Adv Comput Math (2022) 48: 3
https://doi.org/10.1007/s10444-021-09913-3

The role of mesh quality and mesh quality indicators in the virtual element method

T. Sorgente[1] · S. Biasotti[1] · G. Manzini[1] · M. Spagnuolo[1]

Received: 28 January 2021 / Accepted: 11 November 2021 / Published online: 27 December 2021

Abstract

Since its introduction, the virtual element method (VEM) was shown to be able to deal with a large variety of polygons, while achieving good convergence rates. The regularity assumptions proposed in the VEM literature to guarantee the convergence on a theoretical basis are therefore quite general. They have been deduced in analogy to the similar conditions developed in the finite element method (FEM) analysis. In this work, we experimentally show that the VEM still converges, with almost optimal rates and low errors in the L^2, H^1 and L^∞ norms, even if we significantly break the regularity assumptions that are used in the literature. These results suggest that the regularity assumptions proposed so far might be overestimated. We also exhibit examples on which the VEM sub-optimally converges or diverges. Finally, we introduce a mesh quality indicator that experimentally correlates the entity of the violation of the regularity assumptions and the performance of the VEM solution, thus predicting if a mesh is potentially critical for VEM.

Keywords Virtual element method · Polygonal mesh · Mesh regularity assumptions · Mesh quality indicators · Small edges · 2D Poisson problem · Optimal convergence

Communicated by: Lourenco Beirao da Veiga

✉ T. Sorgente
tommaso.sorgente@ge.imati.cnr.it

S. Biasotti
silvia.biasotti@ge.imati.cnr.it

G. Manzini
marco.manzini@imati.cnr.it

M. Spagnuolo
michela.spagnuolo@ge.imati.cnr.it

[1] Istituto di Matematica Applicata e Tecnologie Informatiche 'E. Magenes', Consiglio Nazionale delle Ricerche, Via De Marini 6, 16149 Genoa, Italy

Mathematics Subject Classification (2010) 65N12 · 65N30 · 65N50 · 68U05

1 Introduction

Finite element methods are very successful in the numerical treatment of partial differential equations (PDEs), but their formulation requires an explicit knowledge of the basis functions. Consequently, they are mostly restricted to meshes with elements having a simple geometrical shape, such as triangles or quadrilaterals. This restriction is overcome by polytopal element methods such as the VEM, which are designed to provide arbitrary order of accuracy on more generally shaped elements. In the VEM setting, we partition the computational domain into polytopal elements and the explicit knowledge of the basis functions is not required, since the VEM formulation and its practical implementation is based on suitable polynomial projections that are always computable from a careful choice of the degrees of freedom.

The VEM was originally formulated in [23] as a conforming FEM for the Poisson problem by rewriting in a variational setting the *nodal* mimetic finite difference (MFD) method [8, 16, 26, 31] for solving diffusion problems on unstructured polygonal meshes. A survey on the MFD method can be found in the review paper [14] and the research monograph [27]. The VEM scheme inherits the flexibility of the MFD method with respect to the admissible meshes and this feature is well reflected in the many significant applications that have been developed so far, see, for example, [4, 25, 29, 30, 33]. Because of its origins, the VEM is intimately connected with other finite element approaches. The connection between the VEM and finite elements on polygonal/polyhedral meshes is thoroughly investigated in [17], between the VEM and the BEM-based FEM in [10]. The VEM has been extended to convection-reaction-diffusion problems with variable coefficients in [25].

Optimal convergence rates for the virtual element approximations of the Poisson equation were proved in H^1 and L^2 norms, see for instance [2, 6, 7, 11, 23, 28, 32]. The theoretical results behind the VEM convergence rate involve an estimate of the approximation error, which is due to both analytical assumptions (interpolation and polynomial projections of the virtual element functions) and geometrical assumptions (the geometrical shape of the mesh elements). There is a general concordance in the literature about the analytical assumptions, but the understanding of which geometrical features of the mesh elements influence the most on the approximation error and the convergence rate, is still an open issue. Various geometrical (or *regularity*) assumptions have been proposed to ensure that all elements of any mesh of a given mesh family in the refinement process are sufficiently regular. These assumptions guarantee the VEM convergence and optimal estimates of the approximation error with respect to different norms. However, as already observed from the very first papers, cf. [2], the VEM seems to maintain its optimal convergence rates also when we use mesh families that do not satisfy the usual geometrical assumptions.

As a first contribution of this paper, we overview the geometrical assumptions introduced in the literature of the conforming VEM for the Poisson equation to guarantee the convergence. Other geometrical assumptions can be found, either for other formulations of the method (see [5] for the non-conforming VEM) or for specific

types of elements (see [11] for anisotropic elements). Then, we define a mesh generation framework that allows us to build sequences of meshes (datasets) gradually introducing several pathologies. The so-generated datasets systematically violate the geometrical assumptions, and enhance a correlation analysis between such assumptions and the VEM performance. We experimentally show how the VEM presents a good convergence rate on most examples and only fails in very few situations. We also provide an indicator of the violation of the geometrical assumptions, which depends uniquely on the geometry of the mesh elements. We show a correspondence between this indicator and the performance of the VEM on a given mesh, or mesh family, in terms of approximation error and convergence rate. Our work is focused on developing a strategy to evaluate if a given sequence of meshes is suited to the virtual element discretization, and possibly to predict the behavior of the numerical discretization *before* applying the method. In this sense, we can consider the approach that we present in this work as more in an *a priori* than an *a posteriori* setting.

The paper is organized as follows. In Section 2, we present the VEM and the convergence results for the Poisson equation with Dirichlet boundary conditions. In Section 3, we detail the geometrical assumptions on the mesh elements that are used in the literature to guarantee the convergence of the VEM. In Section 4, we present a number of datasets which do not satisfy these assumptions, and experimentally investigate the convergence of the VEM over them. In Section 5, we propose a mesh quality indicator to predict the behavior of the VEM over a given dataset. In Section 6, we offer our concluding remarks and discuss future developments and work. In the Appendix we present the algorithmic procedures that we used to build the datasets. All the meshes used in this work are available for download at https://github.com/TommasoSorgente/vem-quality-dataset.

1.1 Notation and technicalities

We use the standard definition and notation of Sobolev spaces, norms and seminorms, cf. [1]. Let k be a nonnegative integer number. The Sobolev space $H^k(\omega)$ consists of all square integrable functions with all square integrable weak derivatives up to order k that are defined on the open, bounded, connected subset ω of $\mathbb{R}^d$, $d = 1, 2$. As usual, if $k = 0$, we prefer the notation $L^2(\omega)$. Norm and seminorm in $H^k(\omega)$ are denoted by $||\cdot||_{k,\omega}$ and $|\cdot|_{k,\omega}$, while for the inner product in $L^2(\omega)$ we prefer the integral notation. We denote the space of polynomials of degree less than or equal to $k \geq 0$ on ω by $\mathbb{P}_k(\omega)$ and conventionally assume that $\mathbb{P}_{-1}(\omega) = \{0\}$. In our implementation, we consider the orthogonal basis on every mesh edge through the univariate Legendre polynomials and inside every mesh cell provided by the Gram-Schmidt algorithm applied to the standard monomial basis.

Finally, throughout the paper we use the letter C in the error inequalities to denote a real, positive constant that can have a different value at any occurrence. This constant may depend on the model and on some discretization parameters, such as the coercivity and stability constants of the bilinear form and of the linear functional used in the variational formulation, the mesh regularity constants used when defining the properties of the mesh families to which the numerical method is suitable, and the polynomial order of the method. Nevertheless, this constant is always assumed to

be independent of the mesh size parameter h that characterizes the mesh and will be introduced in the next section.

2 The virtual element method

We investigate the performance of the VEM on the elliptic model problem provided by the Poisson equation with Dirichlet boundary conditions. In this section, we briefly review the model equations in strong and weak form and the formulation of the virtual element approximation.

The Poisson equation and the virtual element approximation Let Ω be an open, bounded, connected subset of $\mathbb{R}^2$ with polygonal boundary Γ. Consider the Poisson equation with homogeneous Dirichlet boundary conditions in strong from:

$$-\Delta u = f \qquad \text{in } \Omega, \tag{1}$$

$$u = 0 \qquad \text{on } \Gamma. \tag{2}$$

The variational formulation of problem (1)–(2) reads as: *Find* $u \in H_0^1(\Omega)$ *such that*

$$a(u, v) = F(v) \qquad \forall v \in H_0^1(\Omega), \tag{3}$$

where the bilinear form $a(\cdot, \cdot) : H^1(\Omega) \times H^1(\Omega) \to \mathbb{R}$ is given by

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, d\mathbf{x} \tag{4}$$

and the right-hand side linear functional $F : L^2(\Omega) \to \mathbb{R}$ is given by

$$F(v) = \int_\Omega f v \, d\mathbf{x}, \tag{5}$$

with the (implicit) assumption that $f \in L^2(\Omega)$. The well-posedness of the discrete formulation (3) stems from the coercivity and continuity of the bilinear form $a(\cdot, \cdot)$, the continuity of the right-hand side linear functional $F(\cdot)$, and the application of the Lax-Milgram theorem [19, Section 2.7].

The numerical method that we consider in this paper is mainly based on References [2, 23], and provides an optimal approximation on polygonal meshes when the diffusion coefficient is variable in space. To ease the presentation, we consider the case of homogeneous Dirichlet boundary conditions, the extension to the non-homogeneous case being deemed as straightforward. Such a case is also considered in the numerical experiments carried out in this paper.

The virtual element approximation of (3) reads as: *Find* $u_h \in V_k^h$ *such that*

$$a_h(u_h, v_h) = F_h(v_h) \qquad v_h \in V_k^h, \tag{6}$$

where u_h, V_k^h, $a_h(\cdot, \cdot)$, $F_h(\cdot)$ are the virtual element approximations of u, $H_0^1(\Omega)$, $a(\cdot, \cdot)$, and $F(\cdot)$. We review the construction of these mathematical objects in the rest of this section.

Mesh notation Let $\mathcal{T} = \{\Omega_h\}_{h\in\mathcal{H}}$ be a set of decompositions Ω_h of the computational domain Ω into a finite set of nonoverlapping polygonal elements E. We refer to $\mathcal{T}$ as the *mesh family* and to each one of its members Ω_h as the *mesh*. The subindex label h, indicating the *mesh size*, is the maximum of the diameters of the mesh elements, defined by $h_E = \sup_{\mathbf{x},\mathbf{y}\in E} |\mathbf{x}-\mathbf{y}|$. We assume that the mesh sizes of the mesh family $\mathcal{T}$ are in a countable subset $\mathcal{H}$ of the real line $(0, +\infty)$ having 0 as its unique accumulation point. Each element E has a nonintersecting polygonal boundary ∂E formed by straight edges e, center of gravity $\mathbf{x}_E = (x_E, y_E)$ and area $|E|$. We denote the edge mid-point $\mathbf{x}_e = (x_e, y_e)$ and its length $|e|$, and with a small abuse of notation, we write $e \in \partial E$ to indicate that edge e is running throughout the set of edges forming the elemental boundary ∂E. The convergence analysis of the VEM and the derivation of the error estimates in the L^2 and H^1 norms require a few suitable assumptions on the mesh family $\mathcal{T}$. Such assumptions are discussed in detail in the next section. On every mesh Ω_h, given an integer $k \geq 0$, we define the space of piecewise discontinuous polynomials of degree k, $\mathbb{P}_k(\Omega_h)$, containing the functions q such that $q_{|E} \in \mathbb{P}_k(E)$ for every $E \in \Omega_h$.

The virtual element spaces Let $k \geq 1$ be an integer number and $E \in \Omega_h$ a generic mesh element. The *conforming virtual element space V_k^h of order k built on mesh Ω_h* is obtained by gluing together the elemental approximation spaces denoted by $V_k^h(E)$, so that

$$V_k^h := \left\{ v_h \in H_0^1(\Omega) \,:\, v_{h|E} \in V_k^h(E) \;\; \forall E \in \Omega_h \right\}. \tag{7}$$

The local virtual element space $V_k^h(E)$ is defined in accordance with the *enhancement strategy* introduced in [2]:

$$V_k^h(E) = \Bigg\{ v_h \in H^1(E) \,:\, v_{h|\partial E} \in C^0(\partial E),\; v_{h|e} \in \mathbb{P}_k(e)\, \forall e \in \partial E, \\ \Delta v_h \in \mathbb{P}_k(E), \text{ and} \\ \int_E (v_h - \Pi_k^{\nabla,E} v_h)\, q\, dV = 0 \;\forall q \in \mathbb{P}_k(E)\backslash\mathbb{P}_{k-2}(E) \Bigg\}, \tag{8}$$

where $\Pi_k^{\nabla,E}$ is the elliptic projection that will be discussed in the next section and $\mathbb{P}_k(E)\backslash\mathbb{P}_{k-2}(E)$ is the space of polynomials of degree equal to $k-1$ and k. We recall that $\mathbb{P}_k(E)$ and $\mathbb{P}_k(e)$ are the linear spaces $\mathbb{P}_k$ respectively defined over an element E or an edge e according to our notation. By definition, the space $V_k^h(E)$ contains $\mathbb{P}_k(E)$ and the global space V_k^h is a conforming subspace of $H^1(\Omega)$.

The elliptic projection operators The definition in (8) requires the *elliptic projection operator* $\Pi_k^{\nabla,E} : H^1(E) \to \mathbb{P}_k(E)$, which, for any $v_h \in V_k^h(E)$, is given by:

$$\int_E \nabla \Pi_k^{\nabla,E} v_h \cdot \nabla q\, dV = \int_E \nabla v_h \cdot \nabla q\, dV \quad \forall q \in \mathbb{P}_k(E), \tag{9}$$

$$\int_{\partial E} \left(\Pi_k^{\nabla,E} v_h - v_h\right) dS = 0. \tag{10}$$

Equation (10) allows us to remove the kernel of the gradient operator from the definition of $\Pi_k^{\nabla,E}$, so that the k-degree polynomial $\Pi_k^{\nabla,E} v_h$ is uniquely defined for every virtual element function $v_h \in V_k^h(E)$. Moreover, projector $\Pi_k^{\nabla,E}$ is a polynomial-preserving operator, i.e., $\Pi_k^{\nabla,E} q = q$ for every $q \in \mathbb{P}_k(E)$. We can also define a global projection operator $\Pi_k^{\nabla} : H^1(\Omega) \to \mathbb{P}_k(\Omega_h)$, which is such that $\Pi_k^{\nabla} v_{h|E} = \Pi_k^{\nabla,E}(v_{h|E})\ \forall E \in \Omega_h$. A major property of the elliptic projection operator is that every projection $\Pi_k^{\nabla,E} v_h$ of a virtual element function $v_h \in V_k^h(E)$ is computable from the degrees of freedom of v_h associated with element E that are defined as follows.

The degrees of freedom. The degrees of freedom of a virtual element function $v_h \in V_k^h(E)$ are given by the following set of values [23]:

(D1) for $k \geq 1$, the values of v_h at the vertices of E;
(D2) for $k \geq 2$, the values of v_h at the $k-1$ internal points of the $(k+1)$-point Gauss-Lobatto quadrature rule on every edge $e \in \partial E$;
(D3) for $k \geq 2$, the cell moments of v_h of order up to $k-2$ on element E:

$$\frac{1}{|E|} \int_E v_h\, q\, dV \quad \forall q \text{ in a basis of } \mathbb{P}_{k-2}(E). \tag{11}$$

These set of values are unisolvent in $V_k^h(E)$, cf. [23]; hence, every virtual element function is uniquely identified by them. The degrees of freedom of a virtual element function in the global space V_k^h are given by collecting the elemental degrees of freedom **(D1)**-**(D3)**. Their unisolvence in V_k^h is an immediate consequence of their unisolvence in every elemental space $V_k^h(E)$. In our implementation of the VEM we considered the standard basis of the scaled monomials and the orthogonal polynomials, and we used the monomials in the numerical experiments.

Orthogonal projections From the degrees of freedom of a virtual element function $v_h \in V_k^h(E)$ we can also compute the orthogonal projections $\Pi_k^{0,E} v_h \in \mathbb{P}_k(E)$, cf. [2]. In fact, the orthogonal projection $\Pi_k^{0,E} v_h$ of a function $v_h \in V_k^h(E)$ is the solution of the variational problem:

$$\int_E \Pi_k^{0,E} v_h\, q\, dV = \int_E v_h\, q\, dV \qquad \forall q \in \mathbb{P}_k(E). \tag{12}$$

The right-hand side is the integral of v_h against the polynomial q, and is computable from the degrees of freedom **(D3)** of v_h when q is a polynomial of degree up to $k-2$, and from the moments of $\Pi_k^{\nabla,E} v_h$ when q is a polynomial of degree $k-1$ and k, cf. (8). Clearly, the orthogonal projection $\Pi_{k-1}^{0,E} v_h$ is also computable. As we have done for the elliptic projection, we can also define a global projection operator $\Pi_k^0 : L^2(\Omega) \to \mathbb{P}_k(\Omega_h)$, which projects the virtual element functions on the space of discontinuous polynomials of degree at most k built on mesh Ω_h. This operator is given by taking the elemental L^2-orthogonal projection $\Pi_k^{0,E} v_h$ in every mesh element E, so that $\left(\Pi_k^0 v_h\right)_{|E} = \Pi_k^{0,E}(v_{h|E})$, which is computable from the degrees of freedom of v_h associated with element E.

Approximation properties in the virtual element space Under a suitable regularity assumption on the mesh family used in the formulation of the VEM (assumption **G1** that will be the topic of the next section), we can prove the following estimates on the projection and interpolation operators:

1. for every s with $1 \leq s \leq k+1$ and for every $w \in H^s(E)$ there exists a $w_\pi \in \mathbb{P}_k(E)$ such that
$$|w - w_\pi|_{0,E} + h_E\,|w - w_\pi|_{1,E} \leq C h_E^s\,|w|_{s,E}\,; \tag{13}$$
2. for every s with $2 \leq s \leq k+1$, for every h, for all $E \in \Omega_h$ and for every $w \in H^s(E)$ there exists a $w_I \in V_k^h(E)$ such that
$$|w - w_I|_{0,E} + h_E\,|w - w_I|_{1,E} \leq C h_E^s\,|w|_{s,E}\,. \tag{14}$$

In these inequalities, C is a real positive constant depending only on the polynomial degree k and on some mesh regularity constants that we will introduce and discuss in the next section.

The virtual element bilinear forms The elliptic and orthogonal projections are needed to define the virtual element bilinear form $a_h(\cdot,\cdot) : V_k^h \times V_k^h \to \mathbb{R}$, and the forcing term $F_h : V_k^h \to \mathbb{R}$. Following the "VEM gospel," we write the discrete bilinear form $a_h(\cdot,\cdot)$ as the sum of elemental contributions

$$a_h(u_h, v_h) = \sum_{E\in\Omega_h} a_h^E(u_h, v_h), \tag{15}$$

where every elemental contribution is a bilinear form $a_h^E(\cdot,\cdot) : V_k^h(E) \times V_k^h(E) \to \mathbb{R}$ designed to approximate the corresponding elemental bilinear form $a^E(\cdot,\cdot) : H^1(E) \times H^1(E) \to \mathbb{R}$

$$a^E(v, w) = \int_E \nabla v \cdot \nabla w\, dV, \quad \forall v, w \in H^1(E).$$

The bilinear form $a_h^E(\cdot,\cdot)$ on each element E is given by

$$a_h^E(u_h, v_h) = \int_E \nabla \Pi_k^{\nabla,E} u_h \cdot \nabla \Pi_k^{\nabla,E} v_h\, dV + S_h^E\Big(\big(I - \Pi_k^{\nabla,E}\big)u_h, \big(I - \Pi_k^{\nabla,E}\big)v_h\Big). \tag{16}$$

The bilinear form $S_h^E(\cdot,\cdot)$ in the definition of $a_h^E(\cdot,\cdot)$ provides the stability term and can be any computable, symmetric, positive definite bilinear form defined on $V_k^h(E)$ for which there exist two positive constants c_* and c^* such that

$$c_* a^E(v_h, v_h) \leq S_h^E(v_h, v_h) \leq c^* a^E(v_h, v_h) \quad \forall v_h \in V_k^h(E) \cap \ker\big(\Pi_k^{\nabla,E}\big). \tag{17}$$

The inequalities in (17) implies that $S_h^E(\cdot,\cdot)$ scales like $a^E(\cdot,\cdot)$ with respect to h_E. Also, the stabilization term in the definition of $a_h^E(\cdot,\cdot)$ is zero if at least one of its two entries is a polynomial of degree (at most) k, since $\Pi_k^{\nabla,E}$ is a polynomial preserving operator.

In our implementation of the VEM, we consider the stabilization proposed in [18]:

$$S_h^E(v_h, w_h) = \sum_{i=1}^{N^{\text{dofs}}} \sigma_i \text{DOF}_i(v_h)\text{DOF}_i(w_h), \tag{18}$$

where $\sigma_i = \max\{\mathcal{A}^E_{ii}, 1\}$ and $\mathcal{A}^E = \left(\mathcal{A}^E_{ij}\right)$ is the matrix resulting from the implementation of the first term in the bilinear form $a^E_h(\cdot,\cdot)$. Let ϕ_i be the i-th "canonical" basis functions generating the virtual element space, which is the function in $V^h_k(E)$ whose i-th degree of freedom for $i = 1, \ldots, N^{\text{dofs}}$ (according to a suitable renumbering of the degrees of freedom in **(D1)**, **(D2)**, and **(D3)**), has value equal to 1 and all other degrees of freedom are zero. These basis functions are unknown in the virtual element framework, but their projections $\Pi^{0,E}_{k-1}\nabla\phi_i$ (and $\nabla\Pi^{\nabla,E}_k\phi_i$) are computable from their degrees of freedom. With this notation, the i,j-th entry of matrix $\mathcal{A}^E$ is given by

$$\mathcal{A}^E_{ij} := a^E\left(\Pi^{\nabla,E}_k\phi_i, \Pi^{\nabla,E}_k\phi_j\right). \tag{19}$$

The stabilization in (18) is sometimes called the "*D-recipe stabilization*" in the virtual element literature, and contains the so-called *dofi-dofi (dd) stabilization* originally proposed in [23] as the special case with $\mathcal{A}_{ii} = 1$:

$$S^{E,\text{dd}}_h(v_h, w_h) = \sum_{i=1}^{N^{\text{dofs}}} \text{DOF}_i(v_h)\text{DOF}_i(w_h). \tag{20}$$

Another possible choice for the stability term, well studied in the literature, is the "*trace stabilization*" proposed in [33] both in the original "H^1-version":

$$S^{\partial E,H^1}_h(v_h, w_h) = h_E \int_{\partial E} \partial_s v_h \partial_s w_h ds, \tag{21}$$

where $\partial_s v_h$ denotes the tangential derivative of v_h along ∂E, and in the "L^2-version":

$$S^{\partial E,L^2}_h(v_h, w_h) = \sum_{e\in\partial E} h_e^{-1} \int_e v_h w_h ds. \tag{22}$$

We explicitly mention other types of stabilization bilinear forms because in Section 4.2 their influence on the performance of the method will be discussed.

The stabilization term, and, in particular, condition (17), is designed so that $a^E_h(\cdot,\cdot)$ satisfies the two fundamental properties:

- *k-consistency*: for all $v_h \in V^h_k(E)$ and for all $q \in \mathbb{P}_k(E)$ it holds that
$$a^E_h(v_h, q) = a^E(v_h, q); \tag{23}$$
- *stability*: there exist two positive constants α_*, α^*, independent of h and E, such that
$$\alpha_* a^E(v_h, v_h) \le a^E_h(v_h, v_h) \le \alpha^* a^E(v_h, v_h) \quad \forall v_h \in V^h_k(E). \tag{24}$$

The virtual element forcing term To approximate the right-hand side of (6), we split it into the sum of elemental contributions and every local linear functional is approximated by using the orthogonal projection $\Pi^{0,E}_k v_h$:

$$F(v_h) = \sum_{E\in\Omega_h} \left(f, \Pi^{0,E}_k v_h\right)_E, \quad \text{where } \left(f, \Pi^{0,E}_k v_h\right)_E = \int_E f\, \Pi^{0,E}_k v_h\, dV. \tag{25}$$

Main convergence properties The well-posedness of the discrete formulation (6) stems from the coercivity of the bilinear form $a_h(\cdot,\cdot)$, the continuity of the right-hand side linear functional $\left(f, \Pi_k^0 \cdot\right)$ and the application of the Lax-Milgram theorem [19, Section 2.7].

In this work, we are interested in checking whether the VEM mantains optimal convergence rates on different mesh families that may display some pathological situations. From a theoretical viewpoint, the convergence estimates hold under some constraints on the shapes of the elements forming the mesh, called *mesh geometrical (or regularity) assumptions*. A summary of the major findings from the literature is available in [22, Section 3.2]; in the next sections we will investigate how breaking such constraints may affect these results.

Let $u \in H^{k+1}(\Omega)$ be the solution to the variational problem (3) on a convex domain Ω with $f \in H^k(\Omega)$. Let $u_h \in V_k^h$ be the solution of the virtual element method (6) on every mesh of a mesh family $\mathcal{T} = \{\Omega_h\}$ satisfying a suitable set of mesh geometrical assumptions. Then, a strictly positive constant C exists such that

– the H^1-error estimate holds:

$$||u - u_h||_{1,\Omega} \leq Ch^k \left(||u||_{k+1,\Omega} + |f|_{k,\Omega}\right); \tag{26}$$

– the L^2-error estimate holds:

$$||u - u_h||_{0,\Omega} \leq Ch^{k+1} \left(||u||_{k+1,\Omega} + |f|_{k,\Omega}\right). \tag{27}$$

Constant C may depend on the stability constants α_* and α^*, on mesh regularity constants which we will introduce in the next section, on the size of the computational domain $|\Omega|$, and on the approximation degree k. Constant C is normally independent of h, but for the most extreme meshes it may depend on the ratio between the longest and shortest edge lengths, cf. [22, Section 3.2].

Finally, we note that the approximate solution u_h is not explicitly known inside the elements. Consequently, in the numerical experiments of Section 4.2, we approximate the error in the L^2-norm as follows:

$$||u - u_h||_{0,\Omega} \approx ||u - \Pi_k^0 u_h||_{0,\Omega}, \tag{28}$$

where $\Pi_k^0 u_h$ is the global L^2-orthogonal projection of the virtual element approximation u_h to u. On its turn, we approximate the error in the energy norm as follows:

$$|u - u_h|_{1,\Omega} \approx ||\nabla u - \Pi_{k-1}^0 \nabla u_h||_{0,\Omega}, \tag{29}$$

where Π_{k-1}^0 is extended component-wisely to the vector fields.

3 Geometrical assumptions

In this section, we review the geometrical assumptions appeared in the VEM literature since their definition in [23]. All the assumptions are defined for a single mesh Ω_h, but the conditions contained in them are required to hold independently of h.

Therefore, when considering a mesh family $\mathcal{T} = \{\Omega_h\}_h$, these assumptions have to be verified simultaneously by every $\Omega_h \in \mathcal{T}$.

It is well-known from the FEM literature that the approximation properties depend on specific assumptions on the geometry of the elements. For example, classical geometrical assumptions for a family of triangulations $(\Omega_h)_{h\to 0}$, are the ones respectively introduced in [12] and [34]:

(a) *Shape regularity condition:* there exists a real number $\gamma \in (0, 1)$, independent of h, such that $\forall E \in \Omega_h$ we have

$$\gamma h_E \leq r_E,$$

where h_E and r_E are, respectively, the longest edge in E and its inradius;

(b) *Minimum angle condition:* there exists $\alpha_0 > 0$, independent of h, such that $\forall E \in \Omega_h$ we have

$$\alpha_E \geq \alpha_0,$$

where α_E is the minimal angle of E.

Similarly, in the VEM we need a set of geometrical assumptions to ensure approximation properties. The first pair of assumptions were proposed in [23] and remained untouched also in [2] and [6]. In these papers, the authors assume that a real constant $\rho \in (0, 1)$ exists, independent of h, such that two conditions hold:

Assumption G1 *Every polygonal cell $E \in \Omega_h$ is star-shaped with respect to a disk with radius ρh_E.*

Assumption G2 *For every polygonal cell $E \in \Omega_h$, the length $|e|$ of every edge $e \in \partial E$ satisfies $|e| \geq \rho h_E$.*

Constant ρ is often referred to as *mesh regularity constant* or *parameter*. Condition **G1** can be weakened in the following way, as specified in [23] and more accurately in [6]:

Assumption G1 - weak *Every polygonal cell $E \in \Omega_h$ is the union of a finite number N of disjoint polygonal subcells $E_1, \ldots, E_N$ such that, for $j = 1, \ldots, N$,*

(a) *element E_j is star-shaped with respect to a disk with radius ρh_{E_j};*

(b) *elements E_j and E_{j+1} share a common edge.*

Assumption **G1** (or **G1 - weak**) is the polygonal extension of the classical conditions for triangular meshes, with h_E indicating the elemental diameter instead of the longest edge. Under assumption **G1 - weak**, and therefore also under **G1**, it can be proved [6] that the simplicial triangulation of E determined by the star-centers (the centers of the disks in **G1** and **G1 - weak**) of $E_1, \ldots, E_N$ satisfies the *shape regularity* and the *minimum angle* conditions. Moreover, for $1 \leq j, k \leq N$ it holds that $h_{E_j}/h_{E_k} \leq \rho^{-|j-k|}$.

These assumptions are more restrictive than necessary, but at the same time they are not particularly demanding, since they allow the method to work on very general decompositions. This fact was already mentioned in the very first papers. For example, in [2, Ahmad et al.] the authors say that:

Actually, we could get away with even more general assumptions, but then it would be long and boring to make precise (among many possible crazy decompositions that nobody will ever use) the ones that are allowed and the ones that are not.

In [28] and [7] assumption **G1** is preserved, but assumption **G2** is substituted by the alternative version:

Assumption G3 *There exists a positive integer N, independent of h, such that the number of edges of every polygonal cell $E \in \Omega_h$ is (uniformly) bounded by N.*

Assumption **G2** implies assumption **G3**. However, assumption **G3** is weaker than assumption **G2**, as it allows for edges arbitrarily small with respect to the element diameter. Both assumption pairs **G1+G2** and **G1+G3** imply that the number of vertices of E and the minimum angle of the simplicial triangulation of E given by connecting the vertices of E and its star-center, are controlled by ρ.

Another step forward in the direction of refining the geometrical assumptions has been made in [32]. In addition to assumption **G1**, the authors imagine to unwrap the boundary ∂E of each polygon $E \in \Omega_h$ onto an interval I_E of the real line, obtaining a one-dimensional mesh $\mathcal{I}_E$. The collection of the unwrapped boundaries of all elements in a mesh Ω_h is denoted by $\{\mathcal{I}_E\}_{E\in\Omega_h}$. Moreover, each one-dimensional mesh $\mathcal{I}_E$ can be subdivided into a number of disjoint sub-meshes $\mathcal{I}_E^1, \ldots, \mathcal{I}_E^N$, corresponding to the edges of E (we consider each $\mathcal{I}_E^i$ as a mesh as it may contain more than one edge, see Fig. 1). Then, the following condition is assumed.

Assumption G4 *For every polygonal cell $E \in \Omega_h$, the family $\{\mathcal{I}_E\}_{E\in\Omega_h}$ is piecewise quasi-uniform, that is:*

(*a*) *each mesh $\mathcal{I}_E$ can be subdivided into at most N disjoint sub-meshes $\mathcal{I}_E^1, \ldots, \mathcal{I}_E^N$, for some $N \in \mathbb{N}$;*

(*b*) *each sub-mesh $\mathcal{I}_E^i$, $i = 1, \ldots, N$, is quasi-uniform: the ratio between the largest and the smallest element in $\mathcal{I}_E^i$ is bounded from above by some $c \in \mathbb{R}^+$ independent of h.*

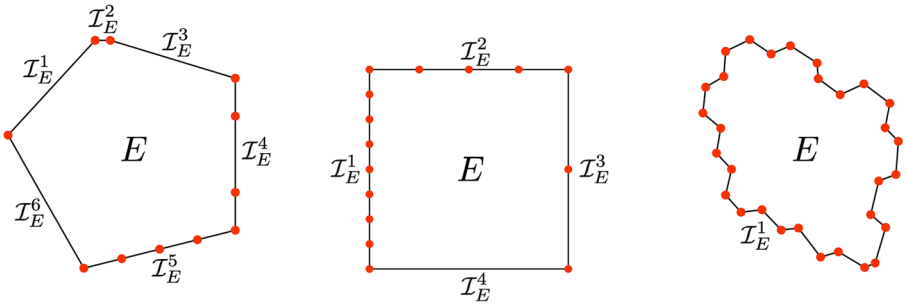

Fig. 1 Examples of admissible elements according to assumption **G4**. Red dots indicate the vertices of the element

Each polygon E is in a one-to-one correspondence to a one-dimensional mesh $\mathcal{I}_E$, but a sub-mesh $\mathcal{I}_E^i \subset \mathcal{I}_E$ might contain more than one edge of E. This implies that assumption **G4** does not require a uniform bound on the number of edges in each element and does not exclude the presence of small edges, cf. Fig. 1. For instance, the mesh families created by agglomeration, cracking, gluing, etc.. of existing meshes are admissible according to **G4**.

According to the literature (see [22, Section 3.2]), possible assumption pairs requested to guarantee the convergence of the VEM are given by combining **G1** (or, equivalently, **G1 - weak**) with either **G2** or **G3** or **G4**.

4 Breaking the geometrical assumptions

In this section, we test the behavior of the virtual element method on a number of mesh "datasets," to stress one or more of the geometrical assumptions discussed in Section 3. We call a *dataset* a collection $\mathcal{D} := \{\Omega_n\}_{n=0,\dots,N}$ of meshes Ω_n covering the domain $\Omega = (0, 1)^2$ such that

- the mesh Ω_{n+1} has smaller mesh size than Ω_n for every $n = 0, \dots, N-1$;
- the meshes Ω_n follow a common refinement pattern, so that they contain similar polygons organized in similar configurations.

Note that each mesh Ω_n is uniquely identified by its size as Ω_h, therefore we can consider a dataset $\mathcal{D}$ as a subset of a mesh family: $\mathcal{D} = \{\Omega_h\}_{h\in\mathcal{H}'} \subset \mathcal{T}$ where $\mathcal{H}'$ is a finite subset of $\mathcal{H}$.

In addition to the violation of the geometrical assumptions, we are also interested in the behavior of the VEM when the measures of mesh elements and edges scale in a nonuniform way in the refinement process. To this end, for each mesh $\Omega_n \in \mathcal{D}$ we define the following quantities and study their trend for $n \to N$:

$$A_n = \frac{\max_{E\in\Omega_n} |E|}{\min_{E\in\Omega_n} |E|} \quad \text{and} \quad e_n = \frac{\max_{e\in\Omega_n} |e|}{\min_{e\in\Omega_n} |e|}. \tag{30}$$

We specifically designed six datasets in order to consider several possible combinations of the geometrical assumptions of the previous section and the scaling indicators A_n and e_n, as shown in Table 1. Note that most of the considered datasets do not fulfill any set of geometrical assumptions required by the convergence analysis found in the literature (see [22, Section 3.2]).

4.1 Datasets definition

We now introduce the datasets, describing for each of them how they are built, which geometrical assumptions they fulfill or violate, and how the indicators A_n and e_n depend on n in the limit for $n \to N$. Each dataset is built around (and often named after) a particular polygonal element contained in it, which is meant to stress one or more assumptions or indicators. The detailed construction algorithms, together with the explicit computations of A_n and e_n for all datasets, can be found in the Appendix.

Table 1 Summary of the geometrical conditions violated and the asymptotic trend of the indices A_n and e_n for all datasets (a is a constant such that $e < a < 3$). Assumption **G1-weak** is not explicitly reported because all the considered datasets that violate **G1**, also violate **G1-weak**

Dataset	$\mathcal{D}_{\text{Triangle}}$	$\mathcal{D}_{\text{Maze}}$	$\mathcal{D}_{\text{Star}}$	$\mathcal{D}_{\text{Jenga}}$	$\mathcal{D}_{\text{Slices}}$	$\mathcal{D}_{\text{Ulike}}$
G1		×	×	×	×	×
G2		×		×		×
G3			×	×		×
G4				×		
A_n	c	a^n	n	2^n	2^n	c
e_n	c	$n\log(n)$	n	2^n	c	2^n

All the meshes presented in this section are publicly accessible at https://github.com/TommasoSorgente/vem-quality-dataset.

Reference dataset The first dataset, $\mathcal{D}_{\text{Triangle}}$, contains only triangular meshes that are built by inserting a number of vertices in the domain through the Poisson Disk Sampling algorithm [9], and connecting them in a Delaunay triangulation (see the Appendix). The refinement is obtained by increasing the number of vertices generated by the Poisson algorithm. The meshes in this dataset do not violate any of the geometrical assumptions and the indicators A_n and e_n are almost constant. We use $\mathcal{D}_{\text{Triangle}}$ as the reference dataset to evaluate the other datasets by comparing the performance of the VEM over them.

Hybrid datasets Next, we consider some hybrid datasets, characterized by a progressive insertion in Ω of one or more identical polygonal elements (called the *initial polygons*), the rest of the domain being tessellated by triangles. These triangles are created by the library *Triangle* [20], bounding the area of the triangular elements with the area of the initial polygons. Steiner points [20] can be added, and the edges of the initial polygons are split when necessary by the insertion of new vertices. The refinement is iterative, with parameters to indicate size, shape and number of the initial polygons; details on this process are provided in the Appendix.

The top and bottom panels of Fig. 2 respectively show the datasets $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$, which we selected as they violate different geometrical assumptions. Other choices for the initial polygons are possible, for instance considering the ones in Benchmark [3].

A "maze" is a 10-sided polygonal element E spiralling around an external point. Progressively, each mesh in $\mathcal{D}_{\text{Maze}}$ contains an increasing number of mazes E with decreasing thickness as $n \to N$. Every E is obviously *not* star-shaped, challenging assumption **G1**. Moreover, the length of the shortest edge e of E decreases faster than the diameter h_E of the polygon. This fact implies, on the one side, that the ratio $|e|/h_E$ of assumption **G2** cannot be bounded from below by a constant ρ that is independent of h, and, on the other side, that assumption **G1-weak** also fails. Indeed, even splitting E into a finite number of rectangles, it is not possible to define a global radius ρ, independent of h, with respect to which the union of these rectangles is

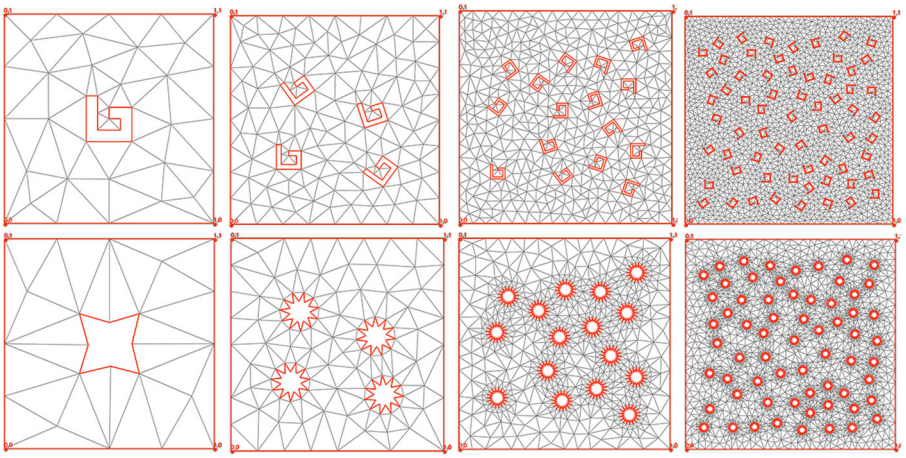

Fig. 2 Meshes Ω_0, Ω_2, Ω_4, Ω_6 from datasets $\mathcal{D}_{\text{Maze}}$ (top) and $\mathcal{D}_{\text{Star}}$ (bottom)

star-shaped according to **G1**, if the shortest edge of E is constantly decreasing. Concerning the scaling indicators, we have $A_n \sim a^n$ for a constant $e < a < 3$ and $e_n \sim n \log(n)$.

Dataset $\mathcal{D}_{\text{Star}}$ is built by inserting star-like polygonal elements, still denoted by E. As $n \to N$, the number of spikes of each E increases and the inner vertices of the star move towards the barycenter of the element. In this case, assumption **G3** is not satisfied because the number of spikes in each E increases from mesh to mesh. Therefore, the total number of vertices and edges in a single element cannot be bounded uniformly.

Last, each star E is star-shaped with respect to the maximum circle inscribed in it. However, as shown in Fig. 3, the radius r of such circle decreases faster than the elemental diameter h_E, therefore it is not possible to define a global $\rho > 0$ able to uniformly bound from below the quantity r/h_E: this violates assumption **G1**. In order to satisfy assumption **G1-weak**, we should split each E into a number of sub-polygons that are star-shaped according to **G1**. Independently of the way we partition E, the number of sub-polygons would always be bigger than or equal to the number of spikes in E, which is constantly increasing. So, the number of sub-polygons would tend to infinity violating condition **G1-weak**. Last, both A_n and e_n scale linearly.

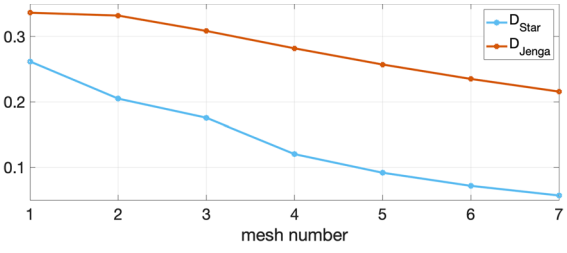

Fig. 3 Ratio r/h_E for datasets $\mathcal{D}_{\text{Star}}$ and $\mathcal{D}_{\text{Jenga}}$

Mirroring datasets Another possible strategy to build a sequence of meshes whose elements are progressively smaller, is to adopt a *mirroring* technique. In practice, we start with the first base mesh $\widehat{\Omega}_0$, which coincides with the first computational mesh Ω_0. At every step $n \geq 1$, we build a new base mesh $\widehat{\Omega}_n$ from the previous base mesh $\widehat{\Omega}_{n-1}$. The computational mesh Ω_n is then obtained by mirroring $\widehat{\Omega}_n$ 4^n times and resizing everything to fit the domain Ω. This construction allows us to obtain a number of vertices and degrees of freedom in each mesh that is comparable to that of the meshes at the same refinement level in datasets $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$.
Examples of meshes from mirrored datasets are presented in Fig. 4; examples of non-mirrored base meshes are visible in the Appendix. Algorithms for the construction of the following datasets, together with the mirroring algorithm are detailed in the Appendix.

In the case of the dataset $\mathcal{D}_{\text{Jenga}}$, we build the n-th base mesh $\widehat{\Omega}_n$ as follows. We start by drawing two horizontal edges that split the domain $(0, 1)^2$ into three horizontal rectangles with area equal to $1/4$, $1/2$ and $1/4$ respectively. Then, we split the rectangle with area $1/2$ vertically, into two equally sized rectangles with area $1/4$. This provides us with base mesh $\widehat{\Omega}_0$, which coincides with mesh Ω_0. At each next refinement step $n \geq 1$, we split the leftmost rectangle in the middle of the base mesh $\widehat{\Omega}_{n-1}$ by adding a new vertical edge, and apply the mirroring technique to obtain Ω_n. This process is shown in the top panels of Fig. 4.

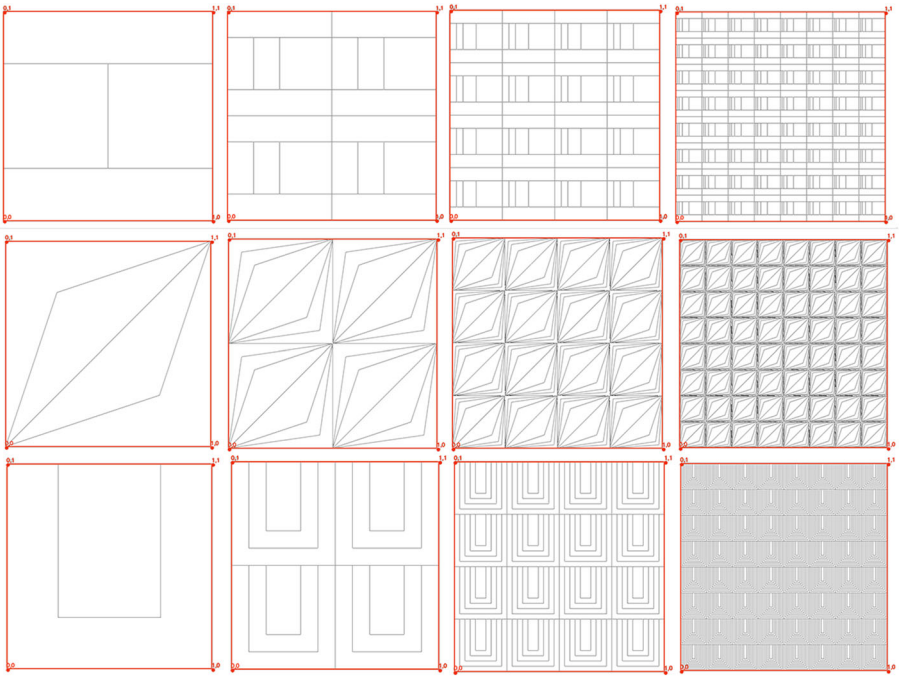

Fig. 4 Meshes Ω_0, Ω_1, Ω_2, Ω_3 from datasets $\mathcal{D}_{\text{Jenga}}$ (top), $\mathcal{D}_{\text{Slices}}$ (middle) and $\mathcal{D}_{\text{Ulike}}$ (bottom)

This mesh family breaks all assumptions **G1** (and **G1-weak**), **G2**, **G3**, and **G4**. In fact, the length of the radius r of the biggest possible disk inscribed into a rectangle is equal to $1/2$ of its shortest edge e. As shown in Fig. 3, the ratio $|e|/h_E$, decreases unboundedly in the left rectangle E every time we split it, and consequently r/h_E decreases at a similar rate. This implies that a lower bound with a uniform constant ρ independent of h cannot exist for these ratios, thus breaking assumptions **G1**, **G1-weak** and **G2**. In addition, the number of edges of the top and bottom rectangular elements also grows unboundedly, against assumption **G3**. Last, the one-dimensional mesh of assumption **G4**, which is built on the elemental boundary of the top and bottom rectangular elements, cannot be subdivided into a finite number of quasi-uniform sub-meshes. In fact, either we have infinite sub-meshes or an infinite edge ratio. Finally, we note that both A_n and e_n scale like 2^n.

In the case of the dataset $\mathcal{D}_{\text{Slices}}$ (Fig. 4, middle), we build the n-th base mesh $\widehat{\Omega}_n$ as follows. First, we sample a collection of points along the diagonal (the one connecting the vertices with coordinates $(0, 1)$ and $(1, 0)$) of the reference square $[0, 1]^2$, and connect them to the vertices $(0, 0)$ and $(1, 1)$. In particular, at each step $n \geq 0$, the base mesh $\widehat{\Omega}_n$ contains the vertices $(0, 0)$ and $(1, 1)$, plus the vertices with coordinates $(2^{-i}, 1 - 2^{-i})$ and $(1 - 2^{-i}, 2^{-i})$ for $i = 1, \ldots, n + 2$. Then, we apply the mirroring technique.

The dataset $\mathcal{D}_{\text{Slices}}$ violates assumptions **G1** and **G1-weak**. In fact, up to a multiplicative scaling factor depending on h, the length of the radius of the biggest inscribed disk in every element E is decreasing faster than the diameter of the element, which is constantly equal to $\sqrt{2}$ times the same scaling factor, thus violating **G1**. Furthermore, the dataset also breaks assumption **G1-weak** because any finite subdivisions of its elements would suffer the same issue. Instead, the other geometrical assumptions are satisfied. Since no edge is split, we find that $e_n \sim c$, while $A_n \sim 2^n$.

In $\mathcal{D}_{\text{Ulike}}$ (Fig. 4, bottom), we build $\widehat{\Omega}_n$ at each step $n \geq 0$ by inserting 2^n equispaced U-shaped continuous polylines inside the domain, creating as many U-like polygons. Then, we apply the mirroring technique.

For arguments similar to the ones brought for $\mathcal{D}_{\text{Maze}}$, $\mathcal{D}_{\text{Ulike}}$ does not satisfy assumptions **G1**, **G1-weak** and **G2**. For connectivity reasons, the lower side of the outer U-shaped polygon of every base mesh must be split into smaller segments when we apply the mirroring technique. Therefore, the number of edges of such cells cannot be limited from above, contradicting assumption **G3**. Nonetheless, assumption **G4** is satisfied because this subdivision is uniform. Last, edge lengths scale exponentially and areas scale uniformly, i.e., $e_n \sim 2^n$, $A_n \sim c$.

Multiple mirroring datasets As a final test, we modified datasets $\mathcal{D}_{\text{Jenga}}$, $\mathcal{D}_{\text{Slices}}$ and $\mathcal{D}_{\text{Ulike}}$ in order to stress the indicators A_n and e_n harder.
This is easily obtained by inserting four new elements at each step instead of one, as explained in the Appendix. The resulting datasets, $\mathcal{D}_{\text{Jenga4}}$, $\mathcal{D}_{\text{Slices4}}$ and $\mathcal{D}_{\text{Ulike4}}$, are qualitatively similar to the mirroring datasets above. These datasets fulfill the same assumptions as their respective original versions, but the number of elements at each refinement step now increases four times faster. The indicators A_n and e_n change

from 2^n to 2^{4n}, but A_n remains constant for $\mathcal{D}_{\text{Ulike4}}$, and e_n remains constant for $\mathcal{D}_{\text{Slices4}}$.

4.2 Performance analysis

We solved the discrete Poisson problem (3) with the VEM (6) described in Section 2 for $k = 1, 2, 3$ over each mesh of each of the datasets defined in Section 4.1, using as groundtruth the function

$$u(x, y) = \frac{\sin(\pi x)\sin(\pi y)}{2\pi^2}, (x, y) \in \Omega = (0, 1)^2. \tag{31}$$

This function has homogeneous Dirichlet boundary conditions, and this choice was appositely made to prevent the boundary treatment from having an influence on the approximation error. In Figs. 5 and 6 we plot the relative L^2-norm and H^1-seminorm as defined in (28), (29):

$$||u - u_h||_{0,\Omega}/||u||_{0,\Omega}, \qquad |u - u_h|_{1,\Omega}/|u|_{1,\Omega},$$

and the relative L^∞-norm

$$||u - u_h||_\infty/||u||_\infty, \text{ where } ||u||_\infty = \text{ess sup}_{x\in\Omega}|u(x)|,$$

of the approximation error $u - u_h$ as the number of DOFs increases (that is, as $n \to N$). We approximate the infinite norm by taking the maximum value of the solution on the degrees of freedom.
The optimal convergence rate of the method, provided by the estimates (26) and (27), is indicated for each k by the slope of the reference triangle. In the case of the L^∞-norm we do not have such theoretical results and we assume the rate of the $\mathcal{D}_{\text{Triangle}}$ dataset as a reference.

We also consider the condition numbers of matrices $\mathbf{G}$ and $\mathbf{H}$ (with the notation adopted in [24]) as numerical indicators of the good behavior of the method, and identities $|\boldsymbol{\Pi}_k^\nabla \mathbf{D} - \mathbf{I}| = 0$ and $|\boldsymbol{\Pi}_k^0 \mathbf{D} - \mathbf{I}| = 0$ as an estimate of the approximation error produced by projectors Π_k^∇ and Π_k^0, represented by matrices $\boldsymbol{\Pi}_k^\nabla$ and $\boldsymbol{\Pi}_k^0$, respectively. The computation of the projectors is obviously affected by the condition numbers of $\mathbf{G}$ and $\mathbf{H}$, but the two indicators are not necessarily related. All of these quantities are computed element-wise and the maximum value among all elements of the mesh is selected. Condition numbers and identity values for $k = 1, 2, 3$ are reported in Table 2 (for $k < 3$ we have $\boldsymbol{\Pi}_k^0 = \boldsymbol{\Pi}_k^\nabla$).

First, the reference dataset $\mathcal{D}_{\text{Triangle}}$ guarantees for the correctness of the VEM, as it performs perfectly according to the theoretical results both in L^2 and in H^1 norms for all k values, maintaining reasonable condition numbers and optimal errors on the projectors Π_k^0 and Π_k^∇.

For the hybrid datasets $\mathcal{D}_{\text{Star}}$ and $\mathcal{D}_{\text{Maze}}$, errors decrease at the correct rate for most of the meshes, and only start deflecting for very high numbers of DOFs and very complicated meshes. These deflections are not due to numerical problems, as

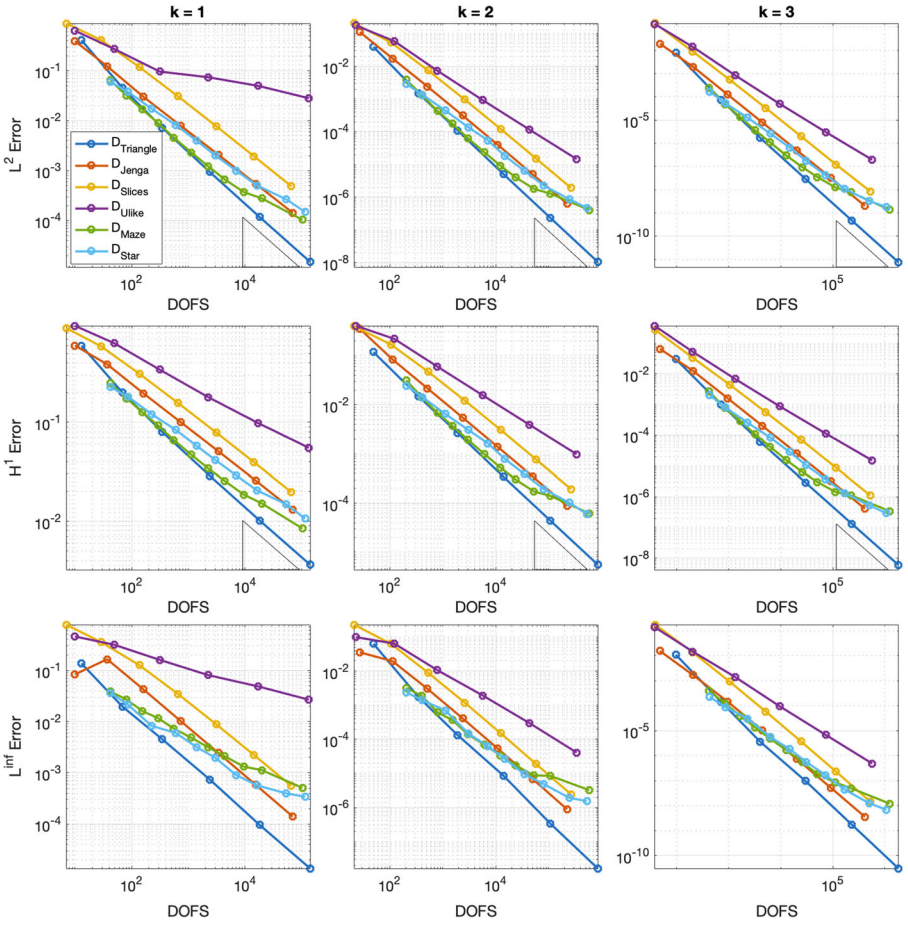

Fig. 5 L^2-norm, H^1-seminorm and L^∞-norm of the approximation errors of the reference, hybrid and mirroring datasets for $k = 1, 2, 3$

in both datasets we have $\text{cond}(\mathbf{G}) < 10^6$ and $\text{cond}(\mathbf{H}) < 10^9$, which are still reasonable values. Projectors seem to work properly: $|\boldsymbol{\Pi}_k^\nabla \mathbf{D} - \mathbf{I}|$ remains below 10^{-8} and $|\boldsymbol{\Pi}_k^0 \mathbf{D} - \mathbf{I}|$ below 10^{-7}. In a preliminary stage of this work, we obtained similar plots (not reported here) using other hybrid datasets built in the same way, with polygons surrounded by triangles. In particular, we did not see big differences when starting with the other initial polygons of Benchmark [3], cf. the construction discussed in "Hybrid datasets" in Section 4.1.

On the meshes from "Mirroring datasets," A_n or e_n may scale non-uniformly, as reported in Table 1 (indeed, they can scale exponentially). This reflects to $\text{cond}(\mathbf{G})$ and $\text{cond}(\mathbf{H})$, which grow up to 10^{10} and 10^{14} for $\mathcal{D}_{\text{Jenga}}$ in the case $k = 3$. Nonetheless, the discrepancy of the projectors identities remains below 10^{-5}, which is not far from what happened with $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$. Dataset $\mathcal{D}_{\text{Jenga}}$ exhibits an almost perfect convergence rate, even though the errors in the different norms are bigger

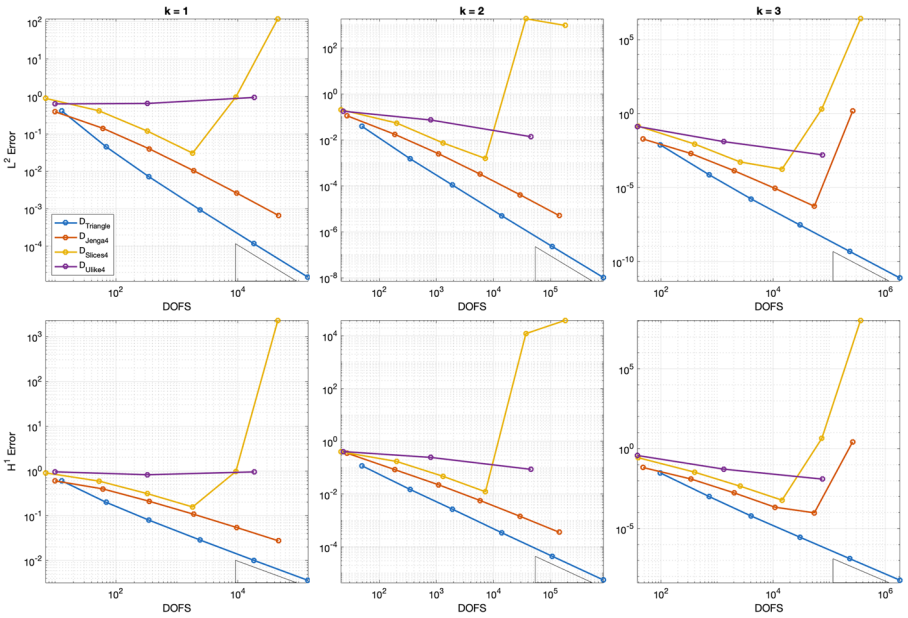

Fig. 6 L^2-norm and H^1-seminorm of the approximation errors of the reference and multiple mirroring datasets for $k = 1, 2, 3$ (the L^∞ error is omitted, being very similar to the other two)

in magnitude than the ones measured for hybrid datasets; $\mathcal{D}_{\text{Slices}}$ shows even bigger errors and a non-optimal convergence rate, and $\mathcal{D}_{\text{Ulike}}$ is the dataset with the poorest performance, but still converges at a decent rate for $k > 1$.

In the setting of "Multiple mirroring datasets," all datasets diverge badly (see Fig. 6), and this is principally due to very poor conditioning in the matrices involved in the calculations (see Table 2). In Fig. 6, for a more compact visualization, we omit the L^∞ error, being it very similar to the other two (as already visible in Fig. 5). Dataset $\mathcal{D}_{\text{Jenga4}}$ and $\mathcal{D}_{\text{Slices4}}$ maintain a similar trend to the ones in Fig. 5 until numerical problems cause cond($\mathbf{G}$) and cond($\mathbf{H}$) to explode up to over 10^{30} for $\mathcal{D}_{\text{Jenga4}}$ and 10^{18} for $\mathcal{D}_{\text{Slices4}}$. In these conditions, projection matrices $\boldsymbol{\Pi}_k^\nabla$ and $\boldsymbol{\Pi}_k^0$ become meaningless and the method diverges. The situation slightly improves for $\mathcal{D}_{\text{Ulike4}}$: cond($\mathbf{H}$) is still 10^{16}, but the discrepancy of $\boldsymbol{\Pi}_k^\nabla$ and $\boldsymbol{\Pi}_k^0$ remain acceptable. As a result, $\mathcal{D}_{\text{Ulike4}}$ does not properly explode, but the approximation error and the convergence rate are much worse than those seen in Fig. 5.

As a preliminary conclusion, by simply looking at the previous plots we observe that the relationship is not particularly strong between the geometrical assumptions respected by a certain dataset and the performance of the VEM. In fact, we obtained reasonable results with meshes violating several assumptions.

Analysis of the stability term In order to investigate the impact of the choice of the stability term on the performance of the method, we compare the results obtained

Table 2 Summary of numerical performance for all datasets. We report the $\log_{10}$ of the original values for the condition number of $\mathbf{G}$ and $\mathbf{H}$ and the discrepancy of projection matrices $\boldsymbol{\Pi}_k^\nabla$ and $\boldsymbol{\Pi}_k^0$. Note that for $k < 3$ we have $\boldsymbol{\Pi}_k^0 = \boldsymbol{\Pi}_k^\nabla$

Dataset	$\mathcal{D}_{\text{Triangle}}$			$\mathcal{D}_{\text{Maze}}$			$\mathcal{D}_{\text{Star}}$			$\mathcal{D}_{\text{Jenga}}$			$\mathcal{D}_{\text{Slices}}$			$\mathcal{D}_{\text{Ulike}}$		
k	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
cond($\mathbf{G}$)	0	2	5	2	3	6	1	3	6	1	5	10	2	4	6	1	4	7
cond($\mathbf{H}$)	2	5	7	2	5	8	3	6	9	4	9	14	2	8	10	3	7	10
$\lvert\boldsymbol{\Pi}_k^\nabla\mathbf{D}-\mathbf{I}\rvert$	−13	−11	−9	−12	−10	−8	−12	−10	−8	−12	−8	−5	−12	−10	−9	−13	−10	−8
$\lvert\boldsymbol{\Pi}_k^0\mathbf{D}-\mathbf{I}\rvert$			−10			−8			−7			−5			−5			−7

Dataset	$\mathcal{D}_{\text{Jenga4}}$			$\mathcal{D}_{\text{Slices4}}$			$\mathcal{D}_{\text{Ulike4}}$		
k	1	2	3	1	2	3	1	2	3
cond($\mathbf{G}$)	6	18	31	6	8	10	2	6	11
cond($\mathbf{H}$)	13	26	39	2	15	18	5	10	16
$\lvert\boldsymbol{\Pi}_k^\nabla\mathbf{D}-\mathbf{I}\rvert$	−9	3	13	−8	−6	−5	−13	−8	−5
$\lvert\boldsymbol{\Pi}_k^0\mathbf{D}-\mathbf{I}\rvert$			20			8			−4

with four different formulations of $S_h^E(\cdot,\cdot)$. The results shown in Figs. 5 and 6 were relative to the D-recipe stabilization (18). In addition to this, we consider the particular case of dofi-dofi stabilization (20), as well as the L^2 and H^1 trace forms (21), (22). In Fig. 7 we exhibit two representative cases relative to $\mathcal{D}_{\text{Slices}}$ and $\mathcal{D}_{\text{Maze}}$ datasets, the others being very similar.

The stability term, at least the considered ones, seem to have a very small impact on the convergence of the method as long as the convergence rate remains optimal. This is the case of $\mathcal{D}_{\text{Slices}}$ (Fig. 7a) and of the first meshes of $\mathcal{D}_{\text{Maze}}$ (Fig. 7b). On the other side, when the method does not work properly, all types of stabilization lead to similar misbehavior, as happens for the last meshes of $\mathcal{D}_{\text{Maze}}$. Analogous results were obtained for all datasets: whenever the method works, all stabilizations lead to equally accurate approximations; when the method does not work, there is not a particular stabilization providing better results than the others. For example, in the case of $\mathcal{D}_{\text{Slices}}$ the L^2 trace stabilization seems to be more problematic than the others, while for $\mathcal{D}_{\text{Maze}}$ the H^1 trace seems the least reliable, and in general the worst stabilization varies from datasets to dataset.

5 Mesh quality indicator

We now aim at defining a mesh quality indicator, that is, a scalar function capable of providing insights on the behavior of the VEM over a particular sequence of meshes, before actually computing the approximated solutions.

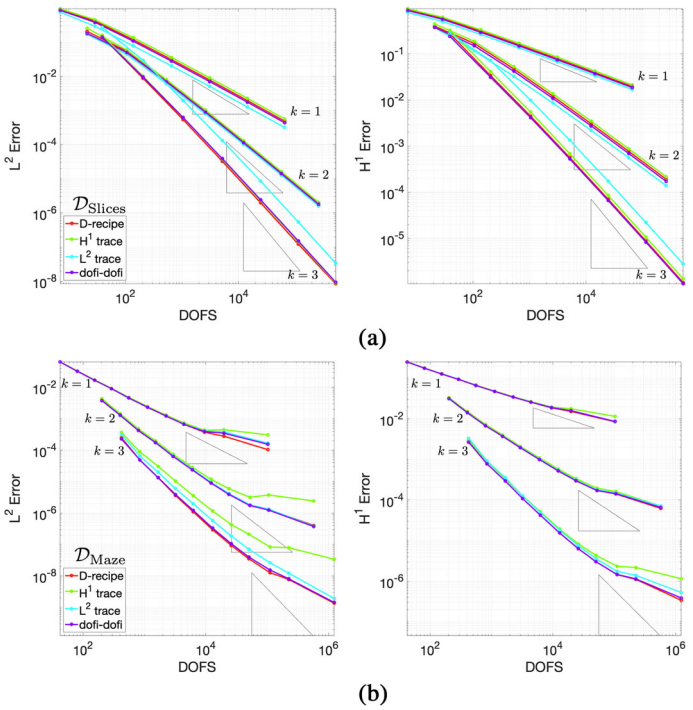

Fig. 7 L^2-norm and H^1-seminorm of the approximation errors relative to $\mathcal{D}_{\text{Slices}}$ (a) and $\mathcal{D}_{\text{Maze}}$ (b) datasets, with different stability terms

5.1 Definition

We start from the geometrical assumptions defined in Section 3. Even if we proved them not to be strictly necessary for the convergence of the method, they can still be good indicators for the general quality of a sequence of meshes. From each geometrical assumption **Gi**, $\mathbf{i} = 1, \dots, 4$, we derived a scalar function $\varrho_i : \{E \subset \Omega_h\} \to [0, 1]$ defined element-wise, which measures how well a polygon $E \in \Omega_h$ meets the requirements of **Gi** from 0 (E does not respect **Gi**) to 1 (E fully respects **Gi**).

$$\varrho_1(E) = \frac{k(E)}{|E|} = \begin{cases} 1 & \text{if } E \text{ is convex} \\ \in (0,1) & \text{if } E \text{ is concave and star-shaped} \\ 0 & \text{if } E \text{ is not star-shaped} \end{cases} \tag{32}$$

$$\varrho_2(E) = \frac{\min(\sqrt{|E|},\ \min_{e \in \partial E} |e|)}{\max(\sqrt{|E|},\ h_E)} \tag{33}$$

$$\varrho_3(E) = \frac{3}{\#\{e \in \partial E\}} = \begin{cases} 1 & \text{if } E \text{ is a triangle} \\ \in (0,1) & \text{otherwise} \end{cases} \tag{34}$$

$$\varrho_4(E) = \min_i \frac{\min_{e \in \mathcal{I}_E^i} |e|}{\max_{e \in \mathcal{I}_E^i} |e|} \tag{35}$$

The operator $k(E)$ in ϱ_1 measures the area of the *kernel* of a polygon E, defined as the set of points in E from which the whole polygon is visible. Therefore, $\varrho_1(E)$ can be interpreted as an estimate of the value of the constant ρ from assumption **G1** on the polygon E. Similarly, the function ϱ_2 returns an estimate of the constant ρ introduced in **G2**, expressed trough the ratio $|e|/h_E$, with the insertion of the quantity $\sqrt{|E|}$ in order to avoid pathological situations. Function ϱ_3 is a simple counter of the number of edges of a polygon, which penalizes elements with numerous edges as required by **G3**. Last, we recall from Section 3 that the boundary of a polygon E can be considered as a one-dimensional mesh $\mathcal{I}_E$, which can be subdivided into a number of disjoint sub-meshes $\mathcal{I}_E^1, \dots, \mathcal{I}_E^N$, each one containing possibly more than one edge of E. In practice, we consider as a sub-mesh the collection of all edges whose vertices lie on the same line. For example, as shown in Fig. 8, the boundary of the top bar E in the base mesh of $\mathcal{D}_{\text{Jenga}}$ is represented by a mesh $\mathcal{I}_E = \{\mathcal{I}_E^1, \mathcal{I}_E^2, \mathcal{I}_E^3, \mathcal{I}_E^4\}$, where $\mathcal{I}_E^1, \mathcal{I}_E^2$ and $\mathcal{I}_E^3$ contain, respectively, the left, top and right edge of E, while $\mathcal{I}_E^4$ contains all the aligned edges in the bottom of E. Function ϱ_4 returns the minimum ratio between the smallest and the largest element in every $\mathcal{I}_E$, that is a measure of the quasi-uniformity of $\mathcal{I}_E$ imposed by **G4**.

Combining together $\varrho_1, \varrho_2, \varrho_3$ and ϱ_4, we define a global function $\varrho : \{\Omega_h\}_h \to [0, 1]$ which measures the overall quality of a mesh Ω_h. Given a dataset $\mathcal{D}$, we can study the behavior of $\varrho(\Omega_h)$ for $\Omega_h \in \mathcal{D}$ and determine the quality of the dataset through the refinement process. In particular, we chose the formula $\varrho_1\varrho_2 + \varrho_1\varrho_3 + \varrho_1\varrho_4$ as it reflects the way in which the relative assumptions are typically imposed: **G1** and **G2**, **G1** and **G3** or **G1** and **G4** (but not, for instance, **G2** and **G3** simultaneously):

$$\varrho(\Omega_h) = \sqrt{\frac{1}{\#\{E \in \Omega_h\}} \sum_{E \in \Omega} \frac{\varrho_1(E)\varrho_2(E) + \varrho_1(E)\varrho_3(E) + \varrho_1(E)\varrho_4(E)}{3}}. \tag{36}$$

We have $\varrho(\Omega_h) = 1$ if and only if Ω_h is made only of equilateral triangles, $\varrho(\Omega_h) = 0$ if and only if Ω_h is made only of non star-shaped polygons, and $0 < \varrho(\Omega_h) < 1$ otherwise. All indicators $\varrho_1, \varrho_2, \varrho_3$ and ϱ_4, and consequently ϱ, only depend on the geometrical properties of the mesh elements; therefore their values can be computed before applying the VEM, or any other numerical scheme.

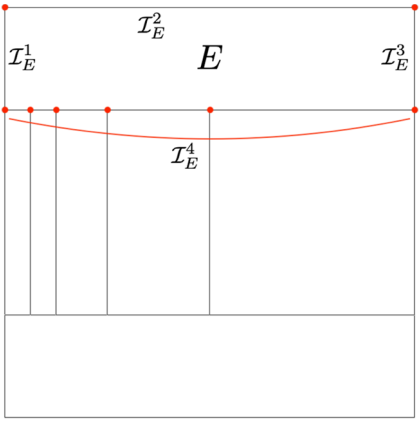

Fig. 8 One-dimensional mesh $\mathcal{I}_E = \{\mathcal{I}_E^1, \mathcal{I}_E^2, \mathcal{I}_E^3, \mathcal{I}_E^4\}$ for the top bar E of a $\mathcal{D}_{\text{Jenga}}$ base mesh

We point out that this approach is easily upgradeable to future developments: whenever new assumptions on the features of a mesh should come up, one simply needs to introduce in our framework a new function ϱ_i that measures the violation of the new assumption and insert it into the formulation of the general indicator ϱ in (36).

5.2 Results

We evaluated the indicator ϱ over the datasets defined for this work; results are shown in Fig. 9.

If we compare Fig. 9a and b with Figs. 5 and 6 respectively, we can look for a correspondence between the behavior of ϱ on a dataset, computed before solving the problem, and the approximation error actually produced by that dataset. Clearly, as ϱ does not depend on the polynomial degree k nor on the type of norm used, we will com pare it to an average of the plots for the different k values and for the different norms.

We preliminarily observe that, for an ideal dataset made by meshes containing only equilateral triangles, ϱ would be constantly equal to 1. We assume this value as a reference for the other datasets: the closer is ϱ on a dataset to the line $y = 1$, the smaller is the approximation error that we expect that dataset to produce. Similarly, the more negative is the ϱ slope, the worse is the convergence rate that we expect over that dataset.

For meshes belonging to $\mathcal{D}_{\text{Triangle}}$, ϱ is almost constant and very close to 1, thus foreseeing the excellent convergence rates and the low errors seen in every sub-figure of Fig. 5. The plots for $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$ in Fig. 9a are close to $\mathcal{D}_{\text{Triangle}}$, hence we expect them to behave similarly. This is confirmed by Fig. 5: $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$ are almost coincident and very close to $\mathcal{D}_{\text{Triangle}}$ until the very last meshes.

The $\mathcal{D}_{\text{Jenga}}$ plot in Fig. 9a anticipates a perfect convergence rate but greater error values with respect to the previous three, and again this behavior is respected in

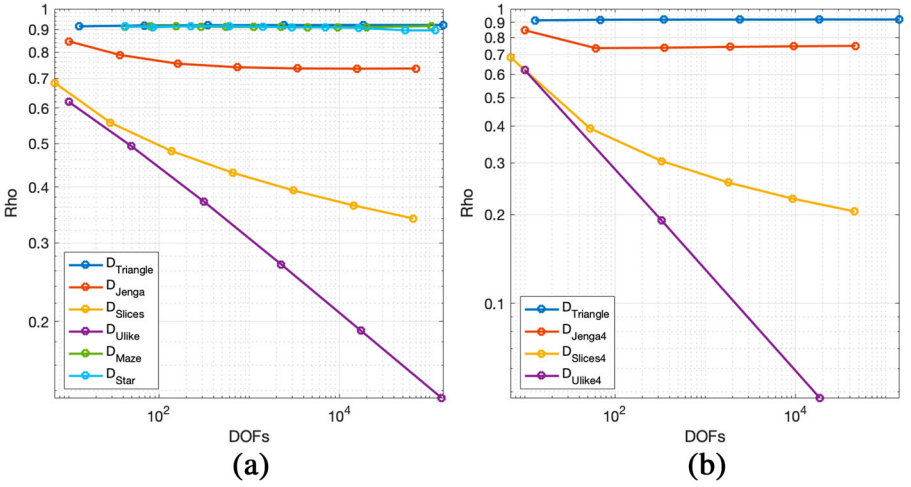

Fig. 9 Indicator ϱ for the reference, hybrid and mirroring datasets (**a**) and for the reference and multiple mirroring datasets (**b**)

Fig. 5. The curve relative to $\mathcal{D}_{\text{Slices}}$ in Fig. 9a is quite distant from the ideal value of 1. Importantly, it keeps decreasing from mesh to mesh, but the plot allows us to assume that it may flatten within a couple more meshes. Looking at Fig. 5, we notice that this dataset produces an error significantly higher than the previous ones ($\mathcal{D}_{\text{Triangle}}$, $\mathcal{D}_{\text{Maze}}$, $\mathcal{D}_{\text{Star}}$, $\mathcal{D}_{\text{Jenga}}$), but the convergence rates are very similar to the theoretical estimates. Last, the ϱ values in Fig. 9 predict huge errors and a completely wrong convergence rate for $\mathcal{D}_{\text{Ulike}}$. This dataset is actually the one with the worst performance in Fig. 5, where it exhibits incorrect convergence rates for all values of k and all types of norm (see in particular the case $k = 1$ with H^1 or L^∞ errors).

As far as multiply refined datasets are concerned, we notice that, since it only depends on the geometry of the elements, ϱ is not affected by numerical errors. The ϱ plot for $\mathcal{D}_{\text{Jenga4}}$ in Fig. 9b is very similar to the plot obtained for $\mathcal{D}_{\text{Jenga}}$ in 9a, therefore we should expect $\mathcal{D}_{\text{Jenga4}}$ in Fig. 6 to perform similarly to $\mathcal{D}_{\text{Jenga}}$ in Fig. 5. This is actually the case at least until the last mesh for $k = 3$, when numerical problems appear which ϱ is not able to predict. Also dataset $\mathcal{D}_{\text{Slices4}}$ has a similar trend to $\mathcal{D}_{\text{Slices}}$ but decreases faster, reaching a ϱ value of ~ 0.2 instead of ~ 0.34 within a smaller number of meshes. As above, $\mathcal{D}_{\text{Slices4}}$ performs similarly to $\mathcal{D}_{\text{Slices}}$ until condition numbers explode, in the last two meshes for every value of k. Last, the ϱ plot of $\mathcal{D}_{\text{Ulike4}}$ is significantly worse than the one of $\mathcal{D}_{\text{Ulike}}$ (and than any other), both in terms of distance from $y = 1$ and slope. In Fig. 6 we can observe how, even if $\mathcal{D}_{\text{Ulike4}}$ does not properly explode (as it suffers less from numerical problems, cf. Table 2), the approximation error and the convergence rate are the worse among all the considered datasets.

Summing up these results, we conclude that indicator ϱ is able, up to a certain accuracy, to predict the behavior of the VEM over the considered datasets, both in terms of error magnitude and convergence rate. The prediction may be inaccurate in presence of very similar performance (the case of $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Star}}$), or in extreme situations in which the numerical problems become so significant to overcome any influence that the geometrical features of the mesh could have on the performance (the last meshes of $\mathcal{D}_{\text{Jenga4}}$ and $\mathcal{D}_{\text{Slices4}}$).

5.3 Error localization

As a further investigation on the quality indicator, we analyze the distribution of the approximation errors across the elements of a mesh. In Fig. 10 we consider two examples from datasets $\mathcal{D}_{\text{Maze}}$ and $\mathcal{D}_{\text{Jenga}}$: we only report the H^1 error because the L^2 and the L^∞ errors produce very similar results.
First, we color each element of the mesh with respect to the value of the elemental quality indicator:

$$\varrho(E) = \sqrt{\frac{\varrho_1(E)\varrho_2(E) + \varrho_1(E)\varrho_3(E) + \varrho_1(E)\varrho_4(E)}{3}}, \quad \forall E \in \Omega_h.$$

Then we visually compare this colored mesh with the same mesh colored with respect to the H^1 error, for $k = 1, 2, 3$, produced by the VEM on it:

$$\epsilon(E) = -\log \frac{|u - u_h|_{1,E}}{|u|_{1,\Omega}}, \quad \forall E \in \Omega_h,$$

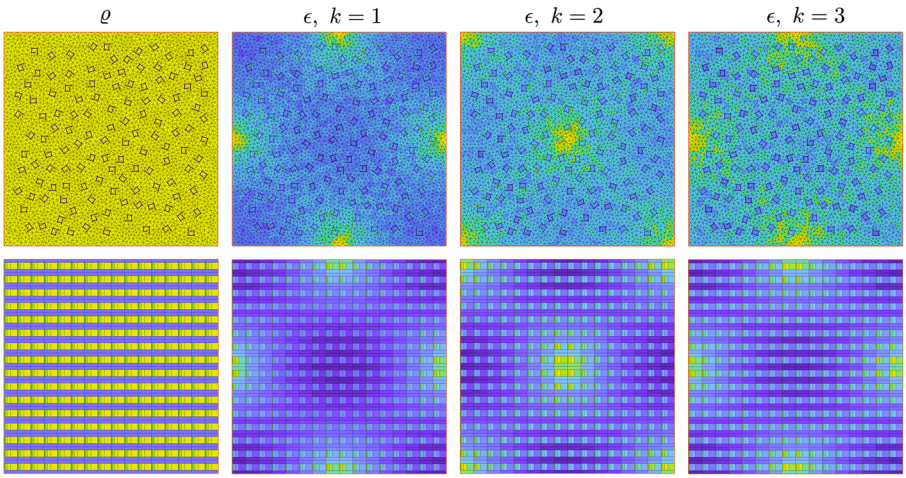

Fig. 10 Localization of the quality indicator ϱ and H^1 error ϵ in meshes from $\mathcal{D}_{\text{Maze}}$ (top) and $\mathcal{D}_{\text{Jenga}}$ (bottom) datasets

where the negative sign is introduced so that high error values correspond in color to low quality elements (remember that $\varrho(E)$ is 1 if E is an equilateral triangle and 0 if E is not star shaped). Moreover, ϵ values are re-scaled in the range $(\min_{E\in\Omega_h}\epsilon(E), \max_{E\in\Omega_h}\epsilon(E))$ in order to highlight differences between the elements. In particular, this means that there is no relationship between a certain color in the figure for $k=1$ and the same color in the case $k=2$ or 3.

In the first column of Fig. 10 we can observe how the quality indicator perfectly "recognizes" the pathological elements, assigning them a deep blue color. Regarding the error, we can see similar color patterns across the columns, which depend on the function we are approximating. Being the groundtruth (31) a sinusoidal function, the error is naturally distributed along "waves" which vary with the order of the method. Besides this, it is still appreciable how poor quality elements produce higher errors than their neighbors, highlighting once again a correlation between the quality indicator and the performance of the VEM.

6 Conclusions

In this work, we collected the regularity assumptions that are used in the literature to guarantee the convergence and the error estimates in the L^2 and H^1 norms for the VEM. These conditions allow a great flexibility for the type and variety of polygons to be used in a mesh, but they still seem overestimated. Experimentally, we verified that the VEM works, with a good convergence rate, also on meshes and datasets that strongly violate these assumptions. We also built examples of datasets for which, violating significantly the regularity assumptions, the VEM shows a convergence rate suboptimal or diverges. Finally, we introduced new indicators to represent how much the regularity hypothesis are violated by a tessellation and combined these indicators

in a single score, aimed at estimating how a dataset can be expected to be performing in the solution of the VEM. The results obtained are encouraging, showing a satisfactory correlation between the errors and this indicator. Consequently, our approach provides an experimental score that is able to predict if a tessellation of a domain can be critical for the VEM.

As possible future developments, we are interested in refining the regularity indicator here proposed, for example, to deduce new decomposition rules of a domain with possible applications to mesh generators, or to adaptive coarsening/refinement algorithms. We are also experimenting similar indicators to evaluate the properties of polyhedral meshes, taking advantage of the 3D extension of the geometric kernel computation presented in [21].

Appendix: Dataset generation

In this appendix, we take a closer look at how the datasets presented in Section 4.1 are built and how to compute the quantities A_n and e_n defined in (30). All algorithms have been written using CinoLib [15]. We recall that a dataset is a finite mesh sequence $\mathcal{D} = \{\Omega_n\}_{n=0,\dots,N}$, ordered decreasingly with respect to the mesh size.

Reference dataset The first dataset, $\mathcal{D}_{\text{Triangle}}$, contains only triangular meshes that are built by inserting a number of points in the domain, and connecting them in a Delaunay triangulation. The point set is defined through the *Poisson Disk Sampling* algorithm proposed in [9], empirically adjusting the distance between points (called *radius* in the original paper) in order to generate meshes with the desired number of vertices. Points are then connected in a Delaunay triangulation using the well known *Triangle library* [20], with the default parameters configuration. In $\mathcal{D}_{\text{Triangle}}$, A_n and e_n are almost constant, as no constraints are imposed to the triangulation process.

Hybrid datasets The construction of hybrid datasets is characterized by the insertion in Ω of one or more polygonal elements, and by a tessellation algorithm. Each hybrid dataset is built around (and named after) an *initial polygon* $E = E(t_n)$ depending on a *deformation parameter* $t_n \in [0, 1)$, which is used to deform E. This parameter directly depends on the mesh number (i.e., $t_n \to 1$ as $n \to N$), and it can be adjusted to improve or worsen the quality of the polygon E (the higher, the worse). At refinement step n, mesh Ω_n is created by inserting a number of identical copies of the deformed polygon $E(t_n)$ (opportunely resized) in the domain Ω, and tessellating the rest of Ω using the Triangle library. Note that, a whole family of other datasets may be generated by simply defining a new initial polygon. More examples can be found in [3].

The initial polygon $E(t_n)$ for dataset $\mathcal{D}_{\text{Maze}}$ is the 10-sided element shown in Fig. 11a, with vertices (0,1), (0,0), (1,0), (1,0.75), (0.5,0.75), $\left(0.5, 0.5 + \frac{t_n}{4}\right)$, $\left(0.75 + \frac{t_n}{4}, 0.5 + \frac{t_n}{4}\right)$, $\left(0.75 + \frac{t_n}{4}, 0.25 - \frac{t_n}{4}\right)$, $\left(0.25 - \frac{t_n}{4}, 0.25 - \frac{t_n}{4}\right)$, $\left(0.25 - \frac{t_n}{4}, 1\right)$. As $t_n \to 1$, the length of the shortest edge (the one with vertices $(0, 1)$ and $(0.25 - t_n/4, 1)$) goes to zero, and so does the area of $E(t_n)$.

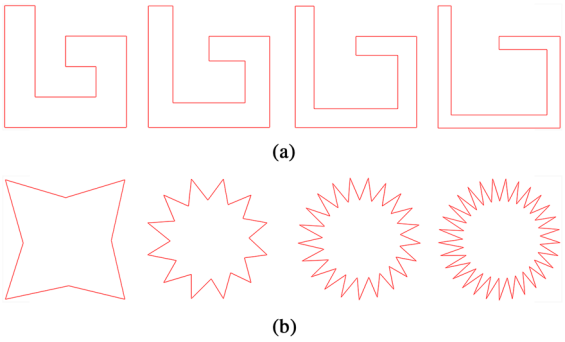

Fig. 11 Initial polygons $E(t_0)$, $E(t_2)$, $E(t_4)$, $E(t_6)$ from datasets $\mathcal{D}_{\text{Maze}}$ (a) and $\mathcal{D}_{\text{Star}}$ (b)

For building the initial polygon $E(t_n)$ of dataset $\mathcal{D}_{\text{Star}}$ (Fig. 11b), we first build a $\bar{i}$-sided regular polygon, with $\bar{i} = 8(1+\lfloor 10t_n \rfloor)$ and vertices $v_0 = (1,0)$, $v_i = \sigma(v_{i-1})$ for $i = 1, \ldots, \bar{i}$, being $\sigma(v)$ the rotation centered at $(0,0)$ of vertex v by an angle of $2\pi/\bar{i}$. Then we project every odd-indexed vertex towards the barycenter of $E(t_n)$: $v'_{2j+1} = s\, v_{2j+1}$, for $j = 0, \ldots, \frac{\bar{i}-1}{2}$, where the projection factor $s \in (0,1)$ is gradually decreased until the angles at the even-indexed vertices become smaller than $(1-t_n)\pi/3$. As $t_n \to 1$ we have an increasing number of edges (from 8 to almost 90), the minimum angle and the area decrease to zero while the length of every edge increases.

Algorithm 1 Hybrid datasets generation.

define the initial polygon
set the initial parameters N, d_0, and T
for $n = 0, \ldots, N$ **do**
    set the main parameters: $e_n := 2^n$, $t_n := n\frac{t_{\max}-t_{\min}}{N}$, $d_n := d_0/2^n$
    use *Poisson Disk Sampling* with $r = 1/\sqrt{2e_n}$ to find e_n points $\{p_i\}_{i=1,\ldots,e_n}$ in Ω
    generate polygon $E(t_n)$ with $|E(t_n)| \leq d_n$
    insert a copy of $E(t_n)$ centered around every p_i
    while polygon $E(t_n)$ intersects with other polygons or with the boundary of Ω **do**
        $d_n := d_n - \epsilon$
        generate a polygon $E(t_n)$ with $|E(t_n)| \leq d_n$
        insert a copy of $E(t_n)$ centered around every p_i
    end while
    generate the Delaunay triangulation Ω_n of Ω, considering polygons $E(t_n)$ as holes
    $\mathcal{D} \leftarrow \Omega_n$
end for

Once we defined the initial polygon $E(t_n)$, we can build the corresponding dataset through Algorithm 1. We have some initial parameters, which are set a priori and remain untouched: the number of meshes in the dataset N, the area of the initial polygon at the first step d_0 and the deformation range $T = [t_{\min}, t_{\max}]$. In this work

we set $N = 10$, $d_0 = 0.03$, which corresponds to 3% of the domain, and $T = [0, 0.95]$.

Then we have three main parameters, $e_n \in \mathbb{N}$, $t_n \in T$ and $d_n \in (0, d_0)$, which respectively regulate the number of initial polygons inserted, the deformation of these polygons and their area. In particular, e_n increases inversely to d_n (Ω_{n+1} has twice as polygons as Ω_n, with halved areas), so that the percentage of the domain covered by polygons (not triangles) is preserved all across the dataset. Due to the complicated shapes of some initial polygons, it may be hard to ask for exactly $|E(t_n)| = d_n$, therefore we only impose $|E(t_n)| \leq d_n$.

Several options are possible for setting e_n, t_n and d_n, and the speeds at which these quantities vary, strongly affect the geometrical qualities of the meshes in the dataset. In our datasets, e_n increases exponentially, t_n increases linearly inside T and d_n decreases exponentially. The exponential increase of the number of initial polygons inserted in the domain may lead to intersections between them, or with the domain boundaries. To avoid this phenomenon, we inserted a *while* loop in Algorithm 1 which decreases d_n until no intersections occur: this ensures stability to the algorithm, but in practice it activates only for very dense meshes and it typically runs only few iterations.

Last, when all polygons have been inserted in Ω, the Triangle algorithm is used to generate a Delaunay triangulation. The already inserted polygons are considered as holes in the domain, and we set no limitations on the number of Steiner points that may appear in the triangulation process. We also impose to have no angles smaller than 20 degrees and set a maximum triangle area constraint equal to d_n. Due to the freedom left to the Triangle algorithm, it is not possible to estimate A_n and e_n precisely; hence, the relative values reported in Table 1 have been measured a posteriori.

Mirroring datasets The construction of $\mathcal{D}_{\text{Jenga}}$, $\mathcal{D}_{\text{Slices}}$ and $\mathcal{D}_{\text{Ulike}}$, at every step $n \geq 1$, consists in a first algorithm for iteratively generating a *base mesh* $\widehat{\Omega}_n$ from the previous base mesh $\widehat{\Omega}_{n-1}$, followed by a *mirroring technique* which returns the computational mesh Ω_n. The base mesh generation algorithm is different for each dataset (Algorithms 2, 3 and 4), while the mirroring algorithm (Algorithm 5) is common to all three datasets. Algorithms 2, 3 and 4 depend on two initial parameters: N indicates the number of meshes in the dataset and N_{el} indicates the number of elements to insert at each step. For mirroring datasets we set $N_{el} = 1$, while for multiple mirroring datasets (described in the next section) we will set $N_{el} = 4$.

In the $\mathcal{D}_{\text{Jenga}}$ base mesh shown in Fig. 12 (top) we have a *top bar*, a *bottom bar* and a *right square* which are fixed independently of n, and $n + 1$ *rectangles* in the left part of the domain. At each refinement step $n \geq 1$, a new rectangular element is created by splitting in two equal parts the leftmost rectangular element in the previous base mesh, and consequently updating the top and the bottom bars with new vertices and edges. Therefore, all elements in $\widehat{\Omega}_n$, except for the top and the bottom bars, are rectangles with height equal to $1/2$ and basis ranging from $1/2$ to $1/2^{n+1}$. Once that the base mesh $\widehat{\Omega}_n$ is generated, the mirroring algorithm is recursively applied for n times to generate the computational mesh Ω_n, as described in Algorithm 2. When computing A_n and e_n, we can restrict our calculations to the base mesh, because these

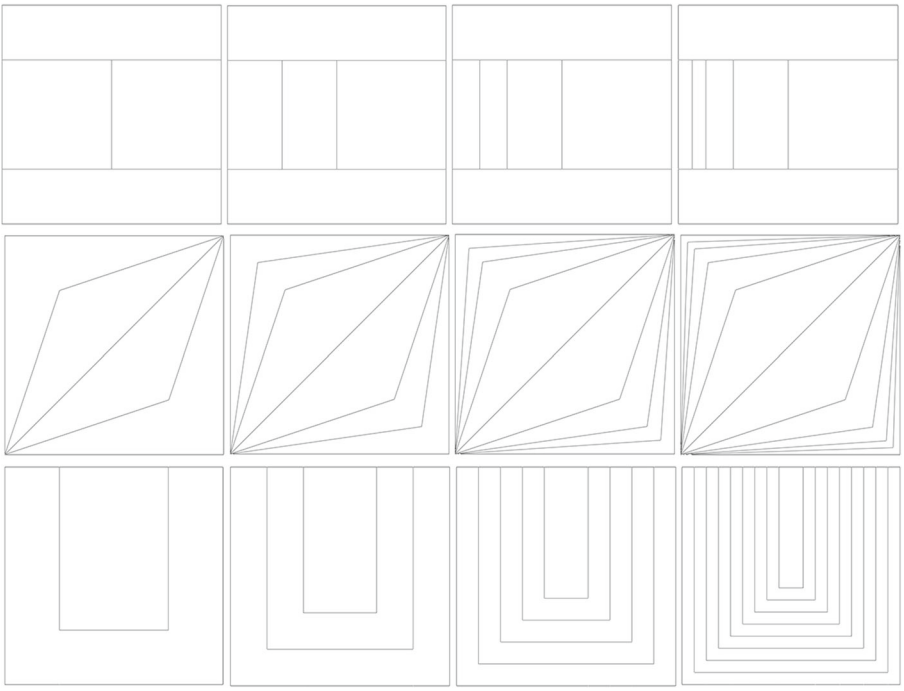

Fig. 12 Non-mirrored base meshes $\widehat{\Omega}_0, \widehat{\Omega}_1, \widehat{\Omega}_2, \widehat{\Omega}_3$ from datasets $\mathcal{D}_{\text{Jenga}}$ (top), $\mathcal{D}_{\text{Slices}}$ (middle) and $\mathcal{D}_{\text{Ulike}}$ (bottom)

ratios are not affected by the mirroring algorithm. In particular, the longest edge in the base mesh is the upper edge of the top bar, which is never split, while the shortest edge is the basis of the leftmost rectangle, which halves at each step: this causes $e_n \sim 2^n$. The top bar is also the element with the greatest area (together with the bottom bar and the right square), which is constantly equal to 1/4, while the leftmost rectangle has area $1/2 * 1/2^{n+1} = 1/2^{n+2}$, therefore $A_n \sim 2^n$.

In the $\mathcal{D}_{\text{Slices}}$ base meshes shown in Fig. 12 (middle), at each step $n \geq 0$, we add the vertices with coordinates $(2^{-i}, 1 - 2^{-i})$ and $(1 - 2^{-i}, 2^{-i})$ for $i = 1, \ldots, n + 2$, and we connect them to the vertices $(0, 0)$ and $(1, 1)$. As a result, at each iteration we create a couple of new polygons, called *upper slice* and *lower slice*, symmetrical with respect to the diagonal, and we add them to the base mesh. The area of the two inner triangles (the biggest polygons in the base mesh) is constantly equal to $1/4$. For evaluating the area of the two most external polygons, we consider them as the union of the two identical triangles obtained by splitting the polygons along the diagonal (the one connecting the vertices with coordinates $(0, 1)$ and $(1, 0)$). Then the smallest area in the base mesh is the sum of the areas of two equal triangles with basis $\sqrt{2}/2$ and height $2^{-n}/\sqrt{2}$, and simple calculations lead to $A_n \sim 2^n$. Last, we notice that all the edges in the base mesh have lengths between 1 and $\sqrt{2}$, because no edge is ever split, hence $e_n \sim c$.

In the $\mathcal{D}_{\text{Ulike}}$ base meshes shown in Fig. 12 (bottom), at each step $n \geq 0$ we insert 2^n U-shaped continuous polylines inside the domain. We have an *internal rectangle* and a sequence of concentric equispaced *U-like polygons* culminating with the *external U*. This last element is not different from the other U-like polygons, but is created separately, because we need to split its lower edge in order to match the base mesh that will appear below it during the mirroring algorithm. In every base mesh, the shortest edge is the one corresponding to the width of each U-like polygon, which measures $2^{-(n+1)}$, and the longest edges are the left and right boundaries of the domain. This causes $e_n \sim 2^n$. Said e the shortest edge, the smallest area is the one of the internal rectangle, equal to $2e(1/2 + e)$, and the biggest area is the one relative to the external U, equal to $3e - 2e^2$. We have

$$A_n = \frac{3 - 2e}{1 + 2e} = \frac{3 - 2(2^{-(n+1)})}{1 + 2(2^{-(n+1)})} = \frac{3 - 2^{-n}}{1 + 2^{-n}} \sim c.$$

Algorithm 2 $\mathcal{D}_{\text{Jenga}}$ dataset generation.

set the number of meshes N and the number of elements N_{el}
for $n = 0, \ldots, N$ **do**
  top bar := {(0, 0.75), (1, 0.75), (1, 1), (0, 1)}
  bottom bar := {(0, 0), (1, 0), (1, 0.25), (0, 0.25)}
  right square := {(0.5, 0.25), (1, 0.25), (1, 0.75), (0.5, 0.75)}
  vector $\boldsymbol{b}$:= sample $n * N_{el}$ equally spaced points inside interval (0, 0.5)
  for $i = 1, \ldots,$ size($\boldsymbol{b}$) **do**
    rectangles[i]:={($\boldsymbol{b}[i-1]$, 0.25), ($\boldsymbol{b}[i]$, 0.25), ($\boldsymbol{b}[i]$, 0.75), ($\boldsymbol{b}[i-1]$, 0.75)}
    top bar $\leftarrow$ ($\boldsymbol{b}[i]$, 0.75)
    bottom bar $\leftarrow$ ($\boldsymbol{b}[i]$, 0.25)
  end for
  generate mesh $\widehat{\Omega}_n$:= {*top bar, bottom bar, right square, rectangles*}
  for $i = 1, \ldots, n$ **do** $\widehat{\Omega}_n$:= mirror mesh($\widehat{\Omega}_n$)
  end for
  $\mathcal{D}_{\text{Jenga}} \leftarrow \widehat{\Omega}_n$
end for

Algorithm 3 $\mathcal{D}_{\text{Slices}}$ dataset generation.

set the number of meshes N and the number of elements N_{el}
for $n = 0, \ldots, N$ **do**
  vector $\boldsymbol{b}$:= $[2^{-1}, 2^{-2}, \ldots, 2^{-n*N_{el}}]$
  for $i = 1, \ldots,$ size($\boldsymbol{b}$) **do**
    upper slices[i] := {(0, 0), ($\boldsymbol{b}[i]$, $1 - \boldsymbol{b}[i]$), (1, 1), ($\boldsymbol{b}[i+1]$, $1 - \boldsymbol{b}[i+1]$)}
    lower slices[i] := {(0, 0), ($1 - \boldsymbol{b}[i]$, $\boldsymbol{b}[i]$), (1, 1), ($1 - \boldsymbol{b}[i+1]$, $\boldsymbol{b}[i+1]$)}
  end for
  generate mesh $\widehat{\Omega}_n$:= {*upper slices, lower slices*}
  for $i = 1, \ldots, n$ **do** $\widehat{\Omega}_n$:= mirror mesh($\widehat{\Omega}_n$)
  end for
  $\mathcal{D}_{\text{Slices}} \leftarrow \widehat{\Omega}_n$
end for

Algorithm 4 $\mathcal{D}_{\text{Ulike}}$ dataset generation.

set the number of meshes N and the number of elements N_{el}
for $n = 0, \ldots, N$ **do**
  vector $\boldsymbol{b}$:= sample $2^{n*N_{el}}$ equally spaced points inside interval $(0, 0.5)$
  for $i = 1, \ldots,$ size($\boldsymbol{b}$) **do**
    U-like polygons$[i] := \{(\boldsymbol{b}[i], 1), (\boldsymbol{b}[i], \boldsymbol{b}[i]), (1-\boldsymbol{b}[i], \boldsymbol{b}[i]), (1-\boldsymbol{b}[i], 1), (1-\boldsymbol{b}[i+1], 1), (1-\boldsymbol{b}[i+1], \boldsymbol{b}[i+1]), (\boldsymbol{b}[i+1], \boldsymbol{b}[i+1]), (\boldsymbol{b}[i+1], 1)\}$
  end for
  internal rectangle := $\{(\boldsymbol{b}[end], 1), (\boldsymbol{b}[end], \boldsymbol{b}[end]), (1 - \boldsymbol{b}[end], \boldsymbol{b}[end]), (1 - \boldsymbol{b}[end], 1)\}$
  external U := $\{(0, 1), (0, 0), (1, 0), (1, 1), (1 - \boldsymbol{b}[0], 1), (1 - \boldsymbol{b}[0], \boldsymbol{b}[0]), (\boldsymbol{b}[0], \boldsymbol{b}[0]), (\boldsymbol{b}[0], 1)\}$
  for $b \in \boldsymbol{b}$ **do**
    external U $\leftarrow \{(b, 0), (1 - b, 0)\}$
  end for
  generate mesh $\widehat{\Omega}_n$:= {*external U, U-like polygons, internal rectangle*}
  for $i = 1, \ldots, n$ **do** $\widehat{\Omega}_n$ = mirror mesh($\widehat{\Omega}_n$)
  end for
  $\mathcal{D}_{\text{Ulike}} \leftarrow \widehat{\Omega}_n$
end for

Multiple mirroring datasets Multiple mirroring datasets are built with the exactly same algorithms of the mirroring datasets, changing the parameter N_{el}. This parameter regulates the number of elements generated in each base mesh of the dataset. In particular, datasets $\mathcal{D}_{\text{Jenga4}}$, $\mathcal{D}_{\text{Slices4}}$ and $\mathcal{D}_{\text{Ulike4}}$ are defined setting $N_{el} = 4$. An example of a multiple mirroring dataset with $N_{el} = 4$ is shown in Fig. 13, where the first three base meshes of $\mathcal{D}_{\text{Ulike4}}$ are presented. The N_{el} value influences ratios A_n and e_n: if $A_n, e_n \sim 2^n$ for $N_{el} = 1$, these quantities become asymptotic to 2^{4n} when $N_{el} = 4$, except for the cases in which the ratios were constant (see Table 1).

The mirroring algorithm The *mirroring* algorithm (Algorithm 5) generates four adjacent copies of any polygonal mesh $\mathcal{M}$ defined over the domain $\Omega = [0, 1]^2$. In CinoLib [15], a polygonal mesh can be defined by a vector *verts* containing all its vertices and a vector *polys* containing all its polygons. The result of the algorithm is therefore a polygonal mesh $\mathcal{M}'$, generated by some vectors *new-verts* and *new-polys*, containing four times the number of vertices and polygons of $\mathcal{M}$. When

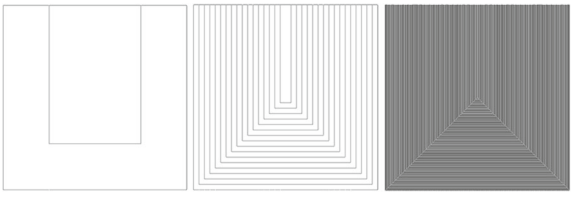

Fig. 13 Non-mirrored base meshes $\widehat{\Omega}_0$, $\widehat{\Omega}_1$ and $\widehat{\Omega}_2$ from datasets $\mathcal{D}_{\text{Ulike4}}$

iterated a sufficient number of times, this construction allows us to obtain a number of vertices and degrees of freedom in each mesh of the mirroring datasets that is comparable to that of the meshes at the same refinement level in hybrid datasets. Vector *new-verts* contains all vertices $v \in verts$ copied four times and translated by vectors $(0, 0)$, $(1, 0)$, $(1, 1)$ and $(0, 1)$ respectively. The coordinates of all vertices in *new vertices* are divided by 2, so that all new points lie in the same domain as the input mesh. Vector *new-polys* is simply vector *polys* repeated four times. A final cleaning step is required to remove duplicated vertices and edges that may arise in the mirroring process, for example if the initial mesh $\mathcal{M}$ has vertices along its boundary.

Algorithm 5 Mesh mirroring.

input: base mesh $\mathcal{M}$
$verts := \text{verts}(\mathcal{M})$, $polys := \text{polys}(\mathcal{M})$
$new\text{-}verts := verts$
for vertex $v \in verts$ **do** $new\text{-}verts \leftarrow v + (1, 0)$
end for
for vertex $v \in verts$ **do** $new\text{-}verts \leftarrow v + (1, 1)$
end for
for vertex $v \in verts$ **do** $new\text{-}verts \leftarrow v + (0, 1)$
end for
for vertex $v \in new\text{-}verts$ **do** $v := v/2$
end for
$new\text{-}polys := [polys, polys, polys, polys]$
$\mathcal{M}' := \text{mesh}\{new\text{-}verts, new\text{-}polys\}$
remove duplicated vertices and edges from $\mathcal{M}'$

Acknowledgements We are very grateful to Dr. L. Mascotto, University of Vienna, for useful suggestions. We also thank Drs. D. Cabiddu, M. Livesu, G. Patanè and D. Prada for their valuable technical and theoretical support.

Funding This paper has been realized in the framework of ERC Project CHANGE, which has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement no. 694515).

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Adams, R.A., Fournier, J.J.F. Sobolev Spaces, 2nd edn. Academic Press, Cambridge (2003). Pure and Applied Mathematics
2. Ahmad, B., Alsaedi, A., Brezzi, F., Marini, L.D., Russo, A.: Equivalent projectors for virtual element methods. Comput. Math. Appl. **66**, 376–391 (2013)

3. Attene, M., Biasotti, S., Bertoluzza, S., Cabiddu, D., Livesu, M., Patanè, G., Pennacchio, M., Prada, D., Spagnuolo, M.: Benchmarking the geometrical robustness of a virtual element poisson solver. Math. Comput. Simul. **190**, 1392–1414 (2021). https://doi.org/10.1016/j.matcom.2021.07.018
4. Benedetto, M.F., Berrone, S., Pieraccini, S., Scialò, S.: The virtual element method for discrete fracture network simulations. Comput. Methods Appl. Mech. Eng. **280**(0), 135–156 (2014)
5. Bertoluzza, S., Manzini, G., Pennacchio, M., Prada, D.: Stabilization of the nonconforming virtual element method (2021)
6. Brenner, S.C., Guan, Q., Sung, L.Y.: Some estimates for virtual element methods. Comput. Methods Appl. Math. **17**(4), 553–574 (2017)
7. Brenner, S.C., Sung, L.Y.: Virtual element methods on meshes with small edges or faces. Math. Models Methods Appl. Sci. **28**(07), 1291–1336 (2018)
8. Brezzi, F., Buffa, A., Lipnikov, K.: Mimetic finite differences for elliptic problems. M2AN. Math. Model. Numer. Anal. **43**, 277–295 (2009)
9. Bridson, R.: Fast Poisson disk sampling in arbitrary dimensions. SIGGRAPH sketches **10**, 1 (2007)
10. Cangiani, A., Gyya, V., Manzini, G., Sutton, O.J.: Chapter 14: Virtual element methods for elliptic problems on polygonal meshes. In: Hormann, K., Sukumar, N. (eds.) Generalized Barycentric Coordinates in Computer Graphics and Computational Mechanics, pp. 1–20. CRC Press, Taylor & Francis Group, Boca Raton (2017)
11. Cao, S., Chen, L.: Anisotropic error estimates of the linear virtual element method on polygonal meshes. SIAM J. Numer. Anal. **56**(5), 2913–2939 (2018)
12. Ciarlet, P.G.: The Finite Element Method for Elliptic Problems. SIAM, Philadelphia (2002)
13. Di Pietro, D.A., Droniou, J., Manzini, G.: Discontinuous skeletal gradient discretisation methods on polytopal meshes. J. Comput. Phys. **355**, 397–425 (2018). https://doi.org/10.1016/j.jcp.2017.11.018
14. Lipnikov, K., Manzini, G., Shashkov, M.: Mimetic finite difference method. J. Comput. Phys. **257**, 1163–1227 (2014). Review paper
15. Livesu, M.: cinolib: a generic programming header only C++ library for processing polygonal and polyhedral meshes. In: Transactions on Computational Science XXXIV, pp. 64–76. Springer (2019)
16. Manzini, G., Lipnikov, K., Moulton, J.D., Shashkov, M.: Convergence analysis of the mimetic finite difference method for elliptic problems with staggered discretizations of diffusion coefficients. SIAM J. Numer. Anal. **55**(6), 2956–2981 (2017). https://doi.org/10.1137/16M1108479
17. Manzini, G., Russo, A., Sukumar, N.: New perspectives on polygonal and polyhedral finite element methods. Math. Models Methods Appl. Sci. **24**(8), 1621–1663 (2014)
18. Mascotto, L.: Ill-conditioning in the virtual element method: stabilizations and bases. Numer. Methods Partial Diff. Equ. **34**(4), 1258–1281 (2018)
19. Scott, L.R., Brenner, S.C. The Mathematical Theory of Finite Element Methods, 3rd edn. Springer, New York (2008). http://gen.lib.rus.ec/book/index.php?md5=9ad140b0da07318a38c12d35e5616cc2
20. Shewchuk, J.R.: Triangle library. https://www.cs.cmu.edu/quake/triangle.html (2005)
21. Sorgente, T., Biasotti, S., Spagnuolo, M.: A geometric approach for computing the kernel of a polyhedron. In: Frosini, P., Giorgi, D., Melzi, S., Rodolà, E. (eds.) Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference. The Eurographics Association (2021). https://doi.org/10.2312/stag.20211470
22. Sorgente, T., Prada, D., Cabiddu, D., Biasotti, S., Patane, G., Pennacchio, M., Bertoluzza, S., Manzini, G., Spagnuolo, M.: Vem and the mesh (2021)
23. Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L.D., Russo, A.: Basic principles of virtual element methods. Math. Models Methods Appl. Sci. **23**, 119–214 (2013)
24. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: The Hitchhiker's guide to the virtual element method. Math. Models Methods Appl. Sci. **24**(8), 1541–1573 (2014)
25. Beirão da Veiga, L., Brezzi, F., Marini, L.D., Russo, A.: Virtual element methods for general second order elliptic problems on polygonal meshes. Math. Models Methods Appl. Sci. **26**(4), 729–750 (2016)
26. Beirão da Veiga, L., Lipnikov, K., Manzini, G.: Arbitrary order nodal mimetic discretizations of elliptic problems on polygonal meshes. SIAM J. Numer. Anal. **49**(5), 1737–1760 (2011)
27. Beirão da Veiga, L., Lipnikov, K., Manzini, G. The Mimetic Finite Difference Method, MS&A. Modeling, Simulations and Applications, 1st edn., vol. 11. Springer, Berlin (2014)
28. Beirão da Veiga, L., Lovadina, C., Russo, A.: Stability analysis for the virtual element method. Math. Models Methods Appl. Sci. **27**(13), 2557–2594 (2017)

29. Beirão da Veiga, L., Manzini, G.1.: A virtual element method with arbitrary regularity. IMA J Numer Anal **34**(2), 782–799 (2014). https://doi.org/10.1093/imanum/drt018. (first published online 2013)
30. Beirão da Veiga, L., Manzini, G.: Residual a posteriori error estimation for the virtual element method for elliptic problems. ESAIM Math Model Numer Anal **49**, 577–599 (2015). https://doi.org/10.1051/m2an/2014047
31. Beirão da Veiga, L., Manzini, G., Putti, M.: Post-processing of solution and flux for the nodal mimetic finite difference method. Numer. Methods Partial Differ. Equ. **31**(1), 336–363 (2015)
32. Beirão da Veiga, L., Vacca, G.: Sharper error estimates for virtual elements and a bubble-enriched version. arXiv:2005.12009 (2020)
33. Wriggers, P., Rust, W.T., Reddy, B.D.: A virtual element method for contact. Comput. Mech. **58**(6), 1039–1050 (2016). https://doi.org/10.1007/s00466-016-1331-x
34. Zlámal, M.: On the finite element method. Numer. Math. **12**(5), 394–409 (1968)