POLITECNICO DI TORINO
Repository ISTITUZIONALE

Effective polygonal mesh generation and refinement for VEM

Original
Effective polygonal mesh generation and refinement for VEM / Berrone, S., Vicini, F.. - In: MATHEMATICS AND COMPUTERS IN SIMULATION. - ISSN 0378-4754. - ELETTRONICO. - 231:(2025), pp. 239-258. [10.1016/j.matcom.2024.12.007]

Availability:
This version is available at: 11583/3004758 since: 2025-11-03T13:05:54Z

Publisher:
Elsevier

Published
DOI:10.1016/j.matcom.2024.12.007

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

Mathematics and Computers in Simulation 231 (2025) 239–258

Contents lists available at ScienceDirect

Mathematics and Computers in Simulation

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

Original articles

Effective polygonal mesh generation and refinement for VEM

Stefano Berrone[1], Fabio Vicini[1,*]

Dipartimento di Scienze Matematiche "Giuseppe Luigi Lagrange", Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy

ARTICLE INFO

Keywords:
Mesh adaptivity
Virtual Element Method
Polygonal mesh refinement
Convergence and optimality

ABSTRACT

In the present work we introduce a novel refinement algorithm for two-dimensional elliptic partial differential equations discretized with Virtual Element Method (VEM). The algorithm improves the numerical solution accuracy and the mesh quality through a controlled refinement strategy applied to the generic polygonal elements of the domain tessellation. The numerical results show that the outlined strategy proves to be versatile and possibly applicable to each two-dimensional problem where polygonal meshes offer advantages. In particular, we focus on the simulation of flow in fractured media, specifically using the Discrete Fracture Network (DFN) model. A residual a-posteriori error estimator tailored for the DFN case is employed. We chose this particular application to emphasize the effectiveness of the algorithm in handling complex geometries. All the numerical tests demonstrate optimal convergence rates for all the tested VEM orders.

1. Introduction

The necessity for the numerical resolution to partial differential equations (PDEs) in highly complex geometries arises from many applications in both science and engineering. Dealing with these domains, characterized by intricate geometries and where the generation of a high-quality triangular mesh is either computationally expensive or impractical, has led to the adoption of general polygonal meshes in recent years. In this context, there is a natural demand for a numerical method capable to tackle generic polygonal tessellation, and one of such method is the Virtual Element Method (VEM), [1,2].

Moreover, it is widely recognized that a posteriori error estimates are essential for ensuring the reliability of simulation tools. Indeed, ad-hoc adaptive algorithms can be developed to reduce computational effort by leveraging the ability to control the approximation error with cost-effective computable quantities localized on mesh cells.

The study of adaptive schemes in conjunction with polygonal elements has garnered significant attention over the past few years. Noteworthy analyses are currently being conducted in the field of the Discontinuous Galerkin (DG) method, see [3,4], while novel approaches incorporating neural networks have been proposed for both VEM and DG methods, see [5]. Within the VEM literature, numerous works have emerged following the introduction of the a-posteriori estimator in [6,7]. These works span both theoretical [8,9] and practical [10] perspectives.

In this study, we present a novel refinement algorithm, designed for generic two-dimensional second order elliptic PDEs, extending the idea introduced in [10,11]. Our approach starts with a generic polygonal mesh featuring the minimum number of Degrees of Freedoms (DOFs) consistent with the domain geometry, while meeting the minimum regularity requirements for the VEM convergence. The refinement process involves the application of an optimal strategy to enhance both the mesh quality and the

* Corresponding author.
E-mail addresses: stefano.berrone@polito.it (S. Berrone), fabio.vicini@polito.it (F. Vicini).
[1] Member of INdAM research group GNCS.

https://doi.org/10.1016/j.matcom.2024.12.007
Received 15 March 2024; Received in revised form 25 September 2024; Accepted 5 December 2024
Available online 20 December 2024
0378-4754/© 2024 The Authors. Published by Elsevier B.V. on behalf of International Association for Mathematics and Computers in Simulation (IMACS). This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Table 1
Paper notation summary.

Symbol	*Description*
$\Omega \subset \mathbb{R}^2$	Open, connected, and bounded set
$\partial\Omega$	Boundary of Ω
$\mathcal{T}_\Omega$, $\mathcal{E}_\Omega$	Tessellation of Ω and its edges
$E \in \mathcal{T}_\Omega$, $e \in \mathcal{E}_\Omega$	Polygonal mesh element and edge mesh element
$\mathcal{E}_E$, N_E	Set of edges of E and its size
$\lvert E\rvert$, $\mathbf{x}_E$, D_E	Size, barycenter and diameter of E
D	max D_E for all $E \in \mathcal{T}_\Omega$
H_E, h_E	Longest and smallest edge of E
r_E	Min distance between $\mathbf{x}_E$ and the edges of E
R_E	Max distance between $\mathbf{x}_E$ and the vertices of E
AR_E^{Rr}	Aspect ratio indicator $\frac{R_E}{r_E}$
AR_E^{Rh}	Aspect ratio indicator $\frac{R_E}{h_E}$
$\mathbb{I}_E$	Inertia tensor of E with respect to $\mathbf{x}_E$
$\mathcal{I}_E^e$	Edges of E aligned with $e \in \mathcal{E}_E$
$\mathcal{N}_e$	Neighbouring cells of $e \in \mathcal{E}_\Omega$

numerical solution accuracy. This procedure is controlled by two parameters, allowing for a smooth transition of the tessellation from a poor quality complex polygonal mesh to a quasi-triangular or quasi-quadrilateral one within a few steps.

We consider the application of our proposed refinement algorithm to the simulation of the flow in fractured media. The Discrete Fracture Network (DFN) model is employed. We take advantage of the residual a-posteriori error estimator introduced in [7] and extended to the DFN case in [10]. We remark that we choose this model problem due to its intrinsic generation of highly complex geometries and a large number of aligned edges in the initial tessellation, [12–14]. However, we stress that the strategy outlined in this paper is versatile and can be applied to any two-dimensional problem where polygonal meshes offer significant advantages.

The outline of the paper unfolds as follows. In Section 2, we introduce the problem's context, the VEM discrete space, and the adaptive scheme. In Section 3, we focus on the newly proposed refinement strategy. Section 4 describes the DFN problem and the mesh generation on the media. This section also introduces the a-posteriori error estimator. Finally, in Section 5, we present the numerical results across three distinct cases of increasing complexity. Namely a generic two-dimensional application and two scenarios involving DFNs.

Throughout this work, we use the following notations. Given an open set $\omega \subset \mathbb{R}^2$ we indicate with $(\cdot,\cdot)_\omega$ and with $\|\cdot\|_\omega$ the $L^2(\omega)$ inner product and norm, respectively.

2. Setting and target

Table 1 reports a brief summary of the notation used in this paper. We start considering, for this and the following Section 3, a second order Poisson's problem defined on a generic planar polygonal domain $\Omega \subset \mathbb{R}^2$. We prescribe on the whole boundary, $\partial\Omega$, zero Dirichlet boundary conditions, hence we look for the variational solution U living in the space $V := H_0^1$. Different second order Partial Differential Equations (PDE) and boundary conditions can be considered.

For the discrete approximation of U, we assume the existence of a tessellation $\mathcal{T}_\Omega$ on Ω, composed by a finite number of polygonal cells $E \in \mathcal{T}_\Omega$. We admit on $\mathcal{T}_\Omega$ the existence of aligned edges in the same element.

Throughout this section and the subsequent discussion, we use $\#\mathcal{T}_\Omega$ to indicate the total number of polygons in the mesh $\mathcal{T}_\Omega$. We group all the edges of the tessellation into the set $\mathcal{E}_\Omega$. The set of neighbouring cells of an edge $e \in \mathcal{E}_\Omega$ is denoted by $\mathcal{N}_e$. It is worth noting that, in the application under consideration of Section 4, the number of neighbouring cells $\#\mathcal{N}_e$ for and edge $e \in \mathcal{E}_\Omega$ might be higher than two (see Fig. 10). Considering a generic polygon $E \in \mathcal{T}_\Omega$, we use N_E to denote its number of vertices and edges. The set of all the edges of E is indicated by $\mathcal{E}_E$. Additionally, H_E and h_E refer to the longest and the smallest edge length $|e|$, respectively, for all $e \in \mathcal{E}_E$. When selecting an edge $e \in \mathcal{E}_E$, $\mathcal{I}_E^e$ represents the collection of all the edges of E aligned with e. Moreover, $|\mathcal{I}_E^e|$ stands for the sum of the lengths of these aligned edges. The centroid of the element E is denoted by $\mathbf{x}_E = (x_E, y_E)$, and the inertia tensor with respect to $\mathbf{x}_E$ and the x, y-axis is defined as

$$\mathbb{I}_E := \begin{bmatrix} \|y - y_E\|_E^2 & -(x - x_E, y - y_E)_E \\ -(x - x_E, y - y_E)_E & \|x - x_E\|_E^2 \end{bmatrix}.$$

We account by r_E for the minimum distance between the centroid $\mathbf{x}_E$ and the edges of E and by R_E the maximum distance between the centroid $\mathbf{x}_E$ and the vertices of E. Finally, D_E refers to the diameter of the element E and we define the *mesh size* of $\mathcal{T}_\Omega$ as $D := \max_{E\in\mathcal{T}_\Omega} D_E$.

2.1. VEM space

We introduce the Virtual Element Method (VEM) to address the computation of a discrete solution on a polygonal mesh $\mathcal{T}_\Omega$. For a given $k \in \mathbb{N}^+$, we denote by $\mathbb{P}_k(E)$ the set of polynomials of degree up to k defined on each element $E \in \mathcal{T}_\Omega$. With the same notation used in [15], we define the projectors $\Pi_k^\nabla : H^1(E) \to \mathbb{P}_k(E)$ and $\Pi_k^0 : L^2(E) \to \mathbb{P}_k(E)$ such that

$$\begin{cases} \left(\nabla \Pi_k^\nabla v - \nabla v, \nabla p\right)_E = 0 & \forall p \in \mathbb{P}_k, \\ (\Pi_k^\nabla v - v, 1)_{\partial E} = 0 & \text{if } k = 1, \\ (\Pi_k^\nabla v - v, 1)_E = 0 & \text{if } k > 1, \end{cases}$$

and

$$\left(\Pi_k^0 v - v, p\right)_E = 0 \ \forall p \in \mathbb{P}_k.$$

As done in [1], $\forall E \in \mathcal{T}_\Omega$ we introduce the local VEM space as

$$V_k^E = \left\{ v \in H^1(E) : v_{|_e} \in \mathbb{P}_k(e) \ \forall e \in \mathcal{E}_E, \ v_{|_{\partial E}} \in C^0(\partial E), \ \Delta v \in \mathbb{P}_k(E), \ \left(v - \Pi_k^\nabla v, p\right)_E = 0 \ \forall p \in \mathbb{P}_k(E)/\mathbb{P}_{k-2}(E) \right\}, \tag{1}$$

where $\mathbb{P}_k(E)/\mathbb{P}_{k-2}(E)$ denotes the polynomials $\mathbb{P}_k(E)$ that are L^2-orthogonal to $\mathbb{P}_{k-2}(E)$.

Finally, we define the global discrete space $V_k \subseteq V$ as:

$$V_k = \left\{ v \in V : v \in C^0(\Omega), \ v_{|_E} \in V_k^E, \ \forall E \in \mathcal{T}_\Omega \right\}. \tag{2}$$

We uniquely identify a function $v \in V_k$ using the set of Degrees of Freedoms (DOFs):

- the values of v at N_E vertices of E;
- if $k > 1$, the values of v at $k-1$ Gauss internal quadrature points of each of the N_E edges of E;
- if $k > 1$, the internal scaled moments of order $k-2$ of v in E.

This choice is not unique, see another example in [1], however the selected DOFs are unisolvent for V_k [2].

2.2. Target

The goal of this work is to present an adaptive procedure based on the well-known scheme

$$\texttt{SOLVE} \rightarrow \texttt{ESTIMATE} \rightarrow \texttt{MARK} \rightarrow \texttt{REFINE}. \tag{3}$$

The procedure we propose is tailored for a generic problem discretized using the VEM scheme, but can be extended to different polygonal methods. We start the Process (3) with a generic polygonal mesh $\mathcal{T}_\Omega^0$ which presents the number of DOFs as low as possible. To achieve this target, we opt for $\mathcal{T}_\Omega^0$ as the convex polygonal approximation of Ω. This choice minimizes the initial number of DOFs, as suggested in [11,16]. In what follows, we call this particular choice *minimal mesh*. It is worth noting that if Ω is convex, then $\mathcal{T}_\Omega^0$ can be Ω itself.

The modules of Process (3) are defined as follows: for each iteration $m \geq 0$, given a mesh $\mathcal{T}_\Omega^m$

- $[u^m] = \texttt{SOLVE}(\mathcal{T}_\Omega^m)$ creates the VEM discrete solution $u^m \in V_k$ of the PDE;
- $[\eta_\Omega^m] = \texttt{ESTIMATE}(\mathcal{T}_\Omega^m, u^m)$ computes an estimation η^m of the error between U and u^m;
- $[\mathcal{M}^m] = \texttt{MARK}(\mathcal{T}_\Omega^m, \eta_\Omega^m)$ selects a subset of cells $\mathcal{M}^m \subseteq \mathcal{T}_\Omega^m$ candidates for refining;
- $[\mathcal{T}_\Omega^{m+1}] = \texttt{REFINE}(\mathcal{T}_\Omega^m, \mathcal{M}^m)$ creates a new fine-tuned discretization splitting the marked elements.

We aim the REFINE rule to be applied to generic (convex) polygonal elements and to preserve or improve the mesh quality. In cases the initial mesh $\mathcal{T}_\Omega^0$ exhibits poor quality, we aim to enhance this quality during subsequent refinement process iterations. It is crucial to note that mesh quality is essential for producing a robust VEM solution in the SOLVE procedure.

We require that the initial mesh $\mathcal{T}_\Omega^0$ consists of convex elements and satisfies the minimal quality conditions essential for the VEM convergence, as outlined in [17]. Throughout this discussion, we omit the superscript m denoting the iteration in the mesh.

The literature indicates that the VEM convergence is guaranteed combining the condition:

1. there exists $\gamma_r \in (0,1)$ independent of D such that

$$r_E \geq \gamma_r D_E, \forall E \in \mathcal{T}_\Omega, \tag{C1}$$

with either one of the conditions:

2. there exists $\gamma_h \in (0,1)$ independent of D such that

$$h_E \geq \gamma_h D_E, \forall E \in \mathcal{T}_\Omega; \tag{C2}$$

3. there exists $N \in \mathbb{N}^+$ independent of D such that

$$N_E \leq N, \forall E \in \mathcal{T}_\Omega; \tag{C3}$$

4. there exists $\gamma_{al} \in (0,1)$ independent of D such that $\forall E \in \mathcal{T}_\Omega$ the boundary ∂E can be sub-divided in K finite disjoint sequence of edges $\mathcal{I}_E^1, \dots, \mathcal{I}_E^K$ such that

$$\frac{\min_{e \in \mathcal{I}_E^k} |e|}{\max_{e \in \mathcal{I}_E^k} |e|} \geq \gamma_{al}, \forall k \in \{1, \dots, K\}. \tag{C4}$$

Condition (C4) can be interpreted in term of the *piecewise quasi-uniformity* of consecutive edges $\mathcal{I}_E^e$ aligned to $e \in \mathcal{E}_E$ (refer to [18] for more details).

We stress that the convexity of the tessellation elements and the required condition (C1) combined with either one of the conditions (C2)–(C4) on the initial mesh $\mathcal{T}_\Omega^0$ do not inherently ensure high-quality mesh elements. Indeed, the elements of $\mathcal{T}_\Omega^0$ may exhibit undesirable *badly-shaped* characteristics, such as collapsing bulks, small edges, an excess of edges, or irregular aligned edge lengths. Badly-shaped elements are characterized by the constants γ_r, γ_h, and γ_{al} being close to zero or to the values of the constant N large.

To address this, we propose a new `REFINE` strategy with the following objectives:

- Generate new sub-polygons with improved quality in term of γ_r and γ_h, if low;
- Control the number of vertices in the sub-polygons, aiming for their reduction during the refinement iterations;
- Limit and decrease the number of aligned edges, or at least produce quasi-uniform sets $\mathcal{I}_E^k$, $\forall E \in \mathcal{T}_\Omega^m$.

3. The `REFINE` module

Algorithm 1 $[\mathcal{T}_\Omega^{m+1}] = \texttt{REFINE}(\mathcal{T}_\Omega^m, \mathcal{M}^m, c_\rho, c_{al})$

Input: the mesh $\mathcal{T}_\Omega^m$, the marked elements $\mathcal{M}^m$, the parameters $\{c_\rho, c_{al}\}$
Output: the refined mesh $\mathcal{T}_\Omega^{m+1}$

for $E \in \mathcal{M}^m$ **do**
  Compute $\hat{E}$ as the polygon formed by the union of aligned edges of E
  Compute the `MAX-MOMENTUM`($\hat{E}$) direction $\mathbf{t}_{mm}$
  Fix the direction $\mathbf{t}$ using `SMOOTH-DIRECTION`($E, \mathbf{t}_{mm}, c_\rho$)
  Split $E = E_1 \cup E_2$ with the line passing through $\mathbf{x}_E$ and parallel to $\mathbf{t}$
  Deactivate E and mark the split edges $e \in \mathcal{E}_{E_1} \cup \mathcal{E}_{E_2}$
  Update the polygon neighbours $\{\mathcal{N}_e \setminus \{E_1, E_2\}, \forall e \in \mathcal{E}_{E_1} \cup \mathcal{E}_{E_2}$ marked $\}$
  Set $\mathcal{Q} = \{\mathcal{N}_e, \forall e \in \mathcal{E}_{E_1} \cup \mathcal{E}_{E_2}$ marked $\}$
  for $E_q \in \mathcal{Q}$ **do**
    for $e \in \mathcal{E}_{E_q}$ marked **do**
      if `CHECK-QUALITY`($E_q, e, 1, c_\rho, c_{al}$) failed **then**
        Set $\mathcal{M}^m = \mathcal{M}^m \cup E_q$
      else
        Unmark e
      end if
    end for
  end for
end for

Algorithm 2 $[\mathbf{t}_{mm}] = \texttt{MAX-MOMENTUM}(\hat{E})$

Input: the polygon $\hat{E}$ with no aligned edges
Output: the cut direction $\mathbf{t}_{mm}$

if $\hat{E}$ is triangle **then**
  Return $\mathbf{t}_{mm}$ as the `NVB`($\hat{E}$) direction
end if
Compute the inertia tensor $\mathbb{I}_{\hat{E}}$ respect the centroid $\mathbf{x}_{\hat{E}}$
Set $\mathbf{t}_{mm}$ the direction parallel to the eigenvector of the maximum eigenvalue of $\mathbb{I}_{\hat{E}}$

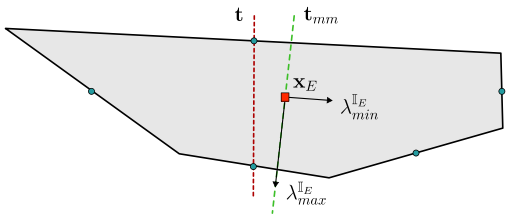


Fig. 1. The split algorithm applied to a generic convex polygon. The green line, $\mathbf{t}_{mm}$, represents the direction of the eigenvector corresponding to the largest eigenvalue of the inertia tensor, denoted as $\lambda^{\mathbb{I}E}max$. The red line, $\mathbf{t}$, indicates the cut direction generated by the SMOOTH-DIRECTION Algorithm. The initial shape of the polygon has measured aspect ratios of $r_E/D_E = 1.06 \cdot 10^{-1}$ and $h_E/D_E = 1.47 \cdot 10^{-1}$. After splitting, the two resulting sub-polygons exhibited improved ratios: $r_E/D_E = 1.67 \cdot 10^{-1}$ and $h_E/D_E = 2.66 \cdot 10^{-1}$ for the left sub-polygon, and $r_E/D_E = 2.09 \cdot 10^{-1}$ and $h_E/D_E = 2.74 \cdot 10^{-1}$ for the right sub-polygon.

Algorithm 3 $[\mathbf{t}] = \texttt{SMOOTH-DIRECTION}(E, \mathbf{t}_{cut}, c_\rho, c_{al})$

Input: the mesh polygon E, the cut direction $\mathbf{t}_{cut}$, the parameters $\{c_\rho, c_{al}\}$
Output: the smoothed direction $\mathbf{t}$

if E is triangle **then**
  Return $\mathbf{t}_{cut}$
end if
Set $\mathbf{t} = \mathbf{t}_{cut}$
for $e \in \mathcal{E}_E$ intersected by $\mathbf{t}_{cut}$ **do**
  if CHECK-QUALITY$(E, e, 2, c_\rho, c_{al})$ failed **then**
    Move $\mathbf{t}$ to intersect the closest vertex of e
  else
    Move $\mathbf{t}$ to intersect the middle point of e
  end if
end for

Algorithm 4 $[bool] = \texttt{CHECK-QUALITY}(E, e, s, c_\rho, c_{al})$

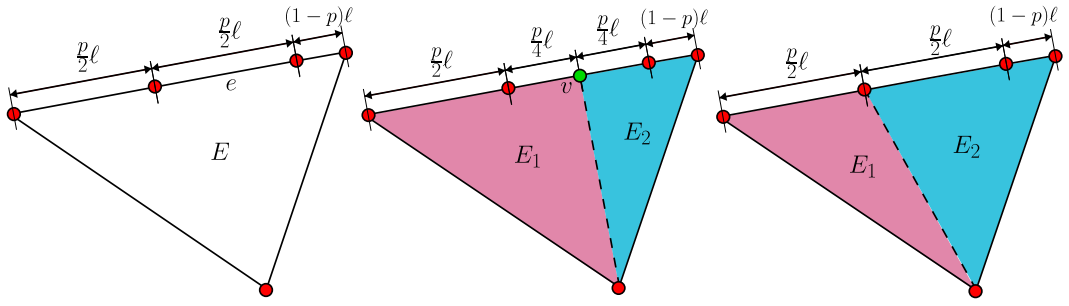
Input: the mesh polygon E, the edge $e \in \mathcal{E}_E$ to check, $s \in \mathbb{N}^+$ the number of e uniform subdivisions, the parameters $\{c_\rho, c_{al}\}$
Output: True if e quality is respected

Set $\rho_e = \max_{E_n} \min\{h_{E_n}, r_{E_n}\}$, $\forall E_n \in \mathcal{N}_e$
if $|e| < c_\rho \rho_e s$ **then**
  Return False
end if
Set $\mathcal{I}^e_{E_n}$ the set of edges $\tilde{e} \in \mathcal{E}_{E_n}$ aligned to e, $\forall E_n \in \mathcal{N}_e$
Compute $|\mathcal{I}^e_{E_n}| = \sum_{e_a \in \mathcal{I}^e_{E_n}} |e_a|$ and $n_{E_n} = \#\mathcal{I}^e_{E_n}$
Set $\mathcal{I}^s_e = \max_{E_n} |\mathcal{I}^e_{E_n}|/(n_{E_n} + (s-1))$
if $|e| < c_{al} \mathcal{I}^s_e s$ **then**
  Return False
end if
Return True

We now discuss the primary contribution of this paper, i.e., the REFINE procedure. Algorithm 1 outlines its pseudo-code. We are going to provide a comprehensive description of the function and subsequently delve into the details of its sub-routines.

At the step m of Process (3), the REFINE method starts from the mesh $\mathcal{T}^m_\Omega$, and the set of marked cells $\mathcal{M}^m$ obtained from the MARK module. For every marked cells $E \in \mathcal{M}^m$, the algorithm splits E to decrease the local a-posteriori error estimator. The objective is to bisect $E = \{E_1, E_2\}$ along a direction $\mathbf{t}$, determined by the combination of the outcomes of the MAX-MOMENT Algorithm 2 and SMOOTH-DIRECTION Algorithm 3. Given the direction $\mathbf{t}$, we split the intersected edge of $\mathcal{E}_E$ producing new edges e that are marked, and we update all the neighbouring cells $\mathcal{N}_e \setminus \{E_1, E_2\}$ of these marked edges e.

The VEM's capability to handle aligned edges simplifies the update of each cell in $\mathcal{N}_e \setminus \{E_1, E_2\}$, resulting in the creation of a new polygon containing the new aligned edges e in place of the original one split by $\mathbf{t}$. Subsequently, a set $\mathcal{Q}$ is generated, encompassing the neighbour cells $\mathcal{N}_e$ of the marked edges e. For all the marked edges in the boundary of cells $E_q \in \mathcal{Q}$, the CHECK-QUALITY Algorithm 4 assesses the quality of the marked edges. If the outcome is positive the edge is unmarked. On the other hand, if the outcome is negative, the cell $E_q \in \mathcal{Q}$ is added to the set of the marked cells $\mathcal{M}^m$. This operation, referred to hereafter as *refinement propagation*, along with its incorporation into the Process (3), constitutes a key innovation in this work to ensure the best rate of convergence of the method.



(a) Original E (b) *Old* algorithm (c) *New* algorithm

Fig. 2. Refinement of E - Case 1 - $c_\rho = 0.5$ and $\frac{4}{5} < p < 1$.

For a generic polygon $\hat{E}$ devoid of aligned edges, the MAX-MOMENT function calculates the *max-momentum* direction $\mathbf{t}_{mm}$. This direction is parallel to the eigenvector associated with the maximum eigenvalue of the inertia tensor $\mathbb{I}_E$. An exception is made if the selected cell $\hat{E}$ is a triangle. In this case, we employ the *newest vertex bisection* (NVB) split criterion, see [19]. As demonstrated in [7,11], the *max-momentum* direction has often the ability to generate sub-cells with improved quality compared to the original $\hat{E}$ for elongated elements, as measured by Condition (C1).

The SMOOTH-DIRECTION Algorithm 3 introduces a slight modification to the original *max-momentum* direction $\mathbf{t}_{mm}$ computed on $\hat{E}$. This adjustment involves shifting the cut direction to the middle of the edge $e \in \mathcal{E}_E$ intersected by $\mathbf{t}_{mm}$, or collapsing it to the closest vertex of e if the CHECK-QUALITY Algorithm 4 fails. As proved in [11], this strategy results in the generation of new polygon cells with, at most, the number of vertices of the original cell E. The number of vertices only rises when the modified cut direction $\mathbf{t}$ splits two consecutive edges; in all the other cases, the children cells have fewer or an equal number of vertices as E, see [11]. This ensures control and improvement of Condition (C3).

Fig. 1 illustrates the application of the proposed algorithm to a generic convex polygon. The image demonstrates that the two new polygons created by the split have enhanced quality, exhibiting more balanced aspect ratios, as noted in the caption.

Remark. The proposed algorithm can also be applied to concave simple polygons. However, handling general concave polygons introduces a wide range of unpredictable scenarios. Unlike the convex case, it is impossible to predict the number of sub-polygons generated, leading to increased computational costs for the algorithm. For these reasons, our discussion and analysis are limited to convex polygons. For concave polygons, a pre-processing step is required to divide the original polygon into convex subsets before applying the algorithm.

The CHECK-QUALITY Algorithm 4 is applied to the polygonal cell E targeted for splitting. It assesses whether the edge $e \in \mathcal{E}_E$ can be divided in s uniform parts. In the REFINE algorithm we select $s = 2$. Thus, the selected edge $e \in \mathcal{E}_E$ is split if the two conditions are satisfied:

$$\frac{|e|}{2} \geq c_\rho \max_{E_n \in \mathcal{N}_e} \min\{h_{E_n}, r_{E_n}\}, \tag{4}$$

$$\frac{|e|}{2} \geq c_{al} \max_{E_n \in \mathcal{N}_e} \frac{|\mathcal{I}^e_{E_n}|}{\#\mathcal{I}^e_{E_n} + 1}. \tag{5}$$

We recall that, $\mathcal{I}^e_E$ denotes the set of edges of E aligned to e, and $|\mathcal{I}^e_E|$ and $\#\mathcal{I}^e_E$ represent the total length and the number of the contiguous aligned edge of e, respectively.

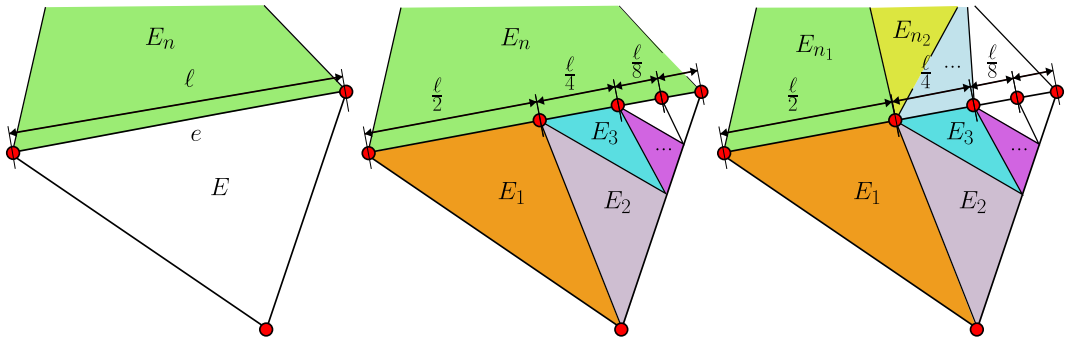
Check (4) ensures that the splitting of the edge contributes to an improvement of the Condition (C1) and Condition (C2). The real parameter $c_\rho \geq 0.0$ is introduced to relax ($0.0 \leq c_\rho < 1.0$) or tighten ($c_\rho \geq 1.0$) the constraint. In addition, we propose Check (5) to satisfy Condition (C4). This check examines whether the newly created edges are *quasi-uniform* compared to the other aligned edges. The real parameter $c_{al} \geq 0.0$ controls the number of aligned edges permitted in the new refined mesh. Namely, a value of c_{al} lower than 1.0 allows aligned edges, while $c_{al} > 1.0$ often rejects any aligned edges.

3.1. Contributions of the new algorithm

The proposed REFINE algorithm represents an extension and enhancement of the concepts introduced in [7,11]. In the subsequent, we elucidate our motivation for developing Algorithm 1 by drawing a comparison with Algorithm 4 presented in [11]. In what follows, we reserve the label *New* to the former algorithm, and we refer as the *Old* to the latter one.

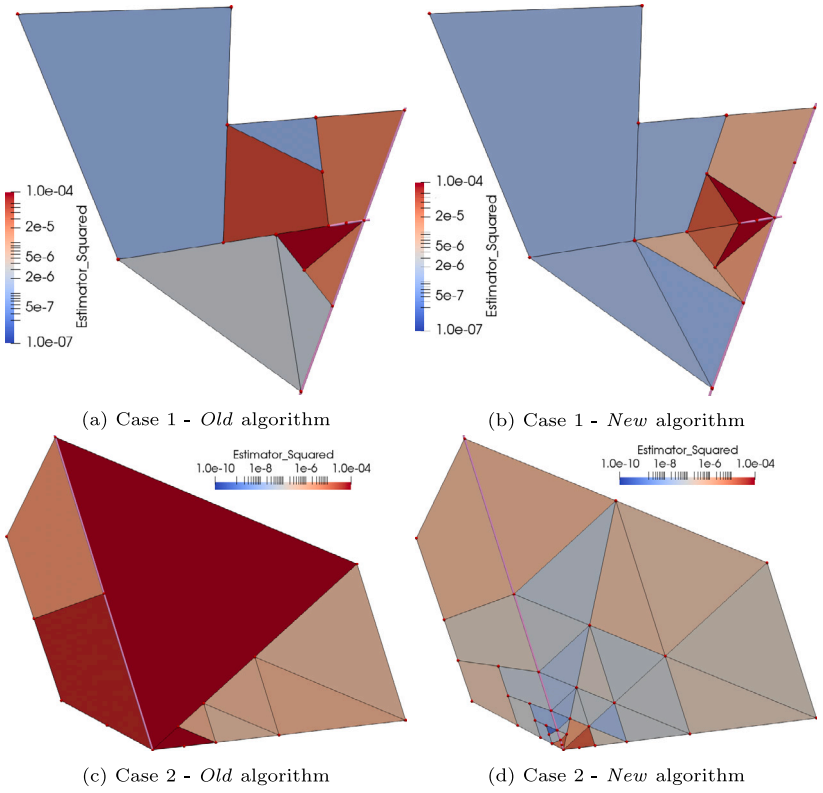
A pivotal innovation in our work is the incorporation of Check (5) in Line 8 within the CHECK-QUALITY Algorithm 4. This subtle modification change is instrumental in controlling Condition (C4), resulting in a comprehensive enhancement of the mesh quality. In Fig. 2, we provide an illustrative example where Check (5) plays a fundamental role.

Consider a scenario where the triangle with aligned edges E of Fig. 2(a) is split in the edge e. We assume $|\mathcal{I}_e| = \ell$, $\#\mathcal{I}_e = 3$, and the existence of a short edge in $\mathcal{I}_e$ of length $(1-p)\ell$, with $c_\rho = 0.5$. In the *Old* algorithm, when the portion $(1-p)$ is sufficiently small,



(a) Original E (b) *Old* algorithm (c) *New* algorithm

Fig. 3. Refinement of E - Case 2 - $c_\rho = 0.5$.



(a) Case 1 - *Old* algorithm (b) Case 1 - *New* algorithm

(c) Case 2 - *Old* algorithm (d) Case 2 - *New* algorithm

Fig. 4. Refinement of E - Measure of the local error estimator on a numerical case.

i.e. $p \in (\frac{4}{5}, 1)$, Check (4) permits the creation of a new aligned edge, as depicted in Fig. 2(b). However, this new edge deteriorates Condition (C4). On the other hand, in the *New* algorithm, the introduction of Check (5) enables the collapse of the cut direction to an original vertex, as illustrated in Fig. 2(c). Figs. 4(a)–4(b) show the local error estimators in a numerical example where this specific condition arises. It is evident from the colouring of the cells that the *Old* algorithm (Fig. 4(a)) results in a higher mean error estimator compared to the *New* proposed version, (Fig. 4(b)) at the same refinement step m.

The *refinement propagation* operation, a key component of the `REFINE` Algorithm 1, is encapsulated in Line 12 and represents the other noteworthy innovation in our work. This operation addresses a crucial aspect of the refinement process by extending it to all the polygonal cells $E_q \in \mathcal{Q}$ containing marked edges that fail Checks (4)–(5) in the `CHECK-QUALITY` algorithm. Note that, during this `CHECK-QUALITY` task, we set $s = 1$ since no edge split is necessary. We recall that we selectively mark the edges $e \in E$ split by the direction $\mathbf{t}$. This strategic approach allows us to handle the scenario depicted in Fig. 3. We illustrate a situation where the estimator consistently requires the refinement of cell E of Fig. 3(a) in each step m. The initial cut applies along the longest edge e. Moreover, we suppose the neighbour cell E_n to have r_{E_n} larger than e and $h_{E_n} = e$. As the refined cells $E_1, E_2, \dots$ are always

triangles, in the *Old* algorithm proposed in [11], as seen in Fig. 3(b), the neighbour cell E_n experiences the creation of a non-uniform aligned edge collection $\mathcal{I}^e_{E_n}$ as a consequence. In contrast, our proposed *New* version, shown in Fig. 3(c), extends the refinement if the quality Check (5) is not met. In Figs. 4(c)–4(d), we present a numerical test to illustrate the local mean estimator substantial improvement in the *New* case. This empirical evidence reinforces the efficacy of our approach.

Remark. The introduced *refinement propagation* operation results in a finite sequence of new refinements, as only two possible situations can occur. If the neighbour is not a triangle, the polygon is subdivided along the direction of maximum momentum without further propagation. If the neighbour is a triangle, the cut direction follows the newest vertex bisection (NVB) rule, potentially leading to additional propagations. However, even in the worst-case scenario with only triangular neighbours, the number of propagation steps is finite and bounded by a constant dependent on the initial discretization (as shown in Lemma 2 of [19]).

We conclude with a final consideration regarding the pivotal role played by the constants c_ρ and c_{al}. As extensively discussed in [11], the configuration of the resulting cells in the mesh $\mathcal{T}^{m+1}_\Omega$ is significantly influenced by the values assigned to c_ρ. Indeed, the number of aligned edges experiences a notable increase when c_ρ tends towards 0. Conversely, when c_ρ assumes a much larger value, the prevalence of cells shape shifts towards triangular mesh elements. The parameter c_{al} plays a analogous role, with an increase in aligned edges as c_{al} approaches zero.

The interdependence of c_ρ and c_{al} is evident, stemming from their inherent connection to the geometric properties of the cells. When $c_\rho \gg 1.0$ and the mesh elements exhibit uniform sizes, Check (4) predominantly fails, rendering Check (5) unused and rendering the value of c_{al} irrelevant. The converse holds true as well. We are going to investigate and discuss the impact of c_ρ and c_{al} on the mesh element shapes in $\mathcal{T}^m_\Omega$ in the section devoted for numerical results.

4. The problem and the discretization

We are going to demonstrate the effectiveness of the newly proposed Process (3) by focusing on the simulation of the flow in fractured porous media. Specifically, we employ the Discrete Fracture Network (DFN) framework. We select this problem due to its natural generation of an initial discretization $\mathcal{T}^0_\Omega$ that exhibits highly *badly-shaped* features. It is crucial to emphasize that the strategy outlined in this paper is applicable to any two-dimensional problem where polygonal meshes are valuable.

Consider a DFN domain Ω, comprising the union of I polygonal planar fractures $F_i \in \mathbb{R}^3$, $\Omega := \bigcup_i F_i$, with $i \in \{1, \dots, I\}$. These fractures are randomly oriented in three-dimensional space, and their intersections create M segments denoted by $S_m \in \mathbb{R}^3$, where $m \in \{1, \dots, M\}$. We assume that each segment S_m results from the intersection of exactly two fractures, establishing a one-to-one relationship between the intersection index m and two fracture indices, i.e., $S_m = \bar{F}_i \cap \bar{F}_j$, $\forall m \in \{1, \dots, M\}$.

The boundary $\partial\Omega = \bigcup_i \partial F_i$ is divided into a Dirichlet portion $\Gamma_D \neq \emptyset \subseteq \partial\Omega$ and a Neumann portion $\Gamma_N = \partial\Omega \setminus \Gamma_D$. Dirichlet conditions are prescribed by the function $g_D \in H^{\frac{1}{2}}(\Gamma_D)$ and a homogeneous Neumann condition is applied on Γ_N. We remark that different boundary conditions can be considered.

For each fracture F_i, we introduce the following functional spaces:

$$V_i^D = \left\{ v \in H^1(F_i) : v_{|_{\Gamma_D \cap \partial F_i}} = g_D \right\}, \quad V_i = \left\{ v \in H^1(F_i) : v_{|_{\Gamma_D \cap \partial F_i}} = 0 \right\}.$$

Moreover, on the whole DFN Ω we define

$$\begin{aligned} V^D &= \left\{ v : v_{|_{F_i}} \in V_i^D \ \forall i \in \{1, \dots, I\},\ \gamma_{S_m}(v_{|_{F_i}}) = \gamma_{S_m}(v_{|_{F_j}}) \ \forall m \in \{1, \dots, M\} \right\}, \\ V &= \left\{ v : v_{|_{F_i}} \in V_i \ \forall i \in \{1, \dots, I\},\ \gamma_{S_m}(v_{|_{F_i}}) = \gamma_{S_m}(v_{|_{F_j}}) \ \forall m \in \{1, \dots, M\} \right\}, \end{aligned} \tag{6}$$

where $\gamma_{S_m} : H^1(F_i) \to H^{\frac{1}{2}}(S_m)$ represents the trace operator onto S_m, $\forall m \in \{1, \dots, M\}$.

We seek the resolution of the following symmetric PDE: find $U \in V^D$ such that:

$$\sum_{F_i} \left(K_i \nabla U_i, \nabla v \right)_{F_i} = \sum_{F_i} \left(Q_i, v \right)_{F_i} \ \forall v \in V. \tag{7}$$

This equation models Darcy's law applied to the DFN. In this context, U represents the hydraulic head on the network, U_i its restriction on F_i, $K_i \in \mathbb{R}$ denotes the fracture transmissivity, and $Q_i \in L^2(F_i)$ acts as the fracture source term.

4.1. DFN discretization

The application of the Process (3) and the resolution to Problem (7) with the VEM requires the ability to construct a partition $\mathcal{T}^0_\Omega$ on the DFN that globally conforms to the fracture intersections. Our approach is based on the methodology detailed in [20]. The generated mesh $\mathcal{T}^0_\Omega$ results from the following steps:

- On each fracture F_i, we generate a local tessellation $\mathcal{T}^{loc}_{F_i}$. This discretization is not necessarily conforming to the fracture intersections and it is entirely independent of other DFN fractures.

- On each F_i, we cut the elements of $\mathcal{T}^{loc}_{F_i}$ using the fracture intersection segments S_m, $\forall S_m \subset F_i$. If a segment S_m terminates inside a polygonal cell of $\mathcal{T}^{loc}_{F_i}$, the segment is extended to the end of the cell. It is important to note that this extension does not alter the original domain Ω shape. This process yields a new polygonal local mesh $\mathcal{T}^{cnf}_{F_i}$ that conforms locally to the fracture intersections.
- On each intersection $S_m = \bar{F}_i \cap \bar{F}_j$, $m \in \{1, \dots, M\}$, we unify all the mesh nodes of $T^{cnf}_{F_i}$ and $T^{cnf}_{F_j}$ lying on S_m. The resulting union points are added to the edges lying on S_m of each mesh $T^{cnf}_{F_i}$ and $T^{cnf}_{F_j}$.

The union of the $T^{cnf}_{F_i}$, $\forall i \in \{1, \dots, I\}$ meshes produces the globally conforming mesh $\mathcal{T}^0_\Omega$.

In this work, we opt for the fracture *minimal mesh* as our choice for $\mathcal{T}^{loc}_{F_i}$, as introduced in Section 2. This entails the use of a discretization formed by the convex polygonal approximation of $\bar{F}_i$. We will henceforth refer to $\mathcal{T}^0_\Omega$ as the *minimal mesh* for the DFN case.

It is important to note that the mesh generation process described naturally translates to the generation of a tessellation with *bad-shape* elements, featuring generic polygonal elements and aligned edges. The quality of the resulting $\mathcal{T}^0_\Omega$ is strongly influenced by the fractures' spatial location in the tree-dimensional space. In real DFNs, $\mathcal{T}^0_\Omega$ exhibits numerous aligned edges and neighbouring polygonal cells with significant variation in size. We rely on the refining strategy proposed here to enhance the mesh quality.

Thanks to the global conformity of the mesh $\mathcal{T}^0_\Omega$ and the ability to identify the DOFs on each segment S_m, where $m \in \{1, \dots, M\}$, we can define the global discrete spaces of Eq. (6) as

$$V_k^D = \left\{ v \in V^D : v_{|_{F_i}} \in C^0(F_i),\ v_{|_E} \in V_k^E,\ \forall E \in F_i,\ \forall i \in \{1, \dots, I\} \right\}, \tag{8}$$
$$V_k = \left\{ v \in V : v_{|_{F_i}} \in C^0(F_i),\ v_{|_E} \in V_k^E,\ \forall E \in F_i,\ \forall i \in \{1, \dots, I\} \right\}.$$

We refer to Eq. (1) for the definition of V_k^E on each element $E \in \mathcal{T}_\Omega$.

We emphasize that all the quantities for $E \in \mathcal{T}_\Omega$ are computed with respect to the reference system tangential to the fracture F_i to which the element E belongs. Despite the DFN being immersed in $\mathbb{R}^3$, the use of the reference tangential system allows us to work exclusively in the geometric dimension 2. Additionally, the reader shall notice that, in Eq. (8), we employ the same symbol V_k of the space as in Eq. (2), as it naturally extends this to the DFN case.

We discretize Problem (7) as follows: find $u \in V_k^D$ such that:

$$\sum_{F_i} \left(K_i \Pi^0_{k-1} \nabla u_i, \nabla \Pi^0_{k-1} v \right)_{F_i} + S^E_{F_i}(u_i - \Pi^\nabla_k u_i, v - \Pi^\nabla_k v) = \sum_{F_i} \left(Q_i, \Pi^0_{k-1} v \right)_{F_i} \quad \forall v \in V_k \tag{9}$$

where $S^E_{F_i}$ is the stabilizing bilinear form. Among all the available choices, see [21], we opt for the standard form:

$$S^E_{F_i}(u, v) = K_i \sum_{j=1}^{\#V_k^E} \mathrm{dof}^E_j(u) \mathrm{dof}^E_j(v) \quad \forall u \in V_k^D, v \in V_k.$$

4.2. DFN mesh adaptivity

We introduce the VEM residual-type error estimator for the DFN case, which we use in the ESTIMATE function of the adaptive Process (3). We follow the analysis proposed in [7] and its extension to the DFN case reported in [10].

Given u^m as the discrete solution generated by the SOLVE task on each iteration m of Process (3), we define the function $\eta_\Omega : V_k^D \to \mathbb{R}$ s.t.

$$\eta^2_\Omega(u^m) := \sum_{E \in \mathcal{T}^m_\Omega} \frac{D_E^2}{K_i} \left\| \Pi^0_{k-1} Q_i + K_i \Delta u_i^\pi \right\|^2_E + \sum_{e \in \mathcal{E}^m_\Omega \setminus \Gamma_D} \frac{|e|}{K_e} \left\| \sum_{\tilde{E} \in \mathcal{N}_e} K_i \nabla u_i^\pi \cdot n_e^{\tilde{E}} \right\|^2_e + \sum_{E \in \mathcal{T}^m_\Omega} \frac{D_E^2}{K_i} \left\| Q_i - \Pi^0_{k-1} Q_i \right\|^2_E, \tag{10}$$

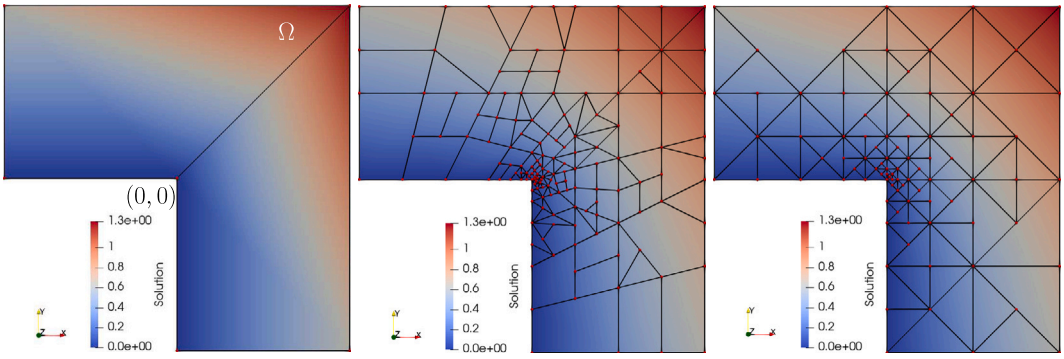
where $u_i^\pi := (\Pi^\nabla_k u^m)_{|_{F_i}}$, $n_e^{\tilde{E}}$ is the unit normal vector to e pointing outward with respect to $\tilde{E} \in \mathcal{N}_e$ and $K_e := \sum_{\tilde{E} \in \mathcal{N}_e} K_i$. The subscript i refers to the fracture F_i to which the element E belongs.

It is possible to prove, as discussed in [7], that there exist two positive constants C_* and C^* such that:

$$C_* \cdot \eta_\Omega(u^m) \le \left| \left| \left| U - \Pi^\nabla_k u^m \right| \right|_\Omega \le C^* \cdot \eta_\Omega(u^m),$$

where U is the solution to Problem (7), and u^m is the solution to Problem (9). For error control, we use the energy norm $|||\cdot||| : V^D \to \mathbb{R}$,

$$|||v|||_\Omega := \sup_{w \in V} \frac{\sum_{F_i} \left(K_i \nabla v, \nabla w \right)_{F_i}}{\left(\sum_{F_i} \left\| \sqrt{K_i} \nabla w \right\|^2_{F_i} \right)^{\frac{1}{2}}} \quad \forall v \in V^D.$$



(a) $\mathcal{T}_\Omega^0$ (b) Step $m = 15$, $c_\rho = 0.5$ (c) Step $m = 15$, $c_\rho = 1.5$

Fig. 5. Test 1 - Domain with solution and, $k = 1$, $c_{al} = 1.0$.

Finally, on each polygon $E \in \mathcal{T}_\Omega^m$, we introduce the local estimator $\eta_E : V_k^D \to \mathbb{R}$ such that:

$$\eta_E^2(u^m) := \frac{D_E^2}{K_i} \left\| \Pi_{k-1}^0 Q_i + K_i \Delta u_i^\pi \right\|_E^2 + \sum_{e \in \mathcal{E}_E \setminus \Gamma_D} \frac{|e|}{\#\mathcal{N}_e \cdot K_e} \left\| \sum_{\hat{E} \in \mathcal{N}_e} K_i \nabla u_i^\pi \cdot n_e^{\hat{E}} \right\|_e^2 + \frac{D_E^2}{K_i} \left\| Q_i - \Pi_{k-1}^0 Q_i \right\|_E^2 . \tag{11}$$

Remark (*Local Flux Estimation*)**.** We incorporate the inverse of $\#\mathcal{N}_e$ to weight the local flux estimation. This choice mirrors the constant $\frac{1}{2}$ typically used in classical $\mathbb{R}^2$ applications, given that $\#\mathcal{N}_e = 2$.

5. Numerical results

In this section, we analyse the performances of the proposed adaptive Scheme (3). For all the performed tests, the initial mesh $\mathcal{T}_\Omega^0$ is selected as the *minimal mesh* introduced in the previous sections.

The marking strategy employed is based on the approach proposed in [22]. The parameter $\theta \in [0.0, 1.0]$ is used to identify a subset of cells $\mathcal{M}^m \subseteq \mathcal{T}_\Omega^m$ such that:

$$\sum_{E \in \mathcal{M}^m} \eta_E^2(u^m) \geq \theta \eta_\Omega^2(u^m),$$

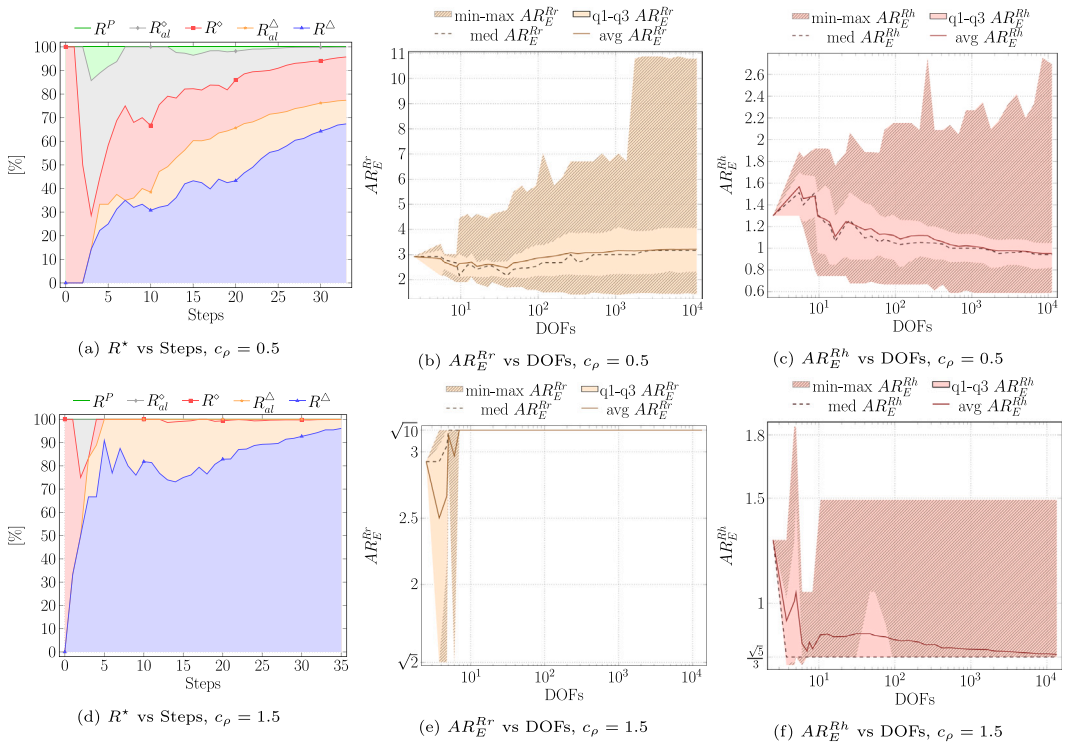
where η_Ω and η_E are the estimators introduced in Eqs. (10) and (11), respectively. Throughout all the tests, we set $\theta = 0.5$, as it yields the most favourable results.

To gauge the mesh $\mathcal{T} = \mathcal{T}_\Omega^m$ quality performance, we introduce the following indicators:

- $AR_E^{Rr} := \frac{R_E}{r_E}$ and $AR_E^{Rh} := \frac{R_E}{h_E}$, serving as local approximations of the inverses of constants γ_r and γ_h, respectively;
- $\#\Delta_\mathcal{T}$ and $\#\diamond_\mathcal{T}$ represent the number of cells $E \in \mathcal{T}$ with 3 and 4 vertices, respectively. These values measure the presence of "real" triangles and quadrilateral on the tessellation.
- $\#\Delta_\mathcal{T}^{al}$ and $\#\diamond_\mathcal{T}^{al}$ denote the number of cells $E \in \mathcal{T}$ whose polygons $\hat{E}$, obtained by unifying the aligned edges $e \in \mathcal{E}_E$, are formed by 3 and 4 vertices, respectively. Note that, $\Delta_\mathcal{T}^{al}$ includes also $\Delta_\mathcal{T}$ cells and some cells of $\diamond_\mathcal{T}$; similarly, $\diamond_\mathcal{T}^{al}$ contains the resulting part of the cells from $\diamond_\mathcal{T}$.
- $R^\Delta := \frac{\#\Delta_\mathcal{T}}{\#\mathcal{T}}$, $R^\diamond := \frac{\#\diamond_\mathcal{T}}{\#\mathcal{T}}$, $R^P := 1 - (R^\Delta + R^\diamond)$ measure the fraction of different polygons contained in the tessellation.
- $R_{al}^\Delta := \frac{\#\Delta_\mathcal{T}^{al}}{\#\mathcal{T}}$, $R_{al}^\diamond := \frac{\#\diamond_\mathcal{T}^{al}}{\#\mathcal{T}}$, measure the percentage of different shapes contained in the tessellation, accounting only polygons with not aligned edges.
- Ef $\frac{\text{DOFs}}{\#\mathcal{T}} := \frac{\#\mathcal{T}}{\#DOFs}$ computes the efficiency of the mesh elements in approximating the numerical solution. A large value indicates better efficiency.

5.1. Test 1: Classic bi-dimensional domain

As first example, we present the solution to Problem (9) with $I = 1$ on an L-shape domain $\Omega := (-1, 1) \setminus (-1, 0)$. This classic test serves as a measure of the application of the newly proposed refinement scheme to a generic two-dimensional problem. The exact solution, prescribed with Dirichlet boundary conditions, is given by $H(r, \beta) = r^{\frac{2}{3}} \sin \frac{2}{3}(\beta + \frac{\pi}{2})$, where r and β denote the polar coordinates. We recall that the exact solution exhibits a singularity at the origin of the axis, as detailed in [23]. Fig. 5(a) displays the domain with the initial *minimal mesh* $\mathcal{T}_\Omega^0$.



(a) $R^\star$ vs Steps, $c_\rho = 0.5$ (b) AR_E^{Rr} vs DOFs, $c_\rho = 0.5$ (c) AR_E^{Rh} vs DOFs, $c_\rho = 0.5$

(d) $R^\star$ vs Steps, $c_\rho = 1.5$ (e) AR_E^{Rr} vs DOFs, $c_\rho = 1.5$ (f) AR_E^{Rh} vs DOFs, $c_\rho = 1.5$

Fig. 6. Test 1 - mesh statistics, $k = 1$, $c_{al} = 1.0$.

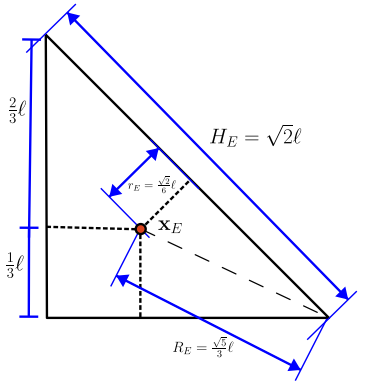
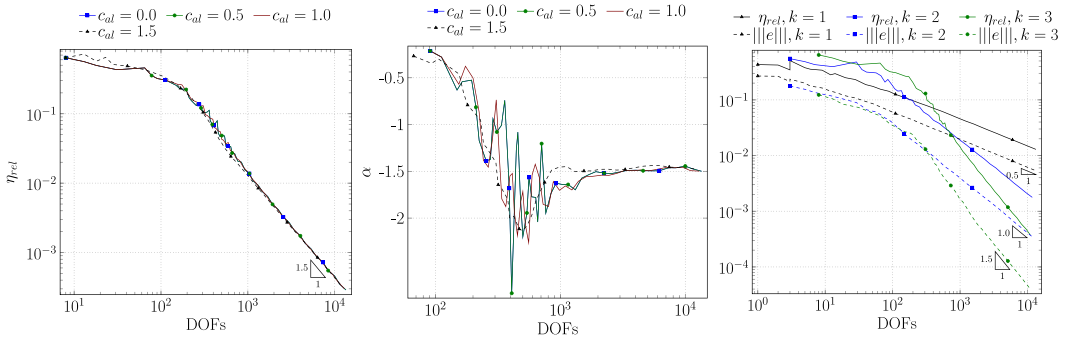


Fig. 7. Ref. Triangle: $AR_E^{Rr} = \sqrt{10}$, $AR_E^{Rh} = \sqrt{5}/3$.

We analyse the mesh quality and the solution approximation obtained during the refinement process for VEM order $k \in \{1, 2, 3\}$, $c_\rho \in \{0.5, 1.5\}$, (as in [11]), and $c_{al} \in \{0.0, 0.5, 1.0, 1.5\}$. The adaptive procedure (3) is interrupted when the total number of DOFs reaches 10^4, irrespectively of the value of a-posteriori error η_Ω.

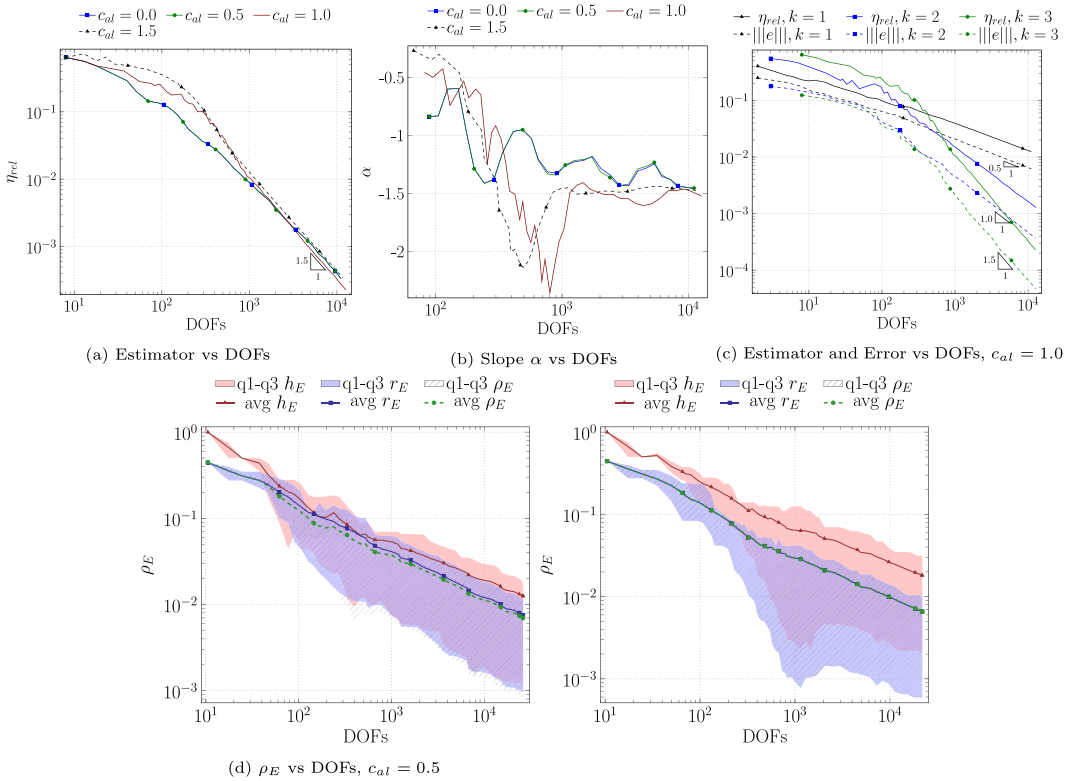
Figs. 5(b) and 5(c) illustrate the mesh $\mathcal{T}_\Omega^m$ after $m = 15$ steps for $c_\rho = 0.5$ and $c_\rho = 1.5$, respectively. We report only the case with $c_{al} = 1.0$ and $k = 1$, as other values exhibits similar attitude. As emphasized in Section 3, a value of $c_\rho < 1$ results in mesh elements with heterogeneous shapes, while $c_\rho > 1$ leads to a more shape-uniform mesh with a higher prevalence of triangular resulting cells. The plots of Figs. 6(a) and 6(d) confirm these observations. In Plot 6(a), we observe a combination of triangles and quadrilaterals ($R^\Delta + R^\diamond \approx 60\%$) and a high percentage of elements with aligned edges ($R^\Delta_{al} + R^\diamond_{al} \approx 40\%$). On the contrary, Plot 6(d) illustrate that the majority of $\mathcal{T}_\Omega^m$ cells are triangles ($R^\Delta \approx 70\%$) or nearly triangles ($R^\Delta_{al} \approx 30\%$) after approximately five steps.

In Figs. 6(b), 6(e) and 6(c), 6(f) we conduct a statistical analysis of the aspect ratios AR_E^{Rr} and AR_E^{Rh}, respectively, on the mesh elements $E \in \mathcal{T}_\Omega^m$ for both the values $c_\rho = 0.5$ and $c_\rho = 1.5$. For each iteration m, we present the minimum and maximum values (min–max $AR_E^\star$ area), the first and third quartile values (q1-q3 $AR_E^\star$ area), the median value (med $AR_E^\star$ curve), and the average value (avg $AR_E^\star$ curve) of $AR_E^\star$, $\star \in \{Rr, Rh\}$. The statistical data reveals that the mesh $\mathcal{T}_\Omega^m$ obtained with $c_\rho = 0.5$ exhibits stable



(a) Estimator vs DOFs, $k = 3$ (b) Slope α vs DOFs, $k = 3$ (c) Estimator and Error vs DOFs, $c_{al} = 1.0$

Fig. 8. Test 1 - convergence analysis, $c_\rho = 1.5$.



(a) Estimator vs DOFs (b) Slope α vs DOFs (c) Estimator and Error vs DOFs, $c_{al} = 1.0$

(d) ρ_E vs DOFs, $c_{al} = 0.5$

Fig. 9. Test 1 - c_{al} analysis, $c_\rho = 0.5$, $k = 3$.

quality as the deviation from the mean value remains constant after a few iterations for both the quality $AR_E^\star$ indicators. In contrast, elements of the mesh $\mathcal{T}_\Omega^m$ obtained with $c_\rho = 1.5$ resemble a rescaled version of the reference triangle in Fig. 7. Indeed, the statistics converge to the characteristic quantities $AR_E^{Rr} = \sqrt{10}$ and $AR_E^{Rh} = \frac{\sqrt{5}}{3}$ of the reference triangle.

As observed in Section 3, when $c_\rho > 1.0$ and the mesh elements present uniform sizes, the parameter c_{al} becomes less relevant. Indeed, in the CHECK-QUALITY Algorithm 4, the Check (4) often fails, and the Check (5) is rarely used. Fig. 8 confirm this observation, presenting an analysis of the parameter $c_{al} \in \{0.0, 0.5, 1.0, 1.5\}$ with a fixed value of $c_\rho = 1.5$. Values $c_{al} = 0.0$ and $c_{al} = 1.5$ represent extremal cases. Value $c_{al} = 0.0$ mimics the scenario where the Check (5) of CHECK-QUALITY Algorithm 4 is disabled. On the other hand, the $c_{al} = 1.5$ value allows no aligned edge. This value is used as a reference, to show the performances of the other meshes compared ($c_{al} \leq 1$) with a pure uniform triangular mesh ($c_{al} = 1.5$).

Fig. 8(a) illustrates the convergence of the relative a-posteriori error estimator

$$\eta_{rel} := \frac{\eta_\Omega}{|||u|||_\Omega}, \tag{12}$$

Table 2
Test 2 - Rates of convergence α for the estimator η_{rel} and the error e_{H^1}, $c_{al} = 1.0$.

c_ρ	Indicator	$k=1$	$k=2$	$k=3$
0.5	η_{rel} vs DOFs	−0.503	−1.019	−1.590
0.5	e_{H^1} vs DOFs	−0.474	−0.980	−1.508
1.5	η_{rel} vs DOFs	−0.498	−1.004	−1.474
1.5	e_{H^1} vs DOFs	−0.497	−1.013	−1.501

and the energy norm relative error

$$|||e||| := \frac{|||U - \Pi_k^\nabla u|||_\Omega}{|||U|||_\Omega}, \tag{13}$$

with respect to the number of DOFs, for different c_{al} values, with VEM order $k = 3$. Moreover, Fig. 8(b) measures, for each iteration $m > 4$ of Process (3), the local convergence rate α based on the previous $(m-5, m-1)$ refinement iterations. The analysis suggests that when $c_\rho > 1.0$ the parameter c_{al} becomes less significant, as at convergence, all curves overlap and reach the optimal rate. The results for orders $k = 1$ and $k = 2$ with different c_{al} values are omitted as they exhibits a similar behaviour. For completeness, Fig. 8(c) displays the convergence of the error estimator for all the VEM orders with $c_\rho = 1.5$ and a fixed value of $c_{al} = 1.0$.

We conclude our analysis by exploring the scenario where $c_\rho \leq 1.0$. In Figs. 9(a) and 9(b) we present the analysis of c_{al} while fixing $k = 3$ and $c_\rho = 0.5$. We display again the convergence of relative estimator η_{rel} of Eq. (12) and the convergence rate α for each iteration m. In Fig. 9(b), we report the rates of convergence and we observe that the best results are achieved with $c_{al} = 1.0$, specifically when $\#\mathcal{I}_E^e \leq 2$, $\forall e \in \mathcal{E}_E$, $\forall E \in \mathcal{T}_\Omega^m$. Notably, the results obtained with $c_{al} = 1.0$ closely resemble those obtained with the pure triangular mesh of $c_{al} = 1.5$. Conversely, when $c_{al} < 1.0$, the curves overlap and the convergence of the estimator appears sub-optimal. We believe that this sub-optimal behaviour may be attributed to the presence of excessively small edges in the mesh. In Figs. 9(d) and 9(e) we provide a statistical analysis of the quantities h_E and r_E, for $c_{al} = 0.5$ and $c_{al} = 1.0$, respectively. The quantity

$$\rho_E := \min\{h_E, r_E\}, \ \forall E \in \mathcal{T}_\Omega^m. \tag{14}$$

is the one used in the CHECK-QUALITY Algorithm 4. We present the first and third quartile values (q1-q3 area) and the average values (avg curve). The analysis of the statistics reveals that when the convergence rates α are not optimal ($c_{al} = 0.5$), the average curve of ρ_E deviates from the average curve r_E, and the value of h_E is nearly equal or lower than r_E. This discrepancy indicates that, on average, when the length of the minimum edge of the mesh elements (h_E) is not significantly higher than the inner radius (r_E), the VEM assumptions regarding the quality of the element edges are not fulfilled, jeopardizing the optimal convergence of the numeric solution, as discussed in Section 2. For these reasons, we select $c_{al} = 1.0$ as the optimal parameter value, and we present the convergence of the estimator η_{rel} in Fig. 9(c) for all VEM orders $k = 1$, $k = 2$, and $k = 3$.

Before proceeding with the DFN tests, we stress that we obtain the optimal order in the VEM approximations selected and for both choices of c_ρ, thanks to the introduction of the parameter c_{al} in Check (5), despite the differences in the resulting mesh shapes.

5.2. Test 2: Regular DFN with three fractures

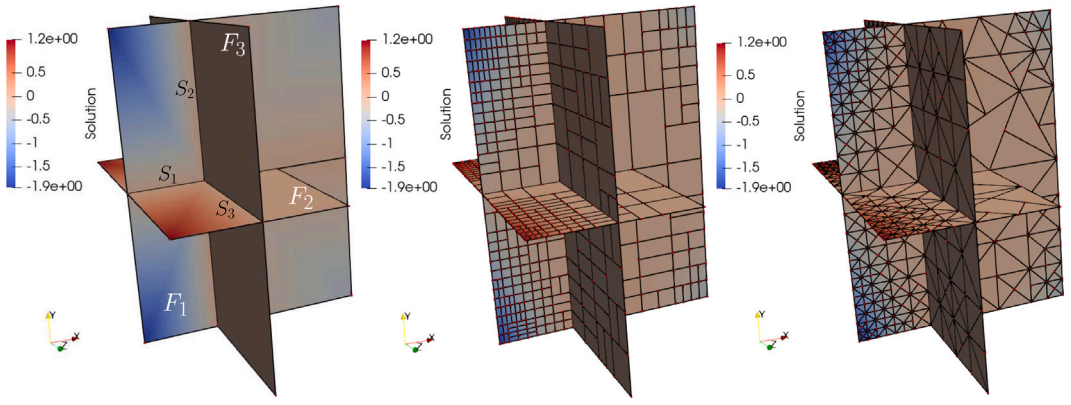
We present a benchmark for a three-fractures DFN test. The detailed problem information are available in [24]. We prescribe the exact solution on each fracture as follows:

$$\begin{aligned} F_1 &\ -\tfrac{1}{10}(x+\tfrac{1}{2})\left[8xy(x^2+y^2)\arctan(y,x)+x^3\right], \\ F_2 &\ -\tfrac{1}{10}(x+\tfrac{1}{2})x^3(1-8\pi|z|), \\ F_3 &\ y(y-1)(y+1)(z-1)z. \end{aligned}$$

Fig. 10(a) illustrates the network with the initial mesh $\mathcal{T}_\Omega^0$. As in the previous case, the subsequent analysis employs VEM order $k = \{1, 2, 3\}$, $c_\rho \in \{0.5, 1.5\}$, and $c_{al} = 1.0$. We do not report any analysis varying c_{al} due to negligible differences obtained, attributed to the large regularity of the original network geometry. This test, in contrast to the concave L-shape domain, encompasses all convex domains with orthogonal intersections. The iterative scheme (3) terminates when the total number of DOFs reaches $5 \cdot 10^5$.

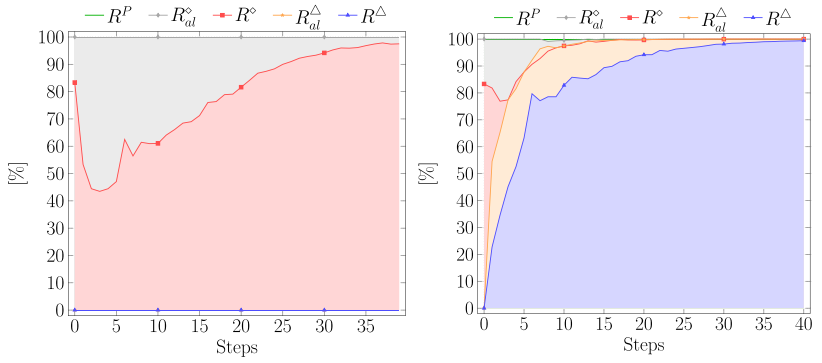
Figs. 10(b) and 10(c) depict the mesh and solution at refinement step $m = 15$ for $c_\rho = 0.5$ and $c_\rho = 1.5$, respectively. Significant differences in the generated meshes are evident. Indeed, the analysis of the $R^\star$ values in Fig. 11(a) reveals that for $c_\rho < 1$ the mesh, $\mathcal{T}_\Omega^m$, converges to a full quadrilateral mesh. This behaviour is attributed, once again, to the regularity of the original network geometry. On the other hand, in Fig. 11(b), when $c_\rho > 1$, the mesh $\mathcal{T}_\Omega^m$ approaches to a fully triangular discretization. The plots of the $AR_E^\star$ indicators are omitted since all indicators trivially converge to the characteristic values of the reference square $[0,1]\times[0,1]$ and the reference triangle shown in Fig. 7 for $c_\rho = 0.5$ and $c_\rho = 1.5$, respectively.

It is important to note that the shape regularity observed in this network is atypical in real DFN applications. Nevertheless, this test was specifically designed to have a known exact solution.



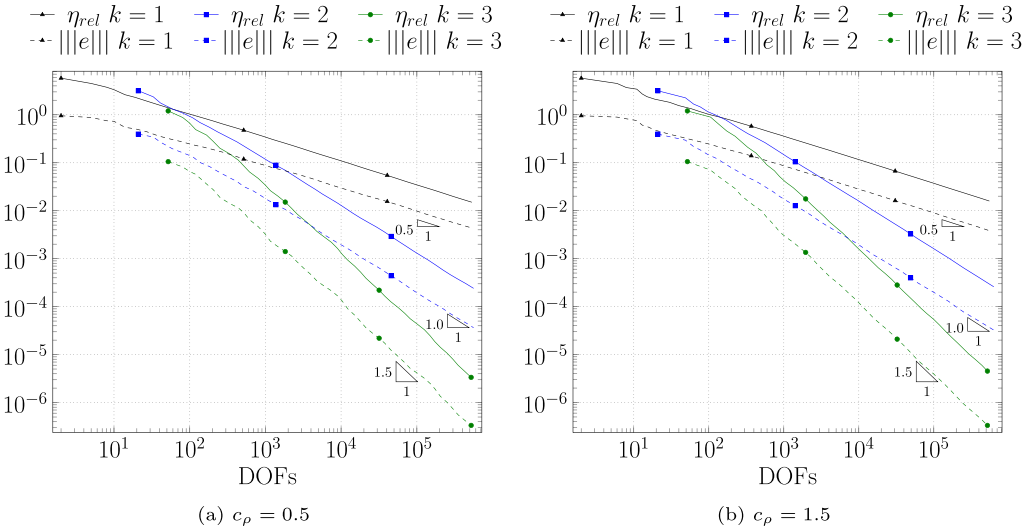
(a) DFN $\mathcal{T}_\Omega^0$ (b) Step $m = 15$, $c_\rho = 0.5$ (c) Step $m = 15$, $c_\rho = 1.5$

Fig. 10. Test 2 - Network with solution, $k = 1$, $c_{al} = 1.0$.



(a) $c_\rho = 0.5$ (b) $c_\rho = 1.5$

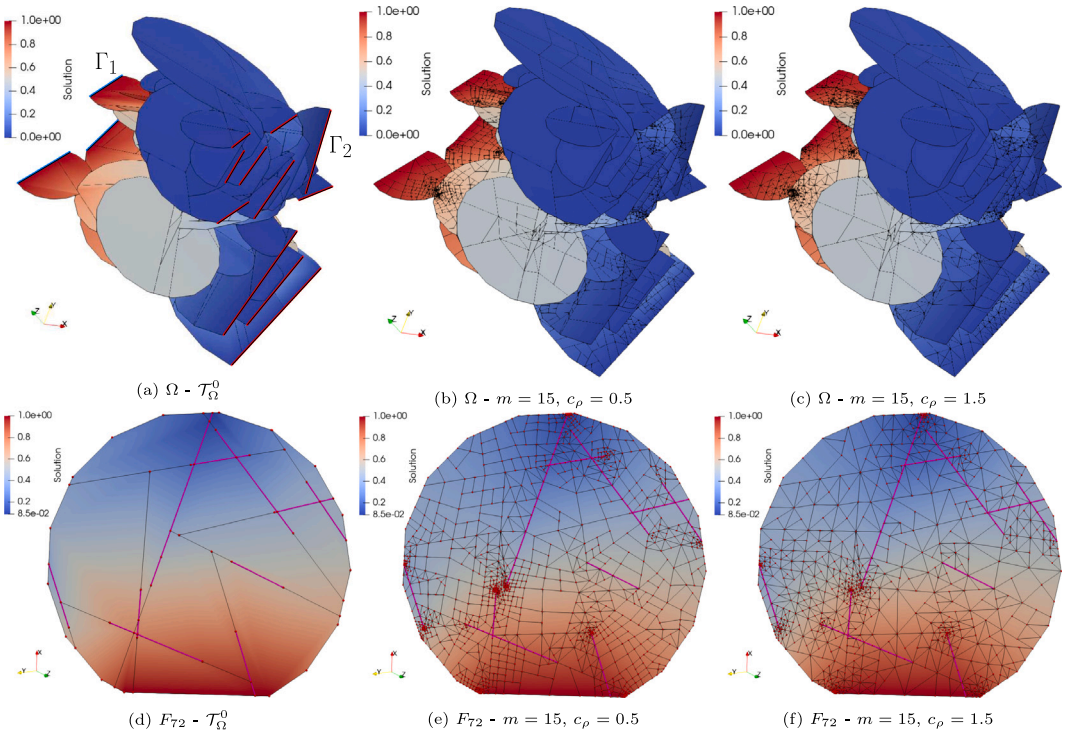
Fig. 11. Test 2 - $R^\star$ versus m-steps, $k = 1$, $c_{al} = 1.0$.



(a) $c_\rho = 0.5$ (b) $c_\rho = 1.5$

Fig. 12. Test 2 - Relative Estimator (η_{rel}) and energy error ($|||e|||$) versus DOFs, $c_{al} = 1.0$.

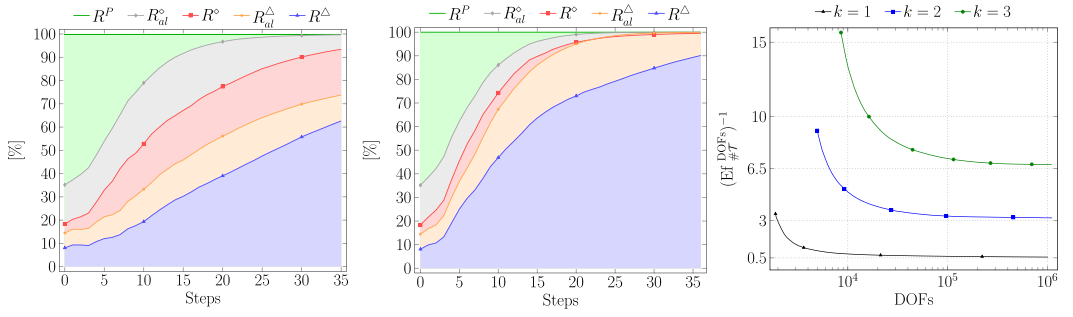
Fig. 12 present the convergence of the relative estimator η_{rel} of Eq. (12) and the energy norm relative error of Eq. (13) for each DOFs. Additionally, Table 2 details the convergence rates α, computed on the last 5 iterations of refinement. It can be affirmed



(a) Ω - $\mathcal{T}^0_\Omega$ (b) Ω - $m = 15$, $c_\rho = 0.5$ (c) Ω - $m = 15$, $c_\rho = 1.5$

(d) F_{72} - $\mathcal{T}^0_\Omega$ (e) F_{72} - $m = 15$, $c_\rho = 0.5$ (f) F_{72} - $m = 15$, $c_\rho = 1.5$

Fig. 13. Test 3 - Solution on the network Ω and on F_{72}, $k = 1$, $c_{al} = 1.0$.



(a) $R^\star$ vs m-steps, $c_\rho = 0.5$ (b) $R^\star$ vs m-steps, $c_\rho = 1.5$ (c) $(\mathrm{Ef}^{\mathrm{DOFs}}_{\#\mathcal{T}})^{-1}$, $c_\rho = 1.5$ vs DOFs

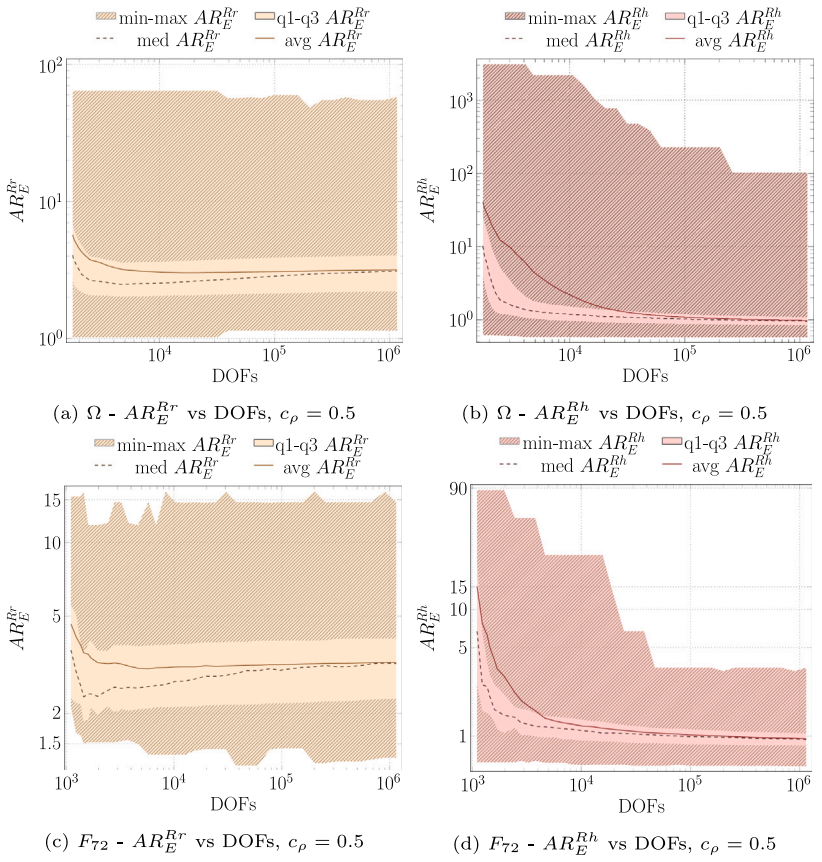
Fig. 14. Test 3 - Mesh $\mathcal{T}^m_\Omega$ shapes, $k = 1$, $c_{al} = 1.0$.

that optimal convergence rates α are achieved for both the estimator η_{rel} and the relative error $|||e|||$ for each VEM order k and each choice of c_ρ.

5.3. Test 3: Random DFN

To conclude the analysis, we discuss the results obtained for a more realistic DFN case. We select the benchmark test analysed in [10]. The network, illustrated in Fig. 13(a), features a computed *minimal mesh* $\mathcal{T}^0_\Omega$ generated by 86 fractures and 159 fracture intersections. We compute the solution to Problem (7) by imposing two constant Dirichlet conditions, $\Gamma_1 = 1.0$ and $\Gamma_2 = 0.0$, on the fracture edges that intersect the planes $x = 0.0$ and $x = 1000.0$, respectively. Other fracture edges have homogeneous Neumann conditions. With blue and red thick edges in Fig. 13(a) we highlight Γ_1 and Γ_2, respectively.

This DFN test is generated with random fracture orientation, size, and localization drawn from random distributions computed from a real fractured media. Consequently, the initial network mesh $\mathcal{T}^0_\Omega$ consists of several generic polygonal elements with differing size and a notable number of aligned edges. Fig. 13(d) offers a closer look at the initial mesh of fracture $F_{72} \in \Omega$ as a detail of the global mesh $\mathcal{T}^0_\Omega$. The fracture intersections are highlighted with purple lines. The complexity and the poor element quality are recognizable.



(a) Ω - AR_E^{Rr} vs DOFs, $c_\rho = 0.5$

(b) Ω - AR_E^{Rh} vs DOFs, $c_\rho = 0.5$

(c) F_{72} - AR_E^{Rr} vs DOFs, $c_\rho = 0.5$

(d) F_{72} - AR_E^{Rh} vs DOFs, $c_\rho = 0.5$

Fig. 15. Test 3 - Mesh $\mathcal{T}_\Omega^m$ quality, $k = 1$, $c_{al} = 1.0$.

The analysis of the mesh involves VEM order $k \in \{1, 2, 3\}$, $c_\rho \in \{0.5, 1.5\}$, and $c_{al} = \{0.0, 0.5, 1.0\}$. We stop the iterative scheme (3) when the total number of DOFs reaches 10^6.

Figs. 13(b)–13(c) and 13(e)–13(f) illustrate the mesh obtained after 15 steps in the network Ω and in F_{72} for $c_\rho = 0.5$ and $c_\rho = 1.5$, respectively. Notably, the mesh refinement algorithm strategically focuses around the tips of fracture intersections where the solution exhibits a singularity $H^{\frac{3}{2}-\epsilon}$.

In the plots of Figs. 14(a) and 14(b), we measure the $R^\star$ indicators for both the c_ρ values. The fracture mesh detail and the analysis of the plots revel that higher values of c_ρ lead to an increased number of mesh triangles, consistent with the observations from previous tests. When $c_\rho = 1.5$, almost the entire number of elements consist of triangular or nearly triangular ($R^4 + R^4_{al} \approx 100\%$). In addition, despite the initially prevalence of generic polygons in $\mathcal{T}^0_\Omega$ ($R^P \approx 65\%$), both meshes ultimately converge to tessellations containing predominantly triangular or quadrilateral elements or those that are nearly triangular and quadrilateral ($R^P \approx 0$). This highlights the robustness of the approach in improving the shape-uniformity of the elements on generic polygonal meshes.

Fig. 14(c) presents the inverse of Ef $^{\text{DOFs}}_{\#\mathcal{T}_\Omega}$ indicator against the number of DOFs for $c_\rho = 1.5$. The data for $c_\rho = 0.5$ is omitted as the plots are qualitatively similar. This indicator $(\text{Ef}\,^{\text{DOFs}}_{\#\mathcal{T}_\Omega})^{-1}$ is presented as it converges to known values for triangular meshes and it evaluates the inefficiency of the mesh elements in approximating the numerical solution as the refinement process progresses. Indeed, the indicator starts from very high values and rapidly converges to characteristic values of a triangular mesh, namely 0.5, 3 and 6.5 for $k = 1$, $k = 2$ and $k = 3$, respectively. Thus, the measurement of $(\text{Ef}\,^{\text{DOFs}}_{\#\mathcal{T}_\Omega})^{-1}$ ensures that, by the end of the refinement process, maximum mesh efficiency in terms of DOFs is achieved.

In Fig. 15, we present a comprehensive statistical analysis of the $AR_E^\star$ quantities, where $\star \in \{Rh, Rr\}$, for the $c_\rho = 0.5$ case in the DFN Ω and in the fracture F_{72}. The test with $c_\rho = 1.5$ yields similar results, so we have chosen to omit the corresponding plots. For each step m, we measure the minimum and maximum values (min–max $AR_E^\star$ area), the first and third quartile values (q1-q3 $AR_E^\star$ area), the median value (med $AR_E^\star$ curve), and the average value (avg $AR_E^\star$ curve). In the DFN, Figs. 15(a)–15(b) show that as the number of DOFs increases, all the average curves approach the median curves. This indicates an improvement in the quality of mesh elements at convergence. The maximum values of the DFN AR_E^{Rr} remain stable during the iterations because some fractures are almost untouched by the refinement process, where the solution is nearly constant (dead-end fractures). It is important to note that these poor-quality elements are in a limited number because the median and the mean curves are order of magnitude far from the maximum values of the plot. On the other hand, in the region where the refinement process is exploited, the effect of

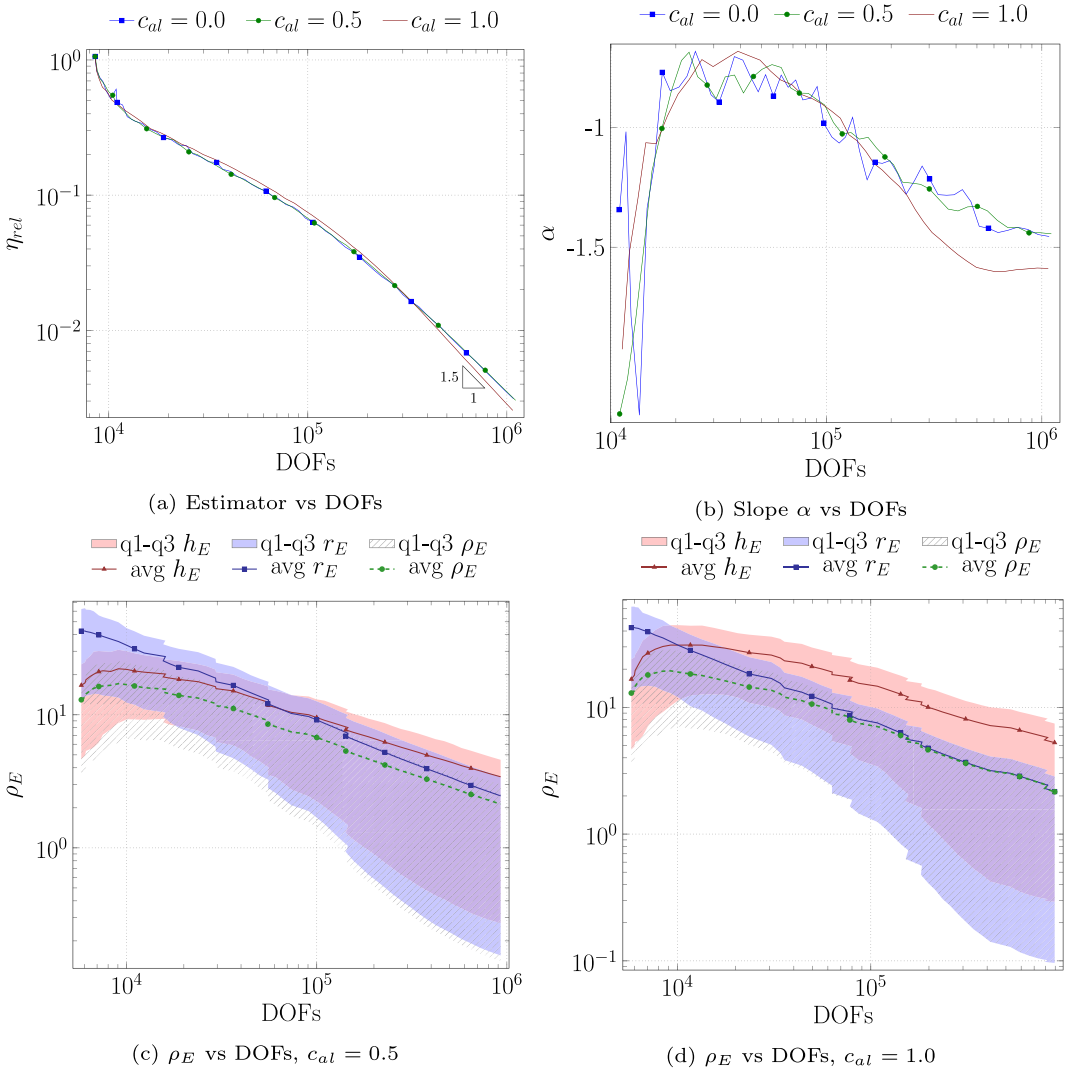
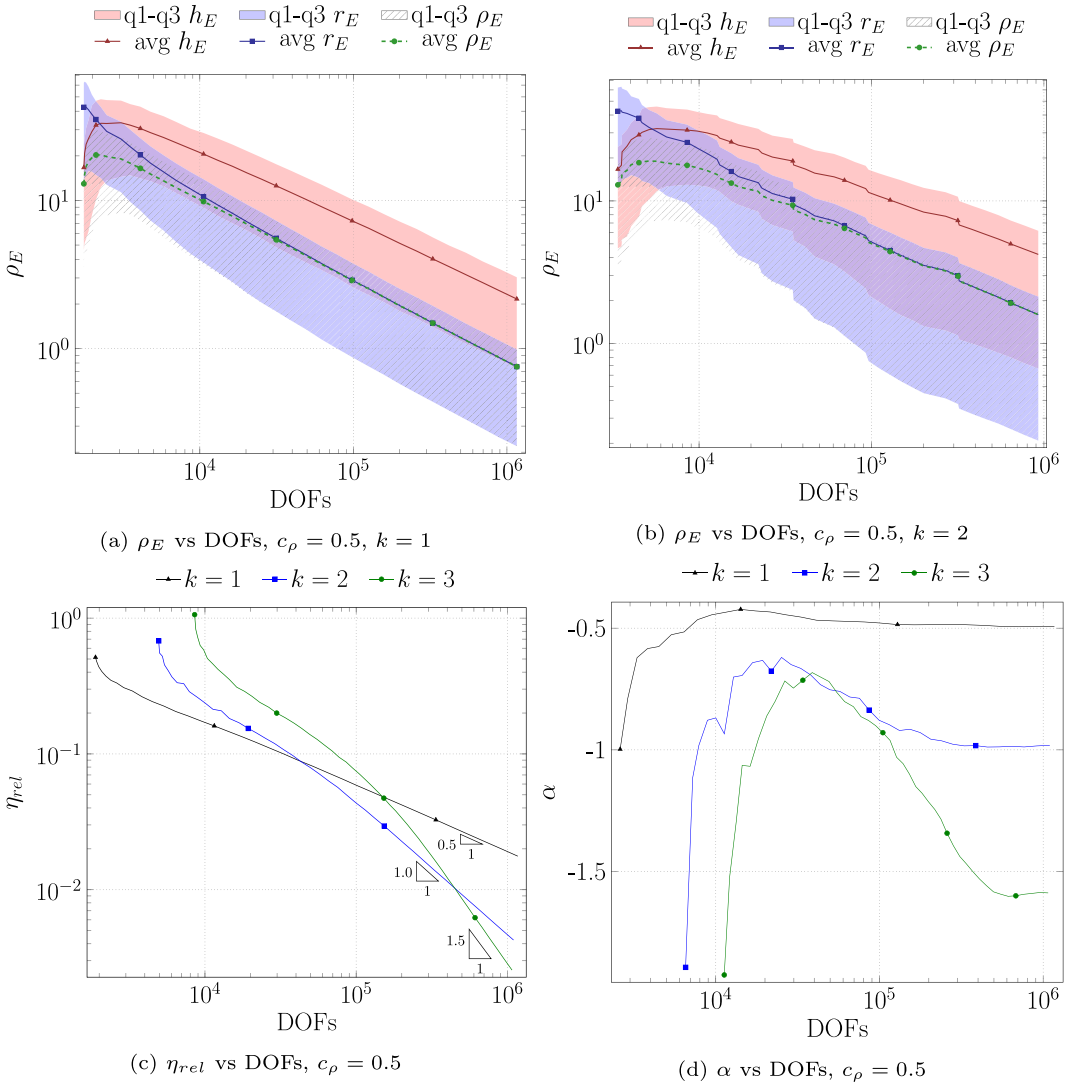


Fig. 16. Test 3 - c_{al} analysis, $c_\rho = 0.5$, $k = 3$.

the mesh quality improvement are visible. Indeed, for the local tessellation of fracture F_{72}, Figs. 15(c)–15(d) reveal that particularly noteworthy is the decay of the AR_E^{Rh} indicator, starting from a value of 90 and ending at a small maximum value lower than 5.

In Fig. 16, we report the analysis of the variation of the parameter c_{al} with $c_\rho = 0.5$. We do not analyse the case $c_\rho = 1.5$ since, as observed in the L-shape test, for this case the variation of c_{al} proves to be irrelevant. Moreover, we present only the case for VEM order $k = 3$, highlighting that the results for $k = 1$ and $k = 2$ are equivalent. Fig. 16(a) displays the convergence of the relative a-posteriori error estimator η_{rel} of Eq. (12) and in Fig. 16(b) we report the convergence rate α at iteration $m > 4$, for $c_{al} \in \{0.0, 0.5, 1.0\}$. As observed in the L-shape test, a value of $c_{al} = 1.0$ facilitates reaching the optimal rate faster than the other values. Once again, the rates for the other c_{al} values are sub-optimal due to the poor quality of the mesh during the refinement process. To support this observation, in Figs. 16(c)–16(d) we present the statistics for each iteration m, showing the evolution of the parameter ρ_E of Eq. (14), and the geometric quantities r_E and h_E. We measure the first and third quartile values (q1-q3 areas), and the average values (avg curves). The measurements confirm that when the length of the minimum edge of the mesh elements (h_E) is small compared to the inner radius (r_E), i.e. $\rho_E = h_E$, the VEM optimal convergence is not fulfilled.

Setting the optimal value of c_{al} as 1.0, Figs. 17(c) and 17(d) depict the convergence curves for $k \in \{1, 2, 3\}$. The graphs exhibit the successful convergence of the scheme to the optimal rates α for all orders. The optimal rate is achieved only after a number of mesh steps, increasing with higher VEM orders: $0.5 \cdot 10^5$, 10^5, and $0.5 \cdot 10^6$ DOFs for $k = 1$, $k = 2$, and $k = 3$, respectively. This behaviour is again closely linked to the poor quality of the initial DFN mesh, $\mathcal{T}_\Omega^0$, illustrated in Figs. 17(a)–17(b). These figures present for the orders $k = 1$ and $k = 2$ the same statistical analysis of quantities h_E, r_E, and ρ_E addressed for the case $k = 3$ in Fig. 16(d). When convergence rates α are sub-optimal (DOFs lower than 10^5) the average curve of ρ_E does not coincide with the



(a) ρ_E vs DOFs, $c_\rho = 0.5$, $k = 1$

(b) ρ_E vs DOFs, $c_\rho = 0.5$, $k = 2$

(c) η_{rel} vs DOFs, $c_\rho = 0.5$

(d) α vs DOFs, $c_\rho = 0.5$

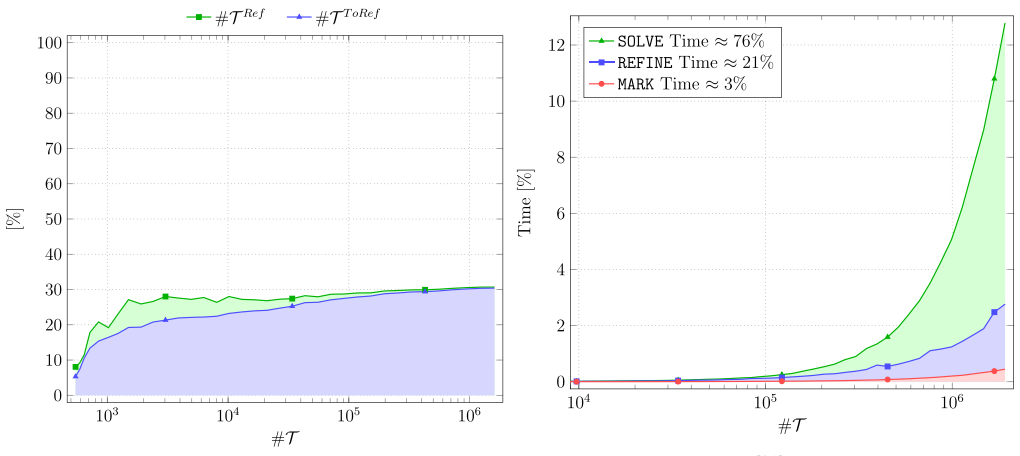
Fig. 17. Test 3 - Estimator η_{rel} analysis, $c_{al} = 1.0$.

average curve r_E. This discrepancy is more pronounced in real DFN applications where the high randomness of fractures leads to a poor quality discretization. Thus, a robust iterative scheme, such as the one presented, is crucial to obtain a good numerical solution with the minimum number of DOFs in the discretization.

In conclusion, we assess the computational impact of the proposed Algorithm 1 on the global adaptive scheme (3). Fig. 18(a) illustrates, for each refinement iteration m, the comparison between the number of marked cells $\mathcal{M}^m$ to refine ($\mathcal{T}_\Omega^{ToRef}$) and the number of refined cells by the algorithm ($\mathcal{T}_\Omega^{Ref}$). Notably, $\mathcal{T}_\Omega^{Ref}$ is higher or equal to $\mathcal{T}_\Omega^{ToRef}$ due to *refinement propagation* proposed in Line 12 of `REFINE` Algorithm 1. The additional marked elements ($\mathcal{T}_\Omega^{Ref} \setminus \mathcal{T}_\Omega^{ToRef}$ area) are minimal compared to the size of the initially selected elements. Moreover, the number of new extended elements approaches to zero after the initial steps. Additionally, Fig. 18(b) presents the time ratio employed for the `SOLVE`, the `MARK`, and the `REFINE` steps of the adaptive scheme (3). As expected, the `SOLVE` phase dominates (≈76%). Remarkably, the time allocated for the new `REFINE` phase aligns well with that measured in [11], despite the new algorithmic additions. It is noteworthy that both the `REFINE` and `MARK` steps maintain linear complexity with respect to the total number of mesh elements $\#\mathcal{T}_\Omega$.

6. Conclusions

We present a novel refinement algorithm to enhance the quality of a generic polygonal tessellation combined with the virtual element method. We show the effectiveness of the new approach in highly intricate domain for flow simulations in fractured media.



(a) $\#\mathcal{T}_\Omega^\star$ vs $\#\mathcal{T}_\Omega$ (b) Time [%] vs $\#\mathcal{T}_\Omega$

Fig. 18. Test 3 - Refinement Impact analysis, $c_\rho = 0.5$, $c_{al} = 1.0$, $k = 1$.

It is important to emphasize that the outlined strategy is applicable to a wide range of problems where polygonal meshes are beneficial.

This process is governed by two parameters, which control the shape and the quality of the final discretization elements. Through numerical examples, we prove the efficacy of the refinement process to improve the representation of the solution, against a starting high number of aligned edges and polygonal cells of varying size. This resilience is achieved by incorporating a new control mechanism for the number of the aligned edges. Moreover, the introduction of marked cell propagation in the refinement algorithm allows us to rapidly obtain a nearly triangular or quadrilateral mesh in few iterations. These innovative contributions do not compromise the computational time complexity of the method, which remains linear concerning the total number of mesh elements.

In conclusion, our numerical tests reveal optimal convergence rates for high VEM orders, even in challenging scenarios such as the real Discrete Fracture Network benchmark test with non-smooth solutions and initial poor mesh quality. This highlights the versatility of the proposed refinement algorithm across diverse applications.

CRediT authorship contribution statement

Stefano Berrone: Writing – original draft, Validation, Supervision, Investigation, Formal analysis, Data curation, Conceptualization. **Fabio Vicini:** Writing – original draft, Validation, Software, Investigation, Formal analysis, Data curation, Conceptualization.

Acknowledgements

The authors are member of the Gruppo Nazionale Calcolo Scientifico-Istituto Nazionale di Alta Matematica (GNCS-INdAM). This research has been partially supported by the European Union through project Next Generation EU, PRIN 2022 PNRR project P2022BH5CB_001 CUP:E53D23017950001 and by INdAM-GNCS, Italy Project CUP:E53C23001670001 and by the MUR PRIN, Italy project 20204LN5N5_003.

References

[1] L. Beirão da Veiga, F. Brezzi, L.D. Marini, A. Russo, Virtual element method for general second-order elliptic problems on polygonal meshes, Math. Models Methods Appl. Sci. 26 (04) (2016) 729–750.
[2] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L.D. Marini, A. Russo, Basic principles of virtual element methods, Math. Models Methods Appl. Sci. 23 (01) (2013) 199–214.
[3] S. Congreve, P. Houston, I. Perugia, Adaptive refinement for hp-Version Trefftz discontinuous Galerkin methods for the homogeneous Helmholtz problem, Adv. Comput. Math. 45 (2019) 361–393.
[4] S. Congreve, P. Houston, Two-grid hp-version discontinuous Galerkin finite element methods for quasilinear elliptic PDEs on agglomerated coarse meshes, Adv. Comput. Math. 48 (2022) 54.
[5] P. Antonietti, E. Manuzzi, Refinement of polygonal grids using convolutional neural networks with applications to polygonal discontinuous Galerkin and virtual element methods, J. Comput. Phys. 452 (2022) 110900.
[6] A. Cangiani, E.H. Georgoulis, T. Pryer, O.J. Sutton, A posteriori error estimates for the virtual element method, Numer. Math. 137 (4) (2017) 857–893.
[7] S. Berrone, A. Borio, A residual a posteriori error estimate for the virtual element method, Math. Models Methods Appl. Sci. 27 (08) (2017) 1423–1458.
[8] L. Beirão da Veiga, C. Canuto, R.H. Nochetto, G. Vacca, M. Verani, Adaptive VEM: Stabilization-free a posteriori error analysis and contraction property, SIAM J. Numer. Anal. 61 (2) (2023) 457–494.
[9] C. Canuto, D. Fassino, Higher-order adaptive virtual element methods with contraction properties, Math. Eng. 5 (6) (2023) 1–33.

[10] S. Berrone, A. Borio, A. D'Auria, Refinement strategies for polygonal meshes applied to adaptive VEM discretization, Finite Elem. Anal. Des. 186 (2021) 103502.
[11] S. Berrone, D. A., A new quality preserving polygonal mesh refinement algorithm for polygonal element methods, Finite Elem. Anal. Des. 207 (2022) 103770.
[12] G. Pichot, P. Laug, J. Erhel, R. Le Goc, C. Darcel, P. Davy, J. de Dreuzy, Simulations in large tridimensional discrete fracture networks (DFN): II. Flow simulations, in: MASCOT 2018-15th IMACS/ISGG Meeting on Applied Scientific Computing and Tools, 2018.
[13] S. Berrone, S. Scialò, F. Vicini, Parallel meshing, discretization, and computation of flow in massive discrete fracture networks, SIAM J. Sci. Comput. 41 (4) (2019) C317–C338.
[14] T. Sorgente, F. Vicini, S. Berrone, S. Biasotti, G. Manzini, M. Spagnuolo, Mesh quality agglomeration algorithm for the virtual element method applied to discrete fracture networks, in: Mesh Quality Agglomeration Algorithm for the Virtual Element Method Applied to Discrete Fracture Networks, Vol. 60, 2023.
[15] B. Ahmad, A. Alsaedi, F. Brezzi, L. Marini, A. Russo, Equivalent projectors for virtual element methods, Comput. Math. Appl. 66 (3) (2013) 376–391.
[16] S. Berrone, A. Borio, A. D'Auria, S. Scialò, F. Vicini, A robust VEM-based approach for flow simulations in poro-fractured media, Math. Models Methods Appl. Sci. 31 (14) (2021) 2855–2885.
[17] T. Sorgente, S. Biasotti, G. Manzini, M. Spagnuolo, The role of mesh quality and mesh quality indicators in the virtual element method, Adv. Comput. Math. 48 (3) (2021).
[18] L. Beirão da Veiga, G. Vacca, Sharper error estimates for virtual elements and a bubble-enriched version, SIAM J. Numer. Anal. 60 (4) (2022) 1853–1878.
[19] R.H. Nochetto, A. Veeser, Primer of adaptive finite element methods, in: Multiscale and Adaptivity: Modeling, Numerics and Applications: C.I.M.E. Summer School, Cetraro, Italy 2009, Editors: Giovanni Naldi, Giovanni Russo, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 125–225.
[20] M. Fernando Benedetto, S. Berrone, S. Scialò, A globally conforming method for solving flow in discrete fracture networks using the virtual element method, Finite Elem. Anal. Des. 109 (2016) 23–36.
[21] L. Beirão da Veiga, C. Lovadina, A. Russo, Stability analysis for the virtual element method, Math. Models Methods Appl. Sci. 27 (13) (2017) 2557–2594.
[22] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM J. Numer. Anal. 33 (3) (1996) 1106–1124.
[23] P. Grisvard, Elliptic Problems in Nonsmooth Domains, Society for Industrial and Applied Mathematics.
[24] M. Fernando Benedetto, S. Berrone, A. Borio, S. Pieraccini, S. Scialò, A hybrid mortar virtual element method for discrete fracture network simulations, J. Comput. Phys. 306 (2016) 148–166.