_h} \rho_h}{|\Omega_h|} - \overline{V} = \frac{\mathbf{V}^T \mathbf{P}^{\mathcal{V}} \mathbf{y}}{\mathbf{V}^T \mathbf{1}} - \overline{V} = \frac{\mathbf{V}^T \mathbf{P}^{\mathcal{V}} \mathbf{P}^{\mathcal{F}} \mathbf{z}}{\mathbf{V}^T \mathbf{1}} - \overline{V}, \tag{59}$$

where $\mathbf{V} = [|E_1|, |E_2|, ..., |E_M|]^T$ is a vector collecting element volumes and $\mathbf{P}^{\mathcal{V}}$ is a constant matrix with its (i, j)th component being:

$$\begin{aligned} \mathbf{P}^{\mathcal{V}}_{(ij)} &= \begin{cases} \frac{1}{|E_i|} \int_{E_i} \varphi_j & \text{if node } j \in \text{ element } E_i \\ 0 & \text{otherwise} \end{cases} \\ &= \begin{cases} \frac{1}{12} \sum_{f\in\mathcal{F}_j\subset E_i} (\mathbf{x}_{\mathcal{G}_f(j)+1} - \mathbf{x}_{\mathcal{G}_f(j)-1}) \wedge (\mathbf{x}_j - \mathbf{x}_c^E) \cdot (\mathbf{x}_j - \mathbf{x}_c^f) & \text{if } \mathbf{x}_j \in E_i \\ 0 & \text{otherwise} \end{cases}. \end{aligned} \tag{60}$$

We also note from the above definition that matrix $\mathbf{P}^{\mathcal{V}}$ is sparse, and thus it is formed and stored as a sparse matrix in our implementation.

In the state equation, the interpolated stiffness function $m_S(\rho_h)$ is utilized, which is assumed to be an element-wise constant functions, such that within element E, $m_S(\rho_h)|_E \doteq m_S(< \rho_h >_E)$, where $< \rho_h >_E$ denotes the volume average of ρ_h over E:

$$< \rho_h >_E = \frac{1}{|E|} \sum_{i=1}^{m} \left(\int_E \varphi_i \right) y_i^E. \tag{61}$$

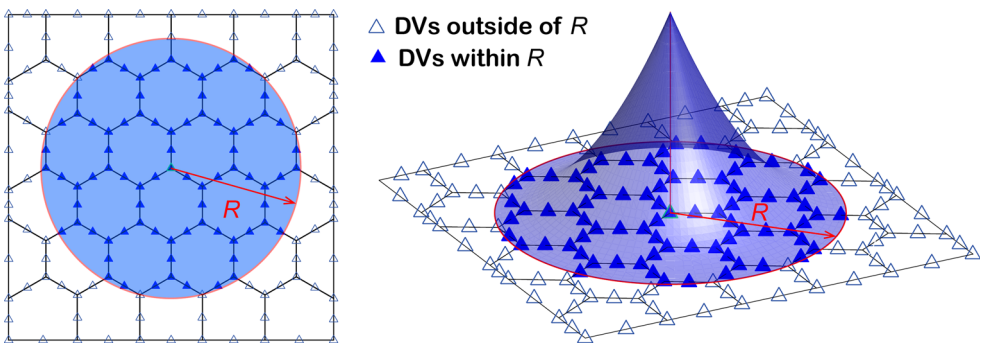


Fig. 6 An illustration of a quadratic (i.e. $q = 2$) density filter mapping for a 2D mesh patch. The DVs with solid markers are within the filter radius R and thus have non-zero weights

Having defined the form of the stiffness interpolation function, we incorporate it in the element-level discrete bilinear form (34) as follows:

$$a_h^{\rho,E}(\mathbf{u}_h,\mathbf{v}_h)=\int_E \boldsymbol{\varepsilon}(\boldsymbol{\Pi}_E^{\nabla}\mathbf{u}_h):\left\{[m_S(<\rho_h>_E)\mathbf{C}]\boldsymbol{\varepsilon}(\boldsymbol{\Pi}_E^{\nabla}\mathbf{v}_h)\right\} +m_S(<\rho_h>_E)\alpha^E S^E(\mathbf{u}_h-\boldsymbol{\Pi}_E^{\nabla}\mathbf{u}_h,\mathbf{v}_h-\boldsymbol{\Pi}_E^{\nabla}\mathbf{v}_h). \quad (62)$$

Notice that the first term (the consistency term) on the right-hand size of the above equation is now a constant function of the form:

$$a_h^{\rho,E}(\mathbf{u}_h,\mathbf{v}_h)=|E|m_S(<\rho_h>_E)\boldsymbol{\varepsilon}(\boldsymbol{\Pi}_E^{\nabla}\mathbf{u}_h):\left[\mathbf{C}\boldsymbol{\varepsilon}(\boldsymbol{\Pi}_E^{\nabla}\mathbf{v}_h)\right] +m_S(<\rho_h>_E)\alpha^E S^E(\mathbf{u}_h-\boldsymbol{\Pi}_E^{\nabla}\mathbf{u}_h,\mathbf{v}_h-\boldsymbol{\Pi}_E^{\nabla}\mathbf{v}_h). \quad (63)$$

Furthermore, we can pull $m_S(<\rho_h>_E)$ out of the above element-level bilinear form and get the global discrete bilinear form $a_h^{\rho}(\mathbf{u}_h,\mathbf{v}_h)$ as:

$$a_h^{\rho}(\mathbf{u}_h,\mathbf{v}_h)=\sum_{i=1}^{M} m_S(<\rho_h>_{E_i})a_h^{\rho,E_i}(\mathbf{u}_h,\mathbf{v}_h). \quad (64)$$

Similarly, we introduce a vector $\mathbf{s}$ in the implementation whose ith component s_i is the value of $m_S(<\rho_h>_E)$ for element E_i. Utilizing the matrix $\mathbf{P}^{\mathcal{V}}$, the vector $\mathbf{s}$ can be expressed as:

$$\mathbf{s}=m_S(\mathbf{P}^{\mathcal{V}}\mathbf{y})=m_S(\mathbf{P}^{\mathcal{V}}\mathbf{P}^{\mathcal{F}}\mathbf{z}). \quad (65)$$

We can then compute the global stiffness matrix for $a_h^{\rho}(m_S(\rho_h),\mathbf{u}_h,\mathbf{v}_h)$ as:

$$\mathbf{K}_{\boldsymbol{\rho}}=\sum_{i=1}^{M} s_i\mathbf{k}^{E_i}. \quad (66)$$

Finally, we arrive at the topology optimization formulation considering nodal densities as:

$$\begin{aligned}&\min_{\mathbf{z}\in[0,1]^{N_D}} \int_{\Gamma_h^{\mathbf{t}}}\mathbf{t}\cdot\mathbf{u}_h=\min_{\mathbf{z}\in[0,1]^{N_D}}\mathbf{F}^T\mathbf{U}\\ &\text{s.t.}\quad \frac{\mathbf{V}^T\mathbf{P}^{\mathcal{V}}\mathbf{P}^{\mathcal{F}}\mathbf{z}}{\mathbf{V}^T\mathbf{1}}-\overline{V}\le 0\\ &\text{with}\quad \mathbf{K}_{\boldsymbol{\rho}}\mathbf{U}=\mathbf{F}.\end{aligned} \quad (67)$$

We next describe how to compute the gradients of the objective and volume constraint functions with respect to the design variable $\mathbf{z}$. For the objective function, we first compute its gradient with respect to the vector $\mathbf{s}$ as:

$$\frac{\partial C}{\partial s_j}=-\mathbf{U}^T\frac{\partial\mathbf{K}_{\boldsymbol{\rho}}}{\partial s_j}\mathbf{U}=-\mathbf{U}^T\mathbf{k}^{E_j}\mathbf{U},\quad j=1,...,M, \quad (68)$$

and then, using the chain rule, we arrive at:

$$\frac{\partial C}{\partial\mathbf{z}}=(\mathbf{P}^{\mathcal{V}}\mathbf{P}^{\mathcal{F}})^T J_{m_S}(\mathbf{P}^{\mathcal{V}}\mathbf{P}^{\mathcal{F}}\mathbf{z})\frac{\partial C}{\partial\mathbf{s}}, \quad (69)$$

where $J_{m_S}(\mathbf{y})\doteq\text{diag}(m_S'(y_1),...,m_S'(y_{N_D}))$ is the Jacobian matrix of the stiffness interpolation function m_S. The

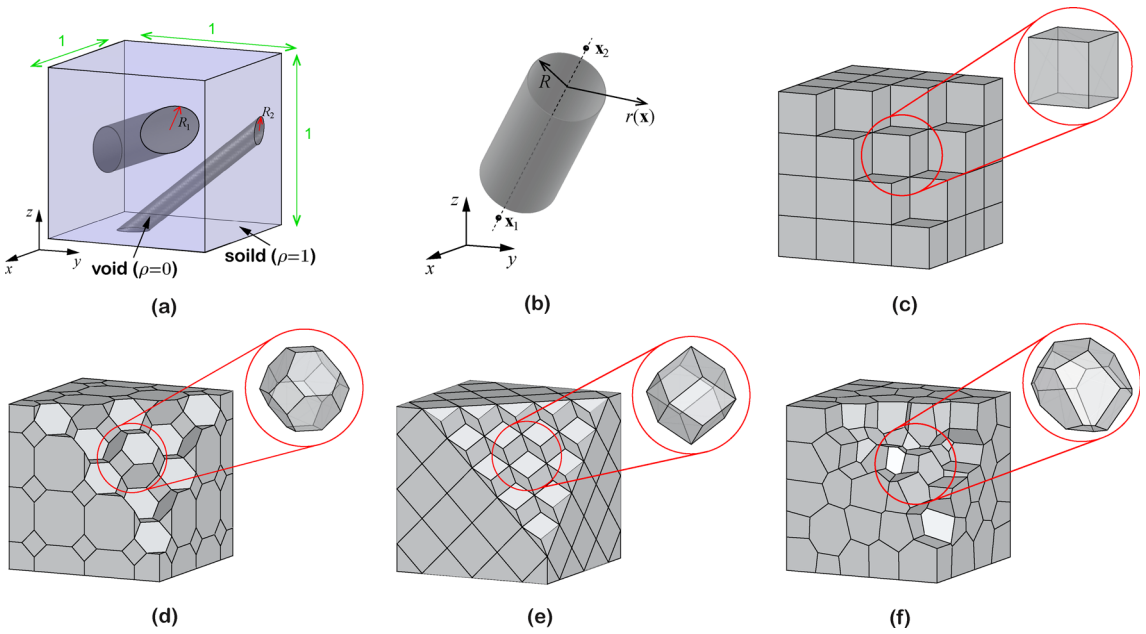


Fig. 7 **a** Problem description of a unit cube embedded with two hollow tubes. **b** Geometrical parametrization of a tube. **c** An example of the mesh consisting of regular hexahedra. **d** An example of the mesh consisting of truncated octahedra. **e** An example of the mesh consisting of Rhombic dodecahedra. **f** An example of the standard CVT mesh

gradient of the volume constraint function can be simply obtained as:

$$\frac{\partial g}{\partial \mathbf{z}} = \frac{(\mathbf{P}^{\mathcal{F}})^T (\mathbf{P}^{\mathcal{V}})^T \mathbf{V}}{\mathbf{V}^T \mathbf{1}}. \tag{70}$$

We conclude this subsection by noting that, even with an enriched space for design and material density functions, the proposed formulation (67), which is formulated in the VEM context using local projections, can be implemented in a similar code structure to the `PolyTop` software (Talischi et al. 2012b).

3.2 A verification example

In this subsection, we verify the proposed continuous density approximation through a simple numerical example and compare its performance with the standard element-wise density approximation. In particular, we want to compare how these two approximations perform in terms of capturing the varying density in a simple boundary value problem.

The setup of the boundary value problem is shown in Fig. 7a, where we consider a unit cube Ω with two hollow ($\rho = 0$) tubes. The remainder of the cube is solid ($\rho = 1$). The tube whose central axis passes through points $[0.22, 0.4, 0]^T$ and $[1, 0.7, 0.78]^T$ has a radius of $R_1 = 0.13$, and the other one whose central axis passes through $[0.5, 0, 0.3]^T$ and $[0.3, 0.6, 1]^T$ has a radius of $R_2 = 0.06$. For this setup, we introduce a continuous parametrization $\rho(\mathbf{x})$ of the material density over Ω using a smooth Heaviside function:

$$\rho(\mathbf{x}) = \frac{1}{1 + e^{-200(r_1(\mathbf{x}) - R_1)}} + \frac{1}{1 + e^{-200(r_2(\mathbf{x}) - R_2)}} + \varepsilon - 1, \tag{71}$$

where ε is a small positive number assigned to ensure the positivity of $\rho(\mathbf{x})$, and $r_1(\mathbf{x})$ and $r_2(\mathbf{x})$ are the distances of a given point $\mathbf{x}$ to the central axes of the first and second tubes, respectively. As illustrated in Fig. 7b, for a tube whose central axis passes through two given points $\mathbf{x}_1$ and $\mathbf{x}_2$, $r(\mathbf{x})$ is given by:

$$r(\mathbf{x}) = \frac{||(\mathbf{x} - \mathbf{x}_1) \wedge (\mathbf{x} - \mathbf{x}_2)||}{||\mathbf{x}_1 - \mathbf{x}_2||}, \tag{72}$$

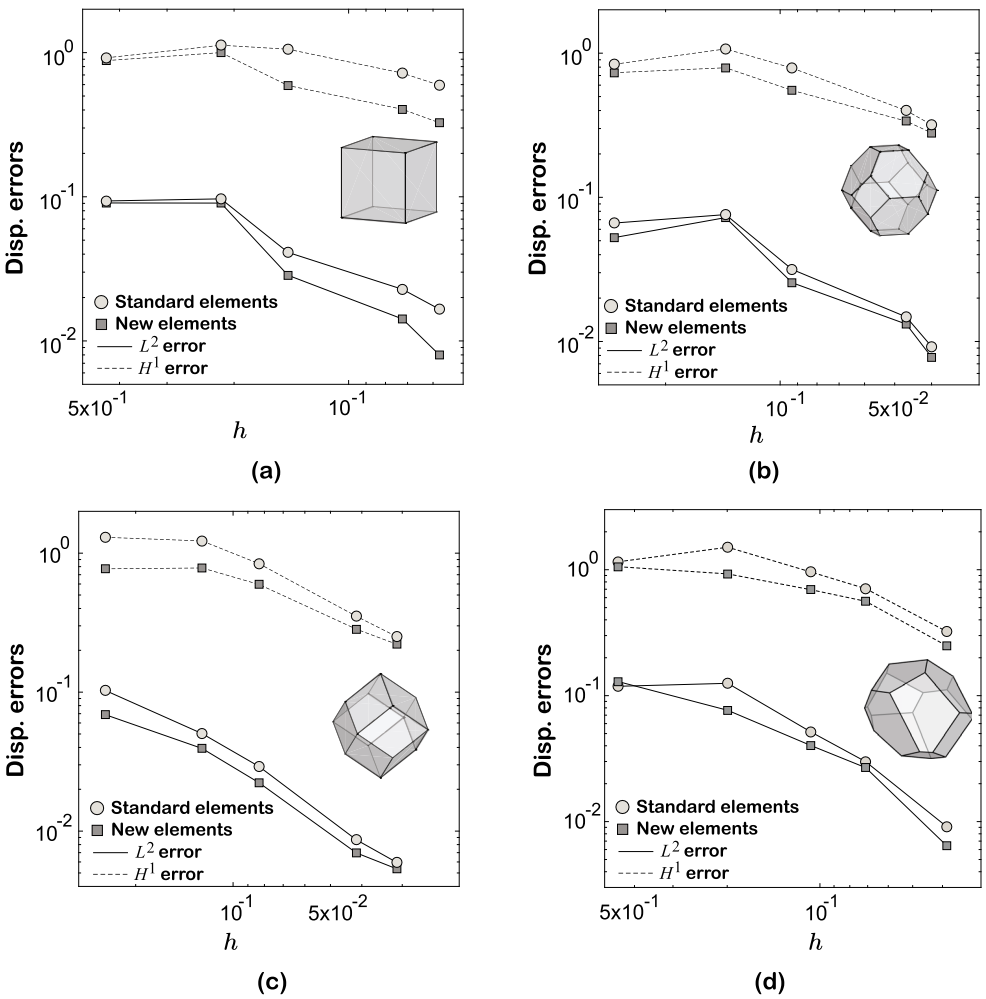


Fig. 8 Convergence of both $\mathcal{L}^2$ and $\mathcal{H}^1$ displacement errors as function of the average mesh size h when the standard and proposed "mixed" elements are used for: **a** regular hexahedral meshes; **b** truncated octahedral meshes; **c** rhombic dodecahedral meshes; and **d** standard CVT meshes

where $||\cdot||$ stands for the Euclidean norm. Making use of $\rho(\mathbf{x})$, we obtain the (continuous) distribution of the elasticity modulus in Ω, which is given by $\mathbf{C}(\mathbf{x}) = \rho(\mathbf{x})\mathbf{C}_0$, where $\mathbf{C}_0$ is the elasticity modulus tensor of the solid material. In the present study, the solid material is considered to be isotropic with Young's modulus $E_Y = 25$ and Poisson's ratio $\nu = 0.3$.

Adopting the method of manufactured solutions, we assume an exact displacement solution $\mathbf{u}$ of the form:

$$u_x = z^2 e^{xy} \quad u_y = 2y^2 z^3 + zx^4 \text{ and } u_z = z\sin(2\pi x)e^y, \tag{73}$$

which is prescribed on the entire boundary of the cube. Accordingly, a body force, which is computed using the exact displacement solution $\mathbf{u}$ and stiffness distribution $\mathbf{C}(\mathbf{x})$, is prescribed in the interior of Ω. Four families of polyhedral meshes are considered: hexahedral, truncated octahedral, rhombic dodecahedral, and standard CVT; and their examples are shown in Fig. 7c–f, respectively. To assess the accuracy of the solution, we make use of the $\mathcal{L}^2$ error of the displacement defined in (47), and the $\mathcal{H}^1$ displacement error, which is defined as:

$$\epsilon_{\mathbf{u},1} = \sqrt{\sum_{E\in\Omega_h}\int_E (\nabla(\mathbf{\Pi}_E^{\nabla}\mathbf{u}_h) - \nabla\mathbf{u})\cdot(\nabla(\mathbf{\Pi}_E^{\nabla}\mathbf{u}_h) - \nabla\mathbf{u})\mathrm{d}\mathbf{x}}, \tag{74}$$

where the integral is evaluated using the same fourth-order integration rule on each tetrahedral subdivision of E.

In the numerical simulations, the material distribution $\rho(\mathbf{x})$ needs to be approximated in order to compute the stiffness matrix. We consider two approaches here. The first approach assigns a constant density to each element with the value of $\rho(\mathbf{x})$ evaluated at the centroid of that element. This approach resembles the situation in the standard density-based topology optimization where element-wise constant densities are used. On the other hand, the second approach evaluates $\rho(\mathbf{x})$ at the vertices and mid-edge nodes, and interpolates them using VEM basis functions. This approach resembles the situation in the proposed topology optimization framework where the DVs are located at the vertices as well as mid-edge nodes of the mesh. Once the material distribution is approximated, the stiffness matrix of the discretized system for both approaches can be constructed. For the former approach, the stiffness matrix is constructed using the standard procedure, whereas, in the latter approach, the stiffness matrix is formed following (64).

Figure 8 a–d show the convergence of both $\mathcal{L}^2$ and $\mathcal{H}^1$ displacement errors as functions of mesh size h for both approaches on the four families of meshes. The standard elements represent the first approach and the new elements represent the second approach. As we can see from the comparison, the second approach (corresponding to the

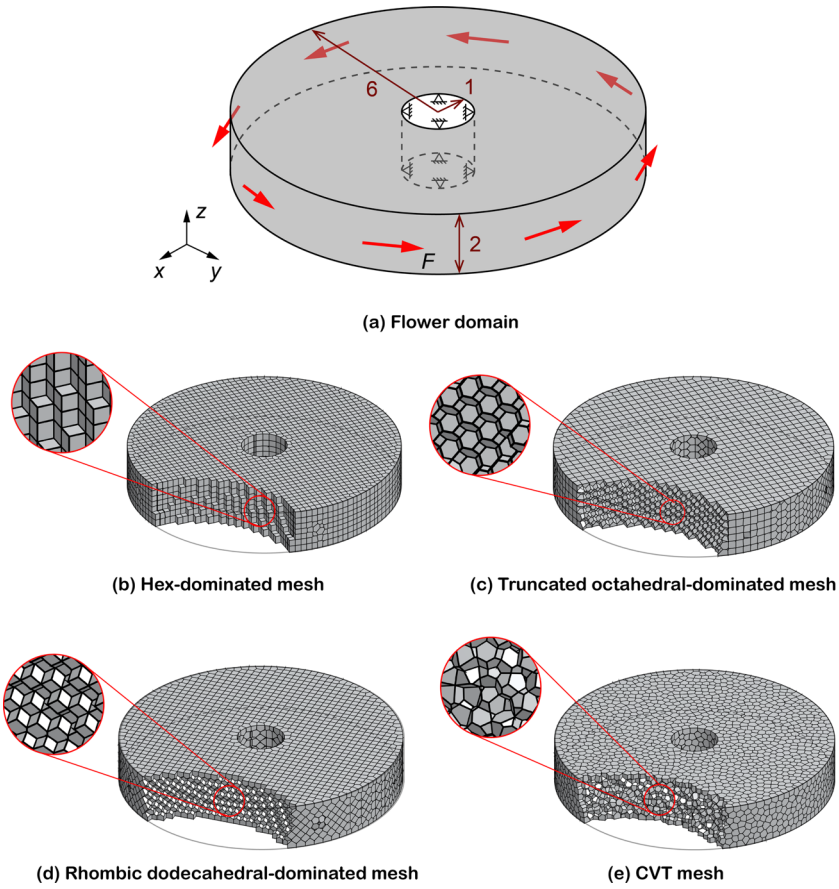


Fig. 9 **a** The geometry, load and boundary conditions of the shear loaded disc problem. **b** Hexahedral-dominated mesh consisting of 31,791 nodes and 12,180 elements. **c** Truncated octahedral-dominated mesh consisting of 65,418 nodes and 10,808 elements. **d** Rhombic dodecaheral-dominated mesh consisting of 52,606 nodes and 12,264 element. **e** Standard CVT mesh consisting of 64,097 nodes and 10,000 elements

Table 1 Statistics of the meshes for the shear loaded disc problem

Mesh	# Nd	# El.	# DV (proposed)	# DV (standard)
Hexahedral	31,791	12,180	104,166	12,180
CVT	64,097	10,000	191,256	10,000
Trun. octahedral	65,418	10,808	196,329	10,808
Rhom. dodecahedral	52,606	12,264	171,275	12,264

proposed topology optimization framework) always gives more accurate displacement solutions than the first approach (corresponding to the standard density-based framework) on a given mesh. This in turn implies that the material densities interpolated from values on both vertices and mid-edge nodes provide better approximations of the exact density distribution $\rho(\mathbf{x})$ than the one constructed from element-wise constant values. Based on such observation, we argue that, in this topology optimization setting, the proposed material density approximation is favored because it gives a better parametrization of the density distribution of the varying topologies. We also remark that, although Fig. 8 a–d suggest that the regular hexaheral and CVT meshes produce more accurate displacement solutions than the other two families of meshes, this conclusion could be problem-dependent and might not hold for other density distributions different from the one defined by (71).

4 Numerical examples

We present several examples involving non-Cartesian design domains to demonstrate the effectiveness and versatility of the proposed VEM-based topology optimization framework. For all the design examples, we select the Ersatz parameter as $\varepsilon = 10^{-9}$ and take Young's modulus and Poisson's ratio of the solid phase to be $E_Y = 100$ and $\nu = 0.3$, respectively. During optimization, the optimality criterion (OC) (Christensen and Klarbring 2009) is chosen as the design-variable update scheme with the damping parameter and move limit being $\eta = 0.5$ and $move = 0.3$. The maximum tolerance for the change of design variables in the convergence criterion is taken to be 0.1%. Moreover, a continuation scheme of the penalization parameter p is adopted here. We start with $p = 1$ and increase it every (maximum) 20 optimization iterations by 1 until $p = 3$. When p reaches 3, we then set the maximum allowable optimization iteration number to be 150. For comparison purposes, the topologies obtained by the standard topology optimization framework (Gain et al. 2015) with element-wise constant densities are also provided. The same VEM formulation is used to solve the state equation in the standard framework as in the proposed framework. Unless otherwise stated, the final topologies are plotted using iso-surface with the cutoff value being 0.5. To distinguish the topologies obtained using the standard framework and the proposed one, we plot the results obtained using the standard framework in blue and those obtained using the proposed framework in red. We also point out that, for a fixed mesh and fixed set of parameters, the computational time of both proposed and standard frameworks are closely identical because the discretized size of state equations are the same.

4.1 Shear loaded disk problem

The first design example is the shear loaded disc problem. As shown in Fig. 9, the design domain is a disc with an outer radius of 6 and an inner radius of 1. The thickness of the disk is taken to be 2. Fixed on the inner surface, the

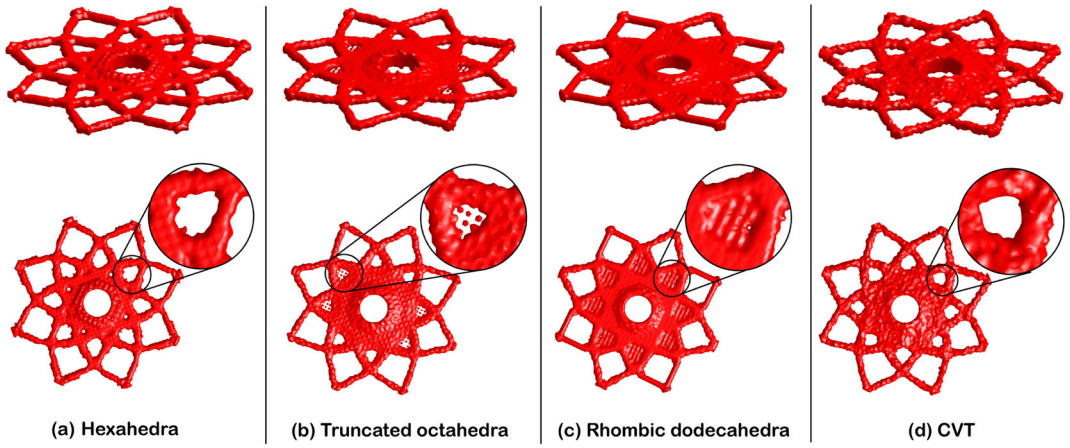


Fig. 10 The final topologies obtained from the proposed topology optimization without the density filter on **a** the hexahedral-dominated mesh; **b** the truncated octahedral-dominated mesh; **c** the rhombic dodecahedral-dominated mesh; and **d** the CVT mesh

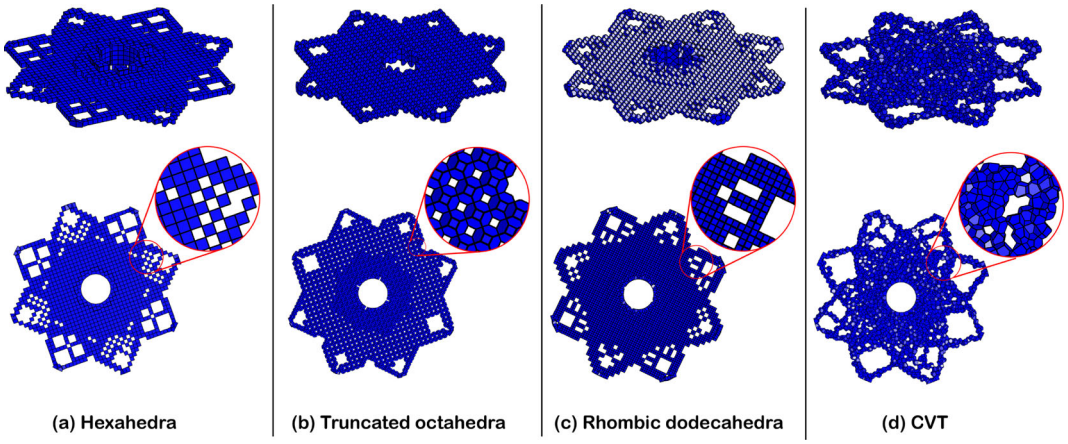


Fig. 11 The final topologies obtained from the standard topology optimization without the density filter on **a** the hexahedral-dominated mesh; **b** the truncated octahedral-dominated mesh; **c** the rhombic dodecahedral-dominated mesh; and **d** the CVT mesh. In the designs, those elements whose densities are above 0.01 are plotted

circumference of this disk is subjected to eight equidistant shear loads of uniform magnitude 1. A volume fraction of 10% is prescribed.

In order to provide a thorough assessment of the performance of the proposed framework on various types of meshes, we consider four meshes in this design example: a hexahedral-dominated mesh, a truncated octahedral-dominated mesh, a rhombic dodecaheral-dominated mesh, and a standard CVT mesh, as shown in Fig. 9b–e, respectively. The first three meshes consist of regular space-filling polyhedra in the interior of the design domain and unstructured polyhedra in the near-boundary regions, while the standard CVT mesh is made up of unstructured polyhedra inside the entire design domain. The mesh statistics are provided in Table 1. Notice that the four meshes have similar numbers of elements.

We first design the shear loaded disk problem without applying the density filter. Figures 10 and 11 show the final topologies obtained from the proposed and the standard frameworks, respectively, on the four meshes. Notice that, for the results obtained with the standard topology optimization framework, we get almost black and white designs for all the four meshes. Thus, for those results, instead of showing the iso-surface plots of the final topology, we simply plot those elements whose densities are above 0.01. In the topology optimization literature, it is well known that, without density or sensitive filters, quadrilateral or hexahedral meshes will produce checkerboard patterns in the standard framework (e.g., see Fig. 11a). For general discretizations, a previous work has demonstrated that 2D regular hexagonal (Talischi et al. 2009) and 2D and 3D CVT meshes (Talischi et al. 2010, 2012a; Gain et al. 2015) are free of checkerboard patterns. From our numerical results in Fig. 11c for truncated octahedral and Fig. 11d for rhombic

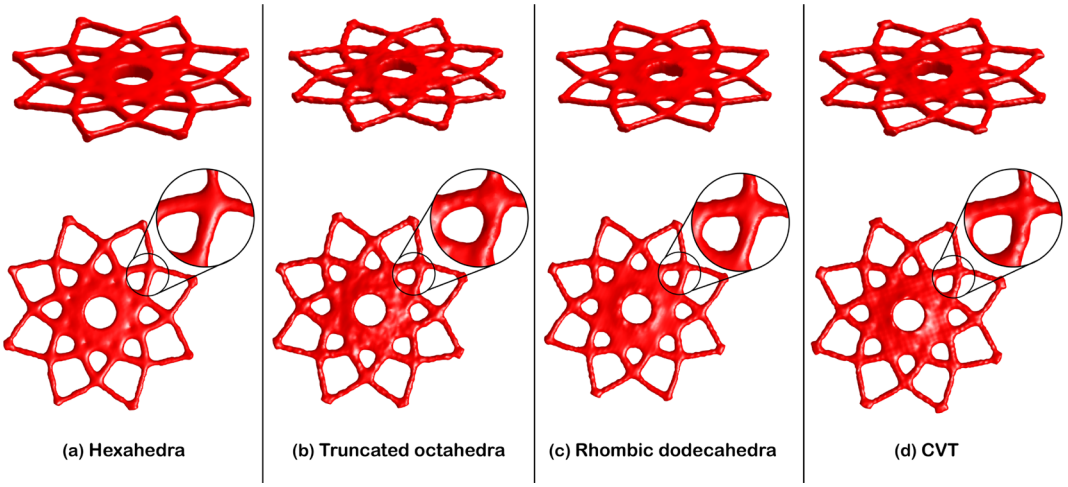


Fig. 12 The final topologies obtained from the proposed topology optimization with a quadratic density filter of radius $R = 0.45$ on **a** the hexahedral-dominated mesh; **b** the truncated octahedral-dominated mesh; **c** the rhombic dodecahedral-dominated mesh; and **d** the CVT mesh

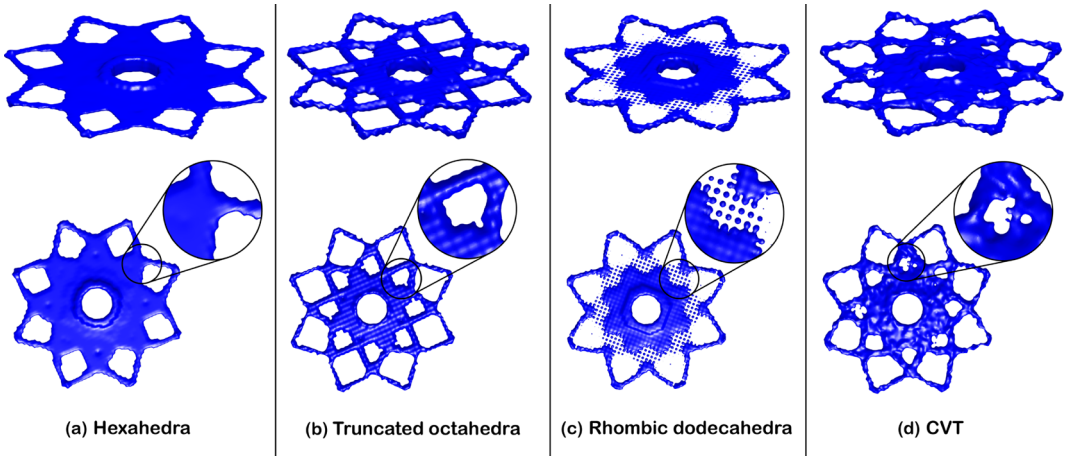


Fig. 13 The final topologies obtained from the standard topology optimization with a quadratic density filter of radius $R = 0.45$ on **a** the hexahedral-dominated mesh; **b** the truncated octahedral-dominated mesh; **c** the rhombic dodecahedral-dominated mesh; and **d** the CVT mesh

dodecahedra, we observe that, unlike the regular polygonal discretization in 2D, regular polyhedral discretization in 3D tends to generate patterns indicating numerical instability. Those patterns provide artificial stiffness and are impractical from a manufacturing perspective. On the contrary, with the proposed topology optimization

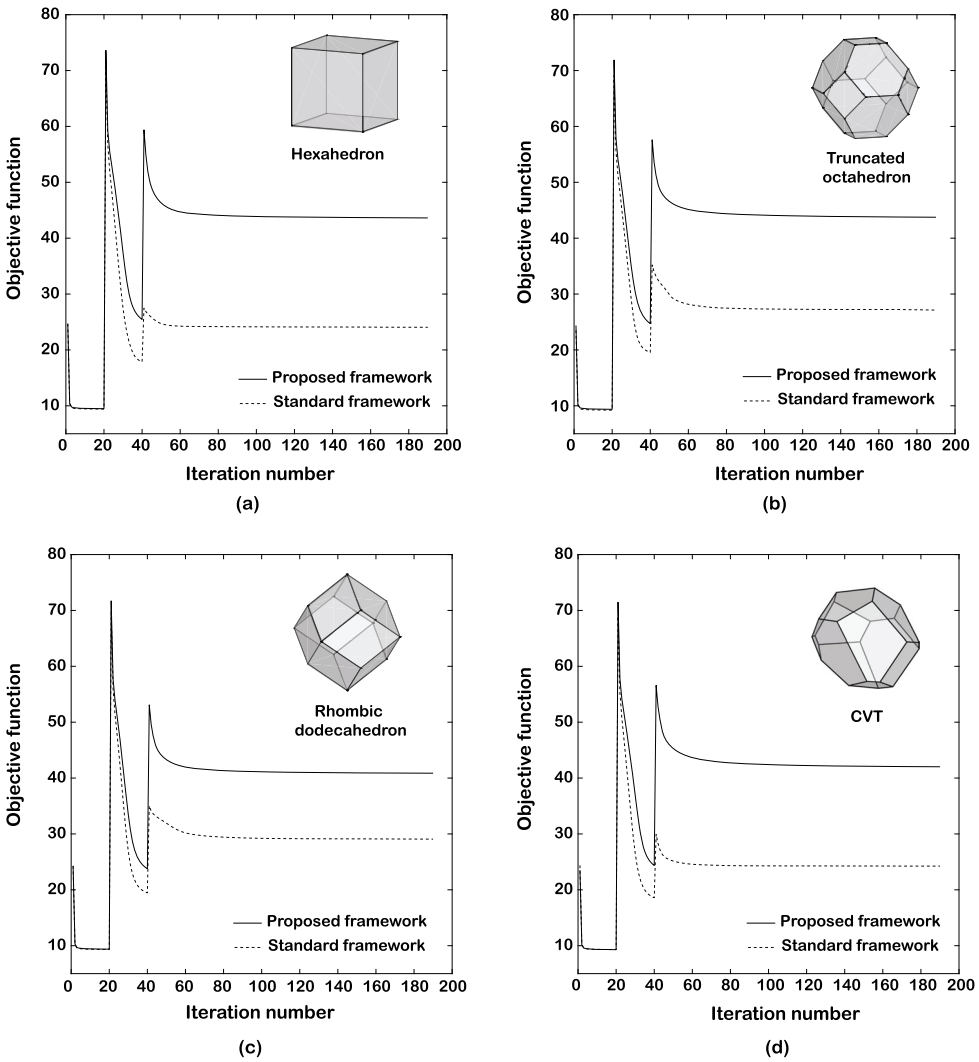


Fig. 14 The convergence history of objective functions for both proposed and standard frameworks on: **a** hexahedral-dominated mesh; **b** truncated octahedral-dominated mesh; **c** rhombic dodecahedral-dominated mesh; and **d** CVT mesh. Both frameworks deliver almost identical convergence rates. We also emphasize that it is unfair to compare the absolute values of objective functions between the two frameworks because different density approximations are adopted

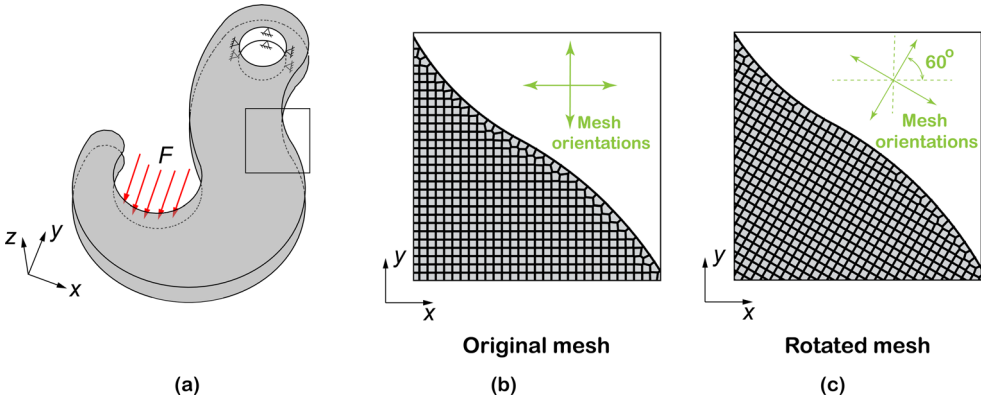


Fig. 15 **a** The geometry, load and boundary conditions of the hook problem. **b** Hexahedral-dominated mesh consisting of 134,634 nodes and 63,350 elements. **c** Hexahedral-dominated mesh generated by rotating the seeds of the mesh counterclockwise 60° in the $x - y$ plane. The resulting mesh contains 132,457 nodes and 63,320 elements

framework, we obtain physical designs on all the four meshes considered, which resemble a flower, although their iso-surface plots exhibit rough boundaries because of the absence of regularization from the density filter.

We then apply a quadratic density filter (i.e., $q = 2$) with a radius of $R = 0.45$, and keep the other parameter settings unchanged. The results obtained are shown in Figs. 12 and 13 using the proposed and standard frameworks, respectively. A few observations can be made. First, compared to the designs obtained from the standard framework, the ones obtained from the proposed framework display iso-surface representation with greatly improved smoothness. This is because, with the proposed framework, we have more DVs as compared to the standard framework on the same discretization, as shown in Table 1. Second, unlike the topologies obtained using the standard framework, the ones obtained using the proposed framework exhibit similar results on all meshes, suggesting that the proposed framework is less sensitive to mesh types than the standard framework. Moreover, Fig. 14 a–d depict the convergence history of the objective functions for both the proposed and standard frameworks on hexahedral-dominated, truncated octahedral-dominated, rhombic dodecahedral-dominated, and CVT meshes, respectively. The comparison in those figures suggests that the proposed and standard frameworks deliver almost identical convergence rate on various types of polyhedral meshes. Because, for any given mesh, the size of the stiffness matrix is identical for both frameworks, the major computational costs for the standard and proposed frameworks are identical as well. However, as shown in Table 1, the proposed framework can handle a significantly larger number of design variables than the standard one on all the meshes considered. Finally, we emphasize that it is unfair to compare the absolute values of the objective function between the standard and proposed frameworks. Even on the same mesh, different approximations of the density field (especially for cases where the SIMP penalization factor $p > 1$) will surely yield different displacement solutions and, eventually, lead to different absolute values of the objective function.

4.2 Hook design

Having investigated the performance of the proposed topology optimization framework through the last design example, we now apply it to several problems involving complex design domains that are non-Cartesian. Let us first look at the hook problem. As shown in Fig. 15a, the design domain in this problem is fixed on the upper half of the circle and is subjected to a uniformly distributed line load in the negative z direction. The volume fraction of this problem is set as $\overline{V} = 15\%$ and the radius of the quadratic density filter is $R = 2.5$. Through this design problem, we aim to quantify the mesh bias in both the standard and proposed topology optimization frameworks, and to demonstrate that the proposed one is less biased to the initial mesh than the standard one.

To that end, we consider two similar polyhedral meshes of the design domain. The first mesh, shown in Fig. 15b, is generated using the Voronoi seeds with Cartesian alignment; and the second mesh, shown in Fig. 15c, is generated using the seed alignment obtained by rotating the seeds of the first mesh counterclockwise 60° in the $x - y$ plane. For both meshes, the seeds in regions near the boundary are then updated using Lloyd's algorithm (Thedin et al. 2014) in order to capture the boundary geometry of the design domain. As a result, both meshes are made up of regular hexahedra in the interior and unstructured polyhedra in regions near the boundary, and contain similar number of nodes and elements, as summarized in Table 2. However,

Table 2 Statistics of the two hexahedral meshes for the hook problem

Mesh	# Nd	# El	# DV (proposed)	# DV (standar d)
Original	134,634	63,350	455,586	63,350
60° rotated	132,457	63,320	449,375	63,320

the two meshes possess different preferred orientations, as shown in Fig. 15b and c.

In Fig. 16a and b, we show the converged designs from both the proposed and standard topology optimization frameworks for the original and rotated meshes, respectively. Our main conclusion from comparing Fig. 16a and b is that the designs obtained from the proposed framework are less biased to the initial meshes than the standard framework. To visualize the bias, let us compare the fan regions of the designs obtain from the two frameworks. For the designs obtained by the proposed framework on both original and rotated meshes, the fan regions resemble the one obtained from a 2D analogue of this problem (Talischi et al. 2012b). The orientation of the members in the fan regions seem to be not influenced by the preferred orientations of the initial meshes. In contrast, for the designs obtained by the standard framework, it is clear that the orientation of the members in the fan regions are biased toward the preferred orientations of the initial meshes. For example, the original mesh has preferred orientation along the x and y axes. As shown in blue design in Fig. 16a, several members in the fan region of its design are clearly biased toward these two orientations. In addition, comparing the final designs obtained by the standard and proposed frameworks, we also notice that the ones obtained from the proposed framework possess smoother iso-surface representations than the ones obtained from the standard framework.

4.3 Wrench design

We investigate the design of a wrench problem. The design domain of this problem is depicted in Fig. 17a. In the domain, the bigger circle is fixed and half of the other circle is subjected to a distributed line load along the negative y direction. Here, we consider two CVT meshes, a coarse one and a refined one, whose statistics are given in Table 3. Again, a quadratic density filter is applied with a radius of $R = 0.05$ and the volume fraction is prescribed as $\overline{V} = 15\%$. In this design example, we prescribe another requirement that the final topology has to be symmetric. To achieve this, we use a matrix $\mathbf{P}^{\mathcal{S}}$ introduced in Talischi et al. (2012b), which is given by:

$$\mathbf{P}^{\mathcal{S}}_{(\ell k)} = \begin{cases} 1 & \text{if 1) the } y \text{ coordinate of the } k\text{th DOF is non-negative; and 2) } k = \ell \text{ or the } k\text{th and } \ell\text{th DOFs are } y\text{ -symmetric} \\ 0 & \text{otherwise} \end{cases} \tag{75}$$

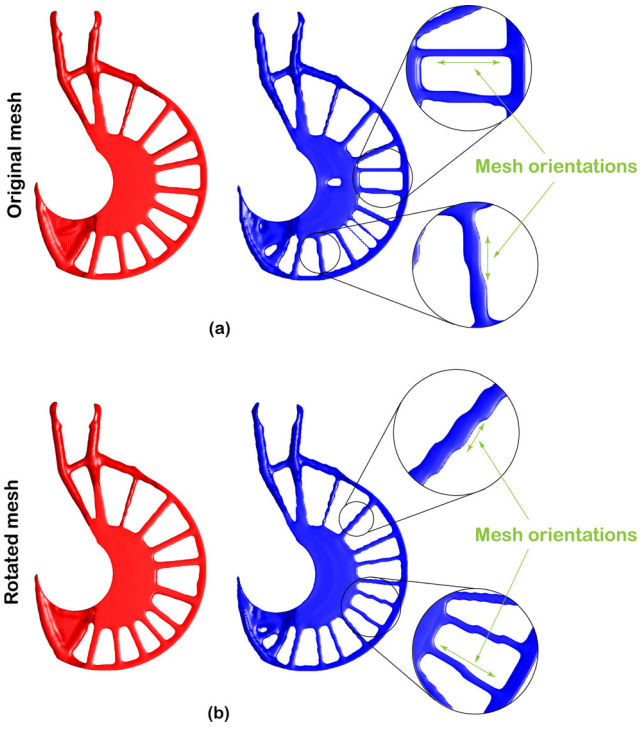


Fig. 16 **a** The final topologies for the hook problem obtained from both the proposed and standard frameworks on the original mesh (c.f. Fig. 15**b**). **b** The final topologies for the hook problem obtained from both the proposed and standard frameworks on the rotated mesh (c.f. Fig. 15**c**)

and the vectors $\mathbf{y}$ and $\mathbf{s}$ are then given by:

$$\mathbf{y} = \mathbf{P}^{\mathcal{S}}\mathbf{P}^{\mathcal{F}}\mathbf{z} \quad \text{and} \quad \mathbf{s} = m_S(\mathbf{P}^{\mathcal{V}}\mathbf{P}^{\mathcal{S}}\mathbf{P}^{\mathcal{F}}\mathbf{z}) \tag{76}$$

respectively. As we can see, the matrix $\mathbf{P}^{\mathcal{S}}$ enforces symmetry through mapping the admissible topologies in the design space to symmetric configurations.

Figure 18 a and b show the final topologies obtained using the proposed and standard frameworks, respectively, for the coarse CVT mesh; and Fig. 19 a and b show the ones for the refined CVT mesh. As an immediate observation, the proposed and standard frameworks yield symmetric[1] designs similar to each other for both the coarse and refined meshes. With the coarse mesh, the proposed framework yields a clearer (and manufacturable) design, which resembles the 2D optimization result in Talischi et al. (2012b), than the standard framework, indicating that the

[1] Although the CVT meshes are generated from reflected seeds to ensure their symmetry, certain regions on the mesh boundaries, especially near the two circles, are not fully symmetric due to the limitation of the meshing software used by Thedin et al. (2014) in representing curved boundaries. Hence, results obtained from the proposed framework (i.e. Figs 18 and 19a) are slightly asymmetric in those regions. However, we note that this minor issue will not affect the overall quality of the designs (and can be resolved when an improved version of meshing software is used).

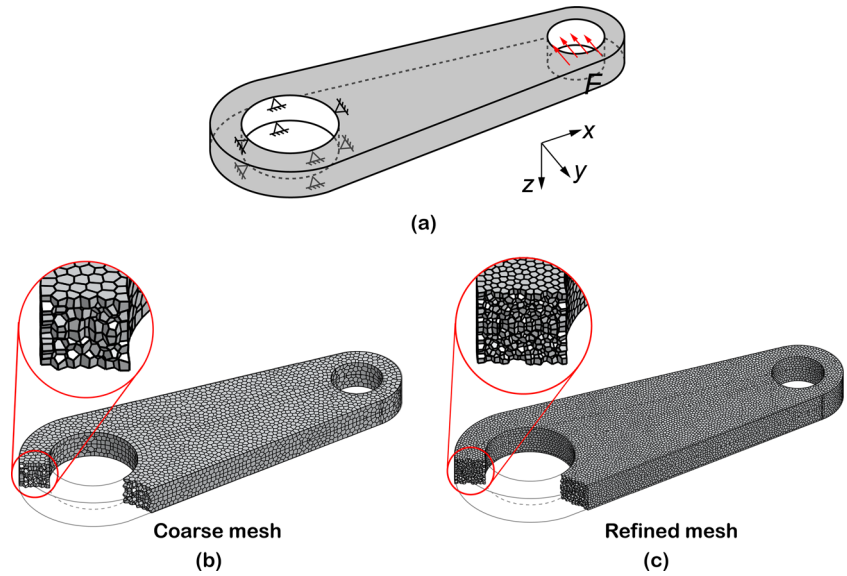


Fig. 17 **a** The geometry, load and boundary conditions of the wrench problem. **b** A relatively coarse CVT mesh consisting of 68,339 nodes and 12,000 elements. **b** A relatively refined CVT mesh consisting of 349,748 nodes and 60,000 elements. Both meshes are symmetric with respect to the x axis and obtained by reflecting the Voronoi seeds along the x axis

Table 3 Statistics of the mesh for the wrench problem

Mesh	# Nd	# El	# DV (standard)	# DV (proposed)
CVT (coarse)	68,339	12,000	205,724	12,000
CVT (refined)	349,748	60,000	1,052,109	60,000

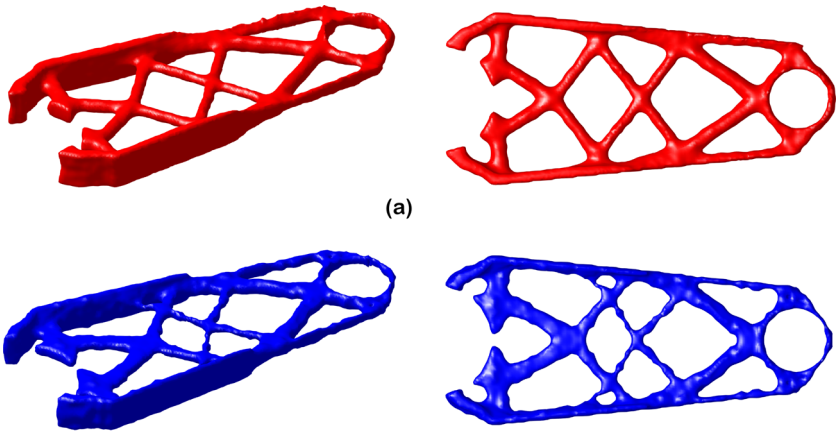


Fig. 18 The final topologies for the wrench problem obtained from the **a** proposed and **b** standard frameworks. Both topologies are obtained on the coarse CVT mesh with a quadratic density filter of radius $R = 0.05$ and prescribed volume fraction of $\overline{V} = 15\%$

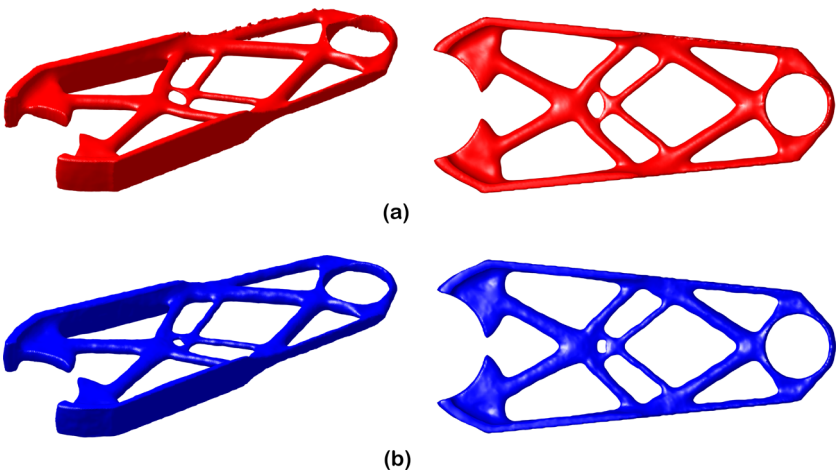


Fig. 19 The final topologies for the wrench problem obtained from the **a** proposed and **b** standard frameworks. Both topologies are obtained on the refined CVT mesh with a quadratic density filter of radius $R = 0.05$ and prescribed volume fraction of $\overline{V} = 15\%$

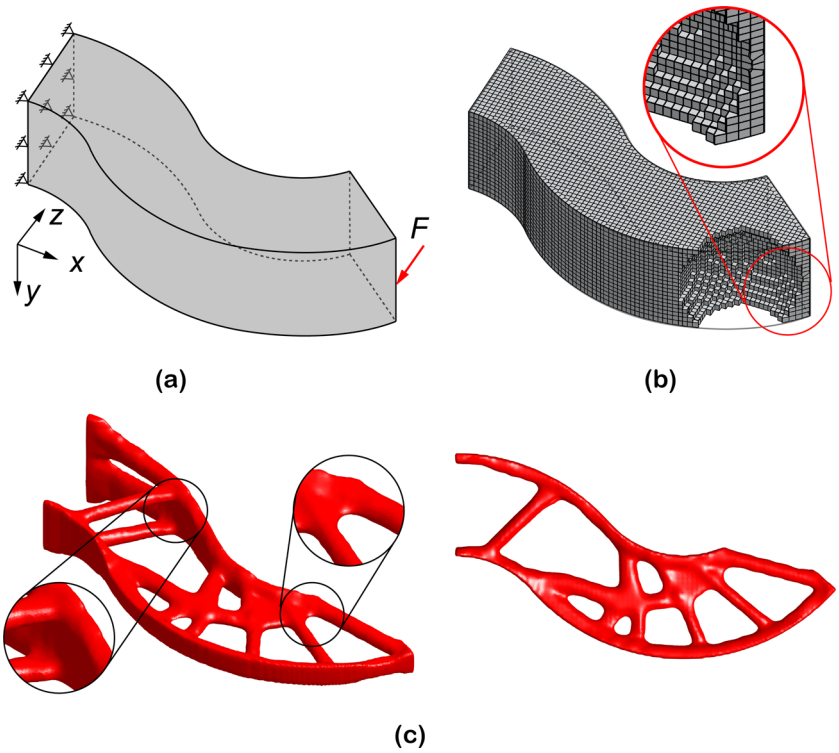


Fig. 20 **a** The geometry, load and boundary conditions of the serpentine problem. **b** A regular hexahedral-dominated mesh consisting of 75,624 nodes and 33,520 elements. **c** The final topologies for the serpentine problem obtained from the proposed framework with a quadratic density filter of radius $R = 0.4$ and prescribed volume fraction of $\overline{V} = 10\%$

proposed framework is more effective on relatively coarser meshes.

4.4 Serpentine design

In the final design example, we perform topology optimization in a serpentine domain. As shown in Fig. 20a, the serpentine domain is fixed on its left face and is subjected to a point load along the negative z direction in the middle of the lower edge of the right face. A volume fraction of $\overline{V} = 10\%$ is prescribed and a quadratic density filter of radius $R = 0.4$ is used. As in the hook example, we consider a mesh composed of regular hexahedra in the interior and unstructured polyhedra in regions near the boundary. The mesh is plotted in Fig. 20b and its statistics is presented in Table 4. In Figs. 20c and 21, we show both the final design produced by the proposed framework and the manufactured design using FDM (Fused Depositon Modeling) 3D printing, respectively. This example demonstrates that the proposed topology optimization framework can lead to designs that are directly manufacturable (Zegard and Paulino 2016).

Table 4 Statistics of the mesh for the serpentine problem

Mesh	# Nd	# El	# DV (standard)
Hexahedral	75,624	33,520	253,675

5 Concluding remarks

This work establishes a 3D VEM-based topology optimization framework on general polyhedral discretizations. The unique feature of this work is that it takes full advantage of the VEM and applies it to both structural and optimization problems. In terms of structural problems, the VEM is adopted to solve the state equation efficiently and effectively. Because VEM does not require explicit computation of the shape functions and their gradients, it does not need numerical integration and is less sensitive to degenerated polyhedra (e.g., ones with skinny faces or small edges) as compared to the FEM. In terms of optimization problems, exploiting the great flexibility of VEM in element geometries and local space definitions, we introduce an enhanced VEM space for the continuous design and material density

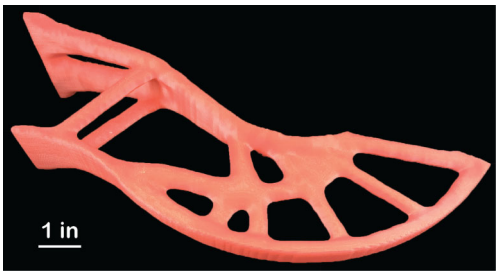


Fig. 21 Printed model of the serpentine design using the FDM 3D printing

functions, which contain DOFs at the vertices as well as the mid-edge nodes of the mesh.[2] The total volume of any topologies in this design space can be computed exactly using a properly defined VEM projection operator. As a result, for a given mesh and under a similar computational cost, the proposed VEM-based topology optimization is shown to produce designs with improved geometrical resolution as compared to the standard topology optimization framework with element-wise constant DVs and material densities. We also demonstrate that the VEM-based topology optimization framework can be implemented in a similar code structure to the `PolyTop` software (Talischi et al. 2012a). In terms of discretizing complex domains in 3D, this work explores two approaches: unstructured polyhedral (i.e., CVT) meshes and regular polyhedra- (i.e., hexahedra, truncated octahedra, and rhombic dodecahedra) dominated meshes.

Both VEM approximations for the displacement field and the material density function are verified through numerical examples. The convergence of the VEM in solving the state equations is verified via a benchmark problem involving beam bending. Through a simple example, we also compare the performance of the proposed continuous density parametrization with the commonly used element-wise constant one in terms of approximating varying material density distributions. Moreover, several design examples involving non-Cartesian domains are presented, showcasing that the proposed VEM-based topology optimization framework produces designs with improved quality and achieves higher computational efficiency. In terms of design quality, we demonstrate through the hook and wrench design examples that the proposed framework is less biased to the initial mesh (see the results in Fig. 16) and can produce designs with smoother iso-surface representations, especially for coarser meshes (e.g., see the comparison results in Fig. 18). In terms of computational efficiency, we demonstrate that the proposed framework delivers the same convergence rate as the standard one on various types of discretizations (see Fig. 14). For a given mesh, because the size of the stiffness matrix is identical for both frameworks, the proposed framework is able to consider a significantly bigger number of design variables than the standard one with almost identical computational effort, and thus achieves a higher efficiency. Additionally, we show that the proposed framework is flexible in imposing various pattern and layout constraints (i.e., symmetry constraints shown in Figs. 18 and 19), and can lead to designs that are directly manufacturable by 3D printing (see Fig. 21).

[2]Such mixed elements deserve further theoretical investigation to address their stability and the balance between the discrete spaces for the displacement field and the density field (see, for example, Jog and Haber 1996; Chi et al. 2016).

Finally, we remark that the proposed VEM-based topology optimization offers an effective tool for mesh adaptation in topology optimization. The potential of this research includes developing efficient mesh adaption strategies for topology optimization by exploiting the advantages of polyhedral elements.

Acknowledgement This paper is dedicated to the memory of Prof. Luiz Eloy Vaz (January 24, 1947-April 15, 2014).

Funding information HC and GHP received financial support from the US National Science Foundation (NSF) under project no. 1663244. We also received an endowment provided by the Raymond Allen Jones Chair at the Georgia Institute of Technology. AP received financial support from the Carlos Chagas Filho Research Foundation of Rio de Janeiro State (FAPERJ) under grant E-26/203.189/2016. AP and IFMM received support from Tecgraf/PUC-Rio (Group of Technology in Computer Graphics), Rio de Janeiro, Brazil, and from National Council for Scientific and Technological Development (CNPq).

Disclaimer The information presented in this publication is the sole opinion of the authors and does not necessarily reflect the views of the sponsors or sponsoring agencies.

Compliance with Ethical Standards

Conflict of interests The authors declare that they have no conflict of interest.

Appendix `PolyTop3D`: an efficient MATLAB implementation of the proposed VEM-based topology optimization framework

An implementation of the proposed VEM-based topology optimization framework into a modular MATLAB code named `PolyTop3D`, which can handle any non-Cartesian design domains (specified by the users) on general polyhedral discretizations (both structured and unstructured), is available in the Electronic Supplementary Material accompanying this publication. The `PolyTop3D` is modularized in a similar manner to the `PolyTop` code, presented in Talischi et al. (2012b), together with a similar naming convention for its variables. Thus, we refer the readers to Talischi et al. (2012b) for a thorough introduction of the structure of the code. *We hope that the modularity and flexibility offered by `PolyTop3D` will motivate the community to explore the proposed VEM-based framework in other topology optimization problems.*

In the sequel, we demonstrate the efficiency of the `PolyTop3D` code by benchmarking it with the `Top3D` code by Liu and Tovar (2014). For purpose of comparison, the cantilever example, presented in Table 4 of Liu and Tovar (2014), is solved on a set of three regular hexahedral meshes whose statistics are shown in Table 5. Each element in those meshes is a unit cube. Throughout this study, the filter radius is set as $R = 1.5$ and the volume constraint is

Table 5 Statistics of three meshes for the cantilever problem

Meshes	Dimensions	# Nd	# El	# DV (PolyTop3D)	# DV (Top3D)
Mesh 1	$48 \times 16 \times 12$	10,829	9,216	41,625	9,216
Mesh 2	$72 \times 24 \times 18$	34,675	31,104	135,013	31,104
Mesh 3	$96 \times 32 \times 24$	80,025	73,728	313,649	73,728

Table 6 Total runtime comparison of PolyTop3D with the Top3D code

	Mesh 1	Mesh 2	Mesh 3
PolyTop3D	340.48	2275.20	13517.09
Top3D	296.22	2085.20	12797.00

The times are reported in seconds for 200 optimization iterations

Table 7 Breakdown of the PolyTop3D run time from 200 optimization iterations

	Mesh 1	Mesh 2	Mesh 3
Forming $\mathbf{P}^{\mathcal{F}}$ and $\mathbf{P}^{\mathcal{V}}$	7.66 (2.25%)	144.02 (6.33%)	766.34 (5.67%)
Forming VEM shape func.	23.31 (6.85%)	82.53 (3.63%)	199.47 (1.48%)
Assemble $\mathbf{K}_\rho$	124.13 (36.46%)	434.55 (19.10%)	1174.41 (8.69%)
Solving $\mathbf{K}_\rho \mathbf{U} = \mathbf{F}$	156.20 (45.88%)	1508.18 (66.29%)	11097.56 (82.10%)
Compliance sensitivity	22.73 (6.68%)	79.034 (3.47%)	210.53 (1.56%)
OC update	2.15 (0.63%)	12.61 (0.55%)	31.46 (0.23%)

The times are in seconds with percentage of total runtime provided in parentheses

taken to be $\overline{V} = 15\%$. For both computer codes, a constant penalty parameter of $p = 3$ is used and 200 optimization iterations are carried out on a desktop computer with an Intel(R) Xeon(R), 3.00 GHz processor and 256 GB of RAM running MATLAB R2016a. For all the meshes, the two codes produce almost identical final topologies and thus are not shown here for the sake of conciseness.

In Table 6, we present a comparison of the total runtimes of PolyTop3D and Top3D for the three meshes. In addition, Table 7 shows the breakdown of the total runtime of the PolyTop3D code into major steps. One immediate conclusion from Tables 6 and 7 is that the PolyTop3D code is able to achieve similar efficiency to the Top3D code using more than four times the number of DVs. The major runtime difference of the two codes comes from the steps of forming projection matrices, $\mathbf{P}^{\mathcal{F}}$ and $\mathbf{P}^{\mathcal{V}}$ (c.f. (56) and (60)), and VEM shape functions φ_i.

References

Abdelkader A, Bajaj CL, Ebeida MS, Mahmoud AH, Mitchell SA, Owens JD, Rushdi AA (2018) Sampling conditions for conforming voronoi meshing by the vorocrust algorithm. arXiv:1803.06078

Ahmad B, Alsaedi A, Brezzi F, Marini LD, Russo A (2013) Equivalent projectors for virtual element methods. Comput Math Appl 66(3):376–391

Andreassen E, Clausen A, Schevenels M, Lazarov BS, Sigmund O (2011) Efficient topology optimization in matlab using 88 lines of code. Struct Multidiscip Optim 43(1):1–16

Antonietti PF, Manzini G, Verani M (2018) The fully nonconforming virtual element method for biharmonic problems. Math Models Methods Appl Sci 28(02):387–407

Antonietti PF, Bruggi M, Scacchi S, Verani M (2017) On the virtual element method for topology optimization on polygonal meshes: a numerical study. Comput Math Appl 74(5):1091–1109

Argyris JH, Pister KS, Szimmat J, Vaz LE, Willam KJ (1978) Finite element analysis of inelastic structural behaviour. Nuclear Eng Des 46(1):235–262

Arroyo M, Ortiz M (2006) Local maximum-entropy approximation schemes: a seamless bridge between finite elements and meshfree methods. Int J Numer Methods Eng 65(13):2167–2202

Artioli E, Beirão da Veiga L, Lovadina C, Sacco E (2017) Arbitrary order 2D virtual elements for polygonal meshes: part II, inelastic problem. Comput Mech 60(4):643–657

Artioli E, De Miranda S, Lovadina C, Patruno L (2017) A stress/displacement virtual element method for plane elasticity problems. Comput Methods Appl Mech Eng 325:155–174

Barber JR (2010) Elasticity, 3rd edn. Springer, Berlin

Beirão da Veiga L, Brezzi F, Marini LD (2013) Virtual elements for linear elasticity problems. SIAM J Numer Anal 51(2):794–812

Beirão da Veiga L, Lovadina C, Mora D (2015) A virtual element method for elastic and inelastic problems on polytope meshes. Comput Methods Appl Mech Eng 295:327–346

Beirão da Veiga L, Dassi F, Russo A (2017) High-order virtual element method on polyhedral meshes. Comput Math Appl 74:1110–1122

Beirão da Veiga L, Brezzi F, Marini L, Russo A (2014) The hitchhiker's guide to the virtual element method. Math Models Methods Appl Sci 24(08):1541–1573

Beirão da Veiga L, Brezzi F, Dassi F, Marini L, Russo A (2017) Virtual element approximation of 2D magnetostatic problems. Comput Methods Appl Mech Eng 327:173–195
Beirão da Veiga L, Brezzi F, Dassi F, Marini L, Russo A (2018) Lowest order Virtual Element approximation of magnetostatic problems. Comput Methods Appl Mech Eng 332:343–362
Beirão da Veiga L, Brezzi F, Cangiani A, Manzini G, Marini LD, Russo A (2013) Basic principles of virtual element methods. Math Models Methods Appl Sci 23(1):199–214
Belytschko T, Xiao S, Parimi C (2003) Topology optimization with implicit functions and regularization. Int J Numer Methods Eng 57(8):1177–1196
Bendsøe MP (1989) Optimal shape design as a material distribution problem. Struct Optim 1(4):193–202
Bendsøe MP, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 71(2):197–224
Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9-10):635–654
Bendsoe MP, Sigmund O (2013) Topology optimization: theory, methods, and applications, Springer Science & Business Media
Benedetto MF, Caggiano A, Etse G (2018) Virtual elements and zero thickness interface-based approach for fracture analysis of heterogeneous materials. Comput Methods Appl Mech Eng 338:41–67
Bishop J (2014) A displacement-based finite element formulation for general polyhedra using harmonic shape functions. Int J Numer Methods Eng 97(1):1–31
Brezzi F, Marini LD (2013) Virtual Element Methods for plate bending problems. Comput Methods Appl Mech Eng 253:455–462
Bourdin B (2001) Filters in topology optimization. Int J Numer Methods Eng 50(9):2143–2158
Christensen P, Klarbring A (2009) An introduction to structural optimization, Springer Science & Business Media Linköping
Chin EB, Lasserre JB, Sukumar N (2015) Numerical integration of homogeneous functions on convex and nonconvex polygons and polyhedra. Comput Mech 56(6):967–981
Chi H, Beirão da Veiga L, Paulino G (2017) Some basic formulations of the virtual element method (VEM) for finite deformations. Comput Methods Appl Mech Eng 318:148–192
Chi H, Talischi C, Lopez-Pamies O, Paulino GH (2016) A paradigm for higher order polygonal elements in finite elasticity. Comput Methods Appl Mech Eng 306:216–251
De Bellis ML, Wriggers P, Hudobivnik B, Zavarise G (2018) Virtual element formulation for isotropic damage. Finite Elem Anal Des 144:38–48
Diaz A, Sigmund O (1995) Checkerboard patterns in layout optimization. Struct Optim 10(1):40–45
Filipov ET, Chun J, Paulino GH, Song J (2016) Polygonal multiresolution topology optimization (PolyMTOP) for structural dynamics. Struct Multidiscip Optim 53(4):673–694
Floater M, Kós G, Reimers M (2005) Mean value coordinates in 3D. Comput Aided Geom Des 22(7):623–631
Floater M, Gillette A, Sukumar N (2014) Gradient bounds for Wachspress coordinates on polytopes. SIAM J Numer Anal 52(1):515–532
Gain AL, Talischi C, Paulino GH (2014) On the virtual element method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. Comput Methods Appl Mech Eng 282:132–160
Gain AL, Paulino GH, Duarte LS, Menezes IF (2015) Topology optimization using polytopes. Comput Methods Appl Mech Eng 293:411–430
Groen JP, Langelaar M, Sigmund O, Ruess M (2017) Higher-order multi-resolution topology optimization using the finite cell method. Int J Numer Methods Eng 110(10):903–920
Guest JK, Smith LCG (2010) Reducing dimensionality in topology optimization using adaptive design variable fields. Int J Numer Methods Eng 81(8):1019–1045
Guest JK, Prévost JH, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int J Numer Methods Eng 61(2):238–254
Haftka RT, Gürdal Z (2012) Elements of structural optimization, Vol. 11, Springer Science & Business Media
Hormann K, Sukumar N (2018) Generalized barycentric coordinates in computer graphics and computational mechanics, CRC Press
Hormann K, Sukumar N (2008) Maximum entropy coordinates for arbitrary polytopes. In: Eurographics symposium on geometry processing, vol 27, pp 1513–1520
Hoshina TYS, Menezes IFM, Pereira A (2018) A simple adaptive mesh refinement scheme for topology optimization using polygonal meshes. J Braz Soc Mech Sci Eng 40(7):348. https://doi.org/10.1007/s40430-018-1267-5
Jang GW, Lee S, Kim YY, Sheen D (2005) Topology optimization using non-conforming finite elements: three-dimensional case. Int J Numer Methods Eng 63(6):859–875
Jang GW, Jeong JH, Kim YY, Sheen D, Park C, Kim MN (2003) Checkerboard-free topology optimization using non-conforming finite elements. Int J Numer Methods Eng 57(12):1717–1735
Jog CS, Haber RB (1996) Stability of finite element models for distributed-parameter optimization and topology design. Comput Methods Appl Mech Eng 130(3-4):203–226
Kang Z, Wang Y (2011) Structural topology optimization based on non-local Shepard interpolation of density field. Comput Methods Appl Mech Eng 200(49-52):3515–3525
Liu K, Tovar A (2014) An efficient 3d topology optimization code written in matlab. Struct Multidiscip Optim 50(6):1175–1196
Manzini G, Russo A, Sukumar N (2014) New perspective on polygonal and polyhedral finite element method. Math Models Methods Appl Sci 24(08):1665–1699
Martin S, Kaufmann P, Botsch M, Wicke M, Gross M (2008) Polyhedral finite elements using harmonic basis functions. In: SGP '08 proceedings of the symposium on geometry processing, vol 27, issue 5, pp 1521–1529
Matsui K, Terada K (2004) Continuous approximation of material distribution for topology optimization. Int J Numer Methods Eng 59(14):1925–1944
Mora D, Rivera G, Velásquez I (2018) A virtual element method for the vibration problem of kirchhoff plates. ESAIM: M2AN 52(4):1437–1456
Nguyen TH, Paulino GH, Song J, Le CH (2010) A computational paradigm for multiresolution topology optimization (MTOP). Struct Multidiscip Optim 41(4):525–539
Nguyen TH, Paulino GH, Song J, Le CH (2012) Improving multiresolution topology optimization via multiple discretizations. Int J Numer Methods Eng 92(6):507–530
Nguyen-Xuan H (2017) A polytree-based adaptive polygonal finite element method for topology optimization. Int J Numer Methods Eng 110(10):972–1000
Paulino GH, Le CH (2009) A modified Q4/Q4 element for topology optimization. Struct Multidiscip Optim 37(3):255–264
Paulino GH, Gain AL (2015) Bridging art and engineering using Escher-based virtual elements. Struct Multidiscip Optim 51(4):867–883
Pereira A, Talischi C, Paulino GH, Menezes IF, Carvalho MS (2016) Fluid flow topology optimization in polytop: stability and computational implementation. Struct Multidiscip Optim 54(5):1345–1364
Pouderoux J, Charest M, Kenamond M, Shashkov M (2017) 2D & 3D voronoi meshes generation with ShaPo. In: The 8th international conference on numerical methods for multi-material fluid flow (MULTIMAT 2017)

Rahmatalla SF, Swan CC (2004) A Q4/Q4 continuum structural topology optimization implementation. Struct Multidiscip Optim 27(1-2):130–135

Rozvany G (2009) A critical review of established methods of structural topology optimization. Struct Multidiscip Optim 37(3):217–237

Rozvany GIN, Zhou M, Birker T (1992) Generalized shape optimization without homogenization. Struct Optim 4(3-4):250–252

Sigmund O (2001) A 99 line topology optimization code written in matlab. Struct Multidiscip Optim 21(2):120–127

Sigmund O, Maute K (2013) Topology optimization approaches. Struct Multidiscip Optim 48(6):1031–1055

Sigmund O, Torquato S, Aksay I (1998) On the design of 1–3 piezocomposites using topology optimization. J Mater Res 13(04):1038–1048

Sutton OJ (2017) The virtual element method in 50 lines of matlab. Numer Algorithms 75(4):1141–1159

Talischi C, Paulino GH (2014) Addressing integration error for polygonal finite elements through polynomial projections: A patch test connection. Math Models Methods Appl Sci 24(08):1701–1727

Talischi C, Paulino GH, Le CH (2009) Honeycomb Wachspress finite elements for structural topology optimization. Struct Multidiscip Optim 37(6):569–583

Talischi C, Paulino GH, Pereira A, Menezes IFM (2010) Polygonal finite elements for topology optimization: A unifying paradigm. Int J Numer Methods Eng 82:671–698

Talischi C, Paulino GH, Pereira A, Menezes IFM (2012a) PolyTop: A Matlab implementation of a general topology optimization framework using unstructured polygonal finite element meshes. Struct Multidiscip Optim 45(3):329–357

Talischi C, Paulino GH, Pereira A, Menezes IFM (2012b) PolyMesher: a general-purpose mesh generator for polygonal elements written in Matlab. Struct Multidiscip Optim 45(3):309–328

Talischi C, Pereira A, Menezes I, Paulino GH (2015) Gradient correction for polygonal and polyhedral finite elements. Int J Numer Methods Eng 102(3-4):728–747

Taylor RL, Artioli E (2018) Vem for inelastic solids. In: Oñate E, Peric D, de Souza Neto E, Chiumenti M. (eds) Advances in computational plasticity: A book in honour of D. Roger J. Owen. Springer, Cham, pp 381–394

Thedin RS, Pereira A, Menezes IF, Paulino GH (2014) Polyhedral mesh generation and optimization for finite element computations. In: Proceedings of the Iberian Latin-American congress on computational methods in engineering, Fortaleze, CE, Brazil, November 23-26

Vaz LE, Hinton E (1995) FE-shape sensitivity of elastoplastic response. Struc Optim 10(3-4):231–238

Wang Y, Kang Z, He Q (2013) An adaptive refinement approach for topology optimization based on separated density field description. Comput Struct 117:10–22

Wei P, Li Z, Li X, Wang MY (2018) An 88-line matlab code for the parameterized level set method based topology optimization using radial basis functions, Structural and Multidisciplinary Optimization, pp 1–19

Wriggers P, Hudobivnik B (2017) A low order virtual element formulation for finite elasto-plastic deformations. Comput Methods Appl Mech Eng 327, 459–477

Wriggers P, Hudobivnik B, Korelc J (2018) Efficient low order virtual elements for anisotropic materials at finite strains. In: Oñate E, Peric D, de Souza Neto E, Chiumenti M (eds) Advances in computational plasticity: a book in Honour of D. Roger J. Owen, vol 46. Springer, Cham, pp 417–434

Wriggers P, Reddy B, Rust W, Hudobivnik B (2017) Efficient virtual element formulations for compressible and incompressible finite deformations. Comput Mech 60:253–268

Zegard T, Paulino GH (2016) Bridging topology optimization and additive manufacturing. Struct Multidiscip Optim 53(1):175–192

Zhao J, Chen S, Zhang B (2016) The nonconforming virtual element method for plate bending problems. Math Models Methods Appl Sci 26(09):1671–1687