POLITECNICO DI TORINO
Repository ISTITUZIONALE

Refinement strategies for polygonal meshes applied to adaptive VEM discretization

*Original*
Refinement strategies for polygonal meshes applied to adaptive VEM discretization / Berrone, Stefano; Borio, Andrea; D'Auria, Alessandro. - In: FINITE ELEMENTS IN ANALYSIS AND DESIGN. - ISSN 0168-874X. - STAMPA. - 186:(2021), pp. 1-16. [10.1016/j.finel.2020.103502]

*Availability:*
This version is available at: 11583/2858568 since: 2025-02-14T17:25:03Z

*Publisher:*
Elsevier

*Published*
DOI:10.1016/j.finel.2020.103502

*Terms of use:*

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

*Publisher copyright*
Elsevier postprint/Author's Accepted Manuscript

© 2021. This manuscript version is made available under the CC-BY-NC-ND 4.0 license
http://creativecommons.org/licenses/by-nc-nd/4.0/.The final authenticated version is available online at:
http://dx.doi.org/10.1016/j.finel.2020.103502

(Article begins on next page)

16 May 2025

# Refinement strategies for polygonal meshes applied to adaptive VEM discretization.[☆]

Stefano Berrone[a,b,*], Andrea Borio[a,b], Alessandro D'Auria[a,b]

[a] *Dipartimento di Scienze Matematiche, Politecnico di Torino*
*Corso Duca degli Abruzzi 24, Torino, 10129, Italy*
[b] *Member of the INdAM research group GNCS*

**Abstract**

In the discretization of differential problems on complex geometrical domains, discretization methods based on polygonal and polyhedral elements are powerful tools. Adaptive mesh refinement for such kind of problems is very useful as well and states new issues, here tackled, concerning good quality mesh elements and reliability of the simulations. In this paper we propose several new polygonal refinement strategies and numerically investigate the quality of the meshes generated by an adaptive mesh refinement process, as well as optimal rates of convergence with respect to the number of degrees of freedom. Among the several possible problems in which these strategies can be applied, here we have considered a geometrically complex geophysical problem as test problem that naturally yields to a polygonal mesh and tackled it by the Virtual Element Method. All the adaptive strategies here proposed, but the "Trace Direction strategy", can be applied to any problem for which a polygonal element method can be useful and any numerical method based on polygonal elements and can generate good quality isotropic mesh elements.

*Keywords:* Mesh adaptivity, Polygonal mesh refinement, Virtual Element Method, Simulations in complex geometries, A posteriori error estimates
*2010 MSC:* 65N30, 65N50, 68U20, 86-08, 86A05

[☆]This research has been supported by INdAM-GNCS Projects 2018 and 2019, by the MIUR project PRIN 201744KLJL_004, and by the MIUR project "Dipartimenti di Eccellenza 2018-2022" (CUP E11G18000350001). Computational resources were partially provided by HPC@POLITO (`http://hpc.polito.it`) and SmartData@POLITO (`http://smartdata.polito.it`).

[*]Corresponding author
*Email addresses:* `stefano.berrone@polito.it` (Stefano Berrone), `andrea.borio@polito.it` (Andrea Borio), `alessandro.dauria@polito.it` (Alessandro D'Auria)

## 1. Introduction

In last years, a growing interest has arisen for the development of numerical methods for the solution of partial differential equations using general polygonal meshes. These methods are well suited for handling domains featuring geometrical complexities that can yield situations where the generation of good quality conforming triangular meshes can be particularly expensive or even unfeasible. On the other hand, the use of polygonal and polyhedral elements in conjunction with adaptive mesh refinement states new problems. In this paper we investigate several refinement strategies for polygonal meshes.

As model problem to test the different refinement strategies we consider the simulation of flow in a fractured medium. When fractures are represented as intersecting planar polygons, this problem naturally yields to polygonal meshes as first step of mesh generation (computation of the sub-fractures) and in this context the proposed refinement strategies can be part of the mesh production process and not only of the mesh refinement process, more details are provided in Section 4. Nevertheless we want to stress that the proposed refinement strategies can be applied to any problem for which a polygonal mesh can be valuable.

Flow simulation in underground fractured media is a key aspect in many applications such as aquifers monitoring, nuclear waste disposal, $CO_2$ geological storage or control of contaminant dispersion in the subsoil. In those situations where the rock matrix can be considered perfectly impervious, the Discrete Fracture Network (DFN) model is often used, representing fractures as planar polygons randomly intersected in the 3D space [1, 2, 3, 4]. These kind of domains usually present many geometrical challenges, in particular when a conforming mesh is required, for example in order to apply standard finite element approaches. To circumvent these difficulties, many approaches have been devised, relying on domain decomposition techniques [5, 6], or devising particular meshing strategies [7, 8, 9, 10], or using extensions of the finite element method [11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. In [22, 23], a PDE-constrained optimization approach was devised, enabling for the use of completely non-conforming meshes, where finite element, extended finite element [24] or virtual element spaces [25] can be used. Here we follow the approach based on the Virtual Element Method (VEM) [26, 27, 28, 29] first introduced for DFN flow simulations in [25] and extended in [30, 31, 32].

The huge cost of large scale simulations on the scale of a geological basin raises the issue of efficiency. Moreover, due to the large uncertainty in geometrical configurations and hydrogeological parameters a stochastic approach is advisable yielding to the need of many simulations. Beside efficiency, reliability of these simulations is of crucial importance being connected with risk analysis of human activities. Efficiency and reliability strongly affect the application of uncertainty quantification techniques for the estimation of relevant quantities of interest [33, 34]. For these reasons adaptivity plays a key role both on the side of reliability and efficiency.

Although we focus on the simulation of flow in DFN using VEM, the refinement strategies explored here are suitable for any kind of method relying

on polygonal meshes, except for the so-called "Trace Direction" refinement, that is particularly suited for DFN problems taking into account the behaviour of the solution approaching the traces. The approach here followed is based on isotropic *a posteriori* error estimates and aims at generating good quality isotropic polygonal elements. Anisotropic estimates and mesh refinement strategies are currently under investigation in [35]. Other a posteriori error estimates and mesh refinement strategies can be found in [36, 37, 38]. In [37], a posteriori error estimates are applied to curved virtual elements, in [36] and [38] hp virtual elements are considered, the a posteriori error estimate in [38] being based on equilibrated fluxes.

In Section 2 we introduce the DFN model used to test the refinement strategies and define some useful notations. In Section 3 we introduce its Virtual Element discretization, we define the error estimators and state the *a posteriori* error estimate. In Section 4 we show an algorithm to generate a globally conforming polygonal mesh for DFNs. This algorithm can be also applied to generate conforming polygonal mesh for 2D domains with embedded 1D subdomains. In Section 5 we describe the adaptive algorithm, and some polygonal refinement strategies that preserve the global conformity of the mesh. In Section 6 we validate their application to some test cases with known solution, and, in Section 7, we analyse some results on a more realistic DFN. In Section 8 we draw conclusions concerning the refinement strategies and the quality of the meshes that are produced.

## 2. Discrete Fracture Networks

Let us consider a set of open convex planar polygonal fractures $F_i \subset \mathbb{R}^3$ with $i = 1, ..., N$, with boundary $\partial F$. A DFN is $\Omega = \bigcup_i F_i$, with boundary $\partial\Omega$. Even though the fractures are planar, their orientations in space are arbitrary, such that $\Omega$ is a 3D set. The set $\Gamma_D \subset \partial\Omega$ is where Dirichlet boundary conditions are imposed, and we assume $\Gamma_D \neq \emptyset$, whereas $\Gamma_N = \partial\Omega \setminus \Gamma_D$, is the portion of the boundary with Neumann boundary conditions. Dirichlet and Neumann boundary conditions are prescribed by the functions $h^D \in \mathrm{H}^{\frac{1}{2}}(\Gamma_D)$ and $g^N \in \mathrm{H}^{-\frac{1}{2}}(\Gamma_N)$ on the Dirichlet and Neumann part of the boundary, respectively. We further set $\Gamma_{iD} = \Gamma_D \cap \partial F_i$, $\Gamma_{iN} = \Gamma_N \cap \partial F_i$, and $h_i^D = {h^D}_{|\Gamma_{iD}}$ and $g_i^N = {g^N}_{|\Gamma_{iN}}$. The set $\mathcal{S}$ collects all the traces, i.e. the intersections between fractures, and each trace $S \in \mathcal{S}$ is given by the intersection of exactly two fractures, $S = \bar{F}_i \cap \bar{F}_j$, such that there is a one to one relationship between a trace $S$ and a couple of fracture indices $\{i, j\} = \mathcal{I}(S)$. We will also denote by $\mathcal{S}_i$ the set of traces belonging to fracture $F_i$.

Subsurface flow is governed by the gradient of the hydraulic head $H = \mathcal{P} + \zeta$, where $\mathcal{P} = p/(\varrho g)$ is the pressure head, $p$ is the fluid pressure, $g$ is the gravitational acceleration constant, $\varrho$ is the fluid density and $\zeta$ is the elevation.

We define the following functional spaces:

$$V_i = \mathrm{H}^1_0(F_i) = \left\{ v \in \mathrm{H}^1(F_i) : v_{|\Gamma_{iD}} = 0 \right\},$$
$$V_i^D = \mathrm{H}^1_{\mathrm{D}}(F_i) = \left\{ v \in \mathrm{H}^1(F_i) : v_{|\Gamma_{iD}} = h_i^D \right\},$$

and

$$V = \left\{ v \colon v_{|F_i} \in V_i,\ \forall i = 1, \dots, N, \gamma_S(v_{|F_i}) = \gamma_S(v_{|F_j}),\ \forall S \in \mathcal{S}_i,\ \{i,j\} = \mathcal{I}(S) \right\},$$

$$V^D = \left\{ v \colon v_{|F_i} \in V_i^D,\ \forall i = 1, \dots, N, \gamma_S(v_{|F_i}) = \gamma_S(v_{|F_j}),\ \forall S \in \mathcal{S}_i,\ \{i,j\} = \mathcal{I}(S) \right\},$$

where $\gamma_S$ is the trace operator onto $S$. In order to formulate the DFN flow problem, let $a_i \colon V_i^D \times V_i \to \mathbb{R}$ be defined as

$$a_i(w, v) = \left( \mathcal{K}_i \nabla \left( w_{|F_i} \right), \nabla \left( v_{|F_i} \right) \right)_{F_i} \quad \forall w \in V^D,\ v \in V,$$

where $\mathcal{K}_i$ is the fracture transmissivity tensor, that we assume to be constant on each fracture. Let us denote by $H \in V^D$ the hydraulic head on the DFN and by $H_i \in V_i^D$ its restriction to the fracture $F_i$ and by $v \in V$ the test functions. Assuming the hydraulic head modelled by the Darcy law, the whole problem on the DFN is: find $H \in V^D$ such that $\forall v \in V$

$$\sum_{i=1}^{N} a_i(H_i, v) = \sum_{i=1}^{N} (f_i, v)_{F_i} + \left\langle g_i^N, v_{|\Gamma_{N_i}} \right\rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{N_i}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{N_i})}. \tag{1}$$

## 3. Virtual Element discretization

In this section we describe the Virtual Element discretization of equation (1) assuming a globally conforming polygonal mesh is given on the DFN. A globally conforming mesh is a polygonal mesh on each fractures such that polygon edges match exactly at traces.

Let $\mathcal{T}_\delta$ be a globally conforming polygonal mesh of $\Omega$ fulfilling the regularity requirements needed by the Virtual Element method [28], $E \in \mathcal{T}_\delta$ be any polygon of this tessellation. Let $\mathbb{P}_k(E)$ be the space of polynomials of degree $\leq k$ defined on $E$. To define the discrete functional space on $E$, we introduce the $\mathrm{H}^1$-orthogonal projector $\Pi^\nabla_{k,E} \colon \mathrm{H}^1(E) \to \mathbb{P}_k(E)$ such that

$$\left( \nabla \left( v - \Pi^\nabla_{k,E} v \right), \nabla p \right)_E = 0, \forall p \in \mathbb{P}_k(E)$$

and

$$\begin{cases} \left( \Pi^\nabla_{k,E} v, 1 \right)_{\partial E} = (v, 1)_{\partial E} & \text{if } k = 1, \\ \left( \Pi^\nabla_{k,E} v, 1 \right)_E = (v, 1)_E & \text{if } k \geq 2. \end{cases}$$

The Virtual Element space of order $k \in \mathbb{N}$ on $E$ is defined as

$$V_\delta^E = \Big\{ v \in \mathrm{H}^1(E) : \Delta v \in \mathbb{P}_k(E)\,,\, v \in \mathbb{P}_k(e)\ \forall e \subset \partial E, \gamma^{\partial E}(v) \in C^0(\partial E)\ , \\ (v,p)_E = \left(\Pi^\nabla_{k,E} v, p\right)_E\ \forall p \in \mathbb{P}_k(E)\,/\mathbb{P}_{k-2}(E) \Big\}\ ,$$

where $\mathbb{P}_k(E)\,/\mathbb{P}_{k-2}(E)$ denotes the subspace of $\mathbb{P}_k(E)$ containing polynomials that are $\mathrm{L}^2(E)$-orthogonal to $\mathbb{P}_{k-2}(E)$. Furthermore, let us denote by $\mathcal{T}_{\delta,i}$ the restriction of $\mathcal{T}_\delta$ to fracture $F_i$. The Virtual Element space on $\mathcal{T}_{\delta,i}$ is

$$V_{\delta,i} = \left\{ v \in C^0(F_i) : v \in V_\delta^E\ \forall E \in \mathcal{T}_{\delta,i} \right\}\ .$$

Since $\mathcal{T}_\delta$ is globally conforming, we can define the global discrete spaces as

$$V_\delta = \left\{ v \in V \colon v_{|F_i} \in V_{\delta,i} \right\}\ , \qquad V_\delta^D = \left\{ v \in V^D \colon v_{|F_i} \in V_{\delta,i} \right\}\ .$$

A function in the above space is uniquely identified by the following set of degrees of freedom:

1. the values at the vertices of the polygons;
2. if $k \geq 2$, for each edge $e$ of the mesh, the value of $v$ at $k-1$ internal points of $e$;
3. if $k \geq 2$, for each $E \in \mathcal{T}_\delta$, the scaled moments $\frac{1}{|E|}\left(v, m_{\boldsymbol{\alpha}}\right)_E$ for all the scaled monomials $m_{\boldsymbol{\alpha}}$, with $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$, $|\boldsymbol{\alpha}| = \alpha_1 + \alpha_2 \leq k-2$, such that
$$\forall (x,y) \in E, \quad m_{\boldsymbol{\alpha}}(x,y) := \frac{(x - x_E)^{\alpha_1}(y - y_E)^{\alpha_2}}{h_E^{\alpha_1+\alpha_2}}\ ,$$
being $(x_E, y_E)$ the centroid of the cell and $h_E$ its diameter.

For any element $E \in \mathcal{T}_\delta$, given a function $v_\delta \in V_\delta^E$, it can be seen [26, 39] that the values of its degrees of freedom are uniquely defined by its $\mathrm{L}^2(E)$-orthogonal projection on $\mathbb{P}_{k-1}(E)$, denoted by $\Pi^0_{k-1,E} v_\delta$, and the orthogonal projection of its gradient on $\mathbb{P}_{k-1}(E) \times \mathbb{P}_{k-1}(E)$, denoted by $\Pi^0_{k-1,E} \nabla v_\delta$. A basis of the local VEM space is defined implicitly as the set of functions that are Lagrangian with respect to the degrees of freedom.

To discretize (1) by the Virtual Element method we suppose to know, for each $E \in \mathcal{T}_{\delta,i}$, $i = 1, \ldots, N$, a bilinear form $S \colon V_\delta^E \times V_\delta^E \to \mathbb{R}$ such that, $\forall v_\delta \in V_\delta^E \cap \ker \Pi^\nabla_{k,E}$,

$$\exists c_*\,, c^* > 0 \colon c_* \left(\mathcal{K}_i \nabla v_\delta, \nabla v_\delta\right)_E \leq S^E\left(v_\delta, v_\delta\right) \leq c^* \left(\mathcal{K}_i \nabla v_\delta, \nabla v_\delta\right)_E\ .$$

With the above assumption, we define, for each $E \in \mathcal{T}_{\delta,i}$, $i = 1, \ldots, N$, the bilinear form $a_\delta^E \colon V_\delta^E \times V_\delta^E \to \mathbb{R}$ such that, $\forall u_\delta, v_\delta \in V_\delta$,

$$a_\delta^E\left(u_\delta, v_\delta\right) = \left(\mathcal{K}_i \Pi^0_{k-1} \nabla u_\delta, \Pi^0_{k-1} \nabla v_\delta\right)_E + \mathcal{K}_i S^E\left(\left(I - \Pi^\nabla_k\right) u_\delta, \left(I - \Pi^\nabla_k\right) v_\delta\right)\ ,$$

and the global bilinear form $a_\delta \colon V_\delta^D \times V_\delta \to \mathbb{R}$ such that

$$a_\delta\left(u_\delta, v_\delta\right) = \sum_{i=1}^{N} \sum_{E \in \mathcal{T}_{\delta,i}} a_\delta^E\left(u_{\delta|F_i}, v_{\delta|F_i}\right).$$

Finally, the Virtual Element discretization of (1) is: find $H_\delta \in V_\delta^D$ such that, $\forall v_\delta \in V_\delta$

$$a_\delta(H_\delta, v_\delta) = \sum_{i=1}^{N} \sum_{E \in \mathcal{T}_{\delta,i}} \left(f_i, \Pi^0_{k-1,E}\left(v_{|F_i}\right)\right)_E + \left\langle g_i^N, v_{|\Gamma_{N_i}} \right\rangle_{\mathrm{H}^{-\frac{1}{2}}\left(\Gamma_{N_i}\right), \mathrm{H}^{\frac{1}{2}}\left(\Gamma_{N_i}\right)} .$$

We remark that the continuity conditions are automatically satisfied by the definition of the functional space and the degrees of freedom, viable because $\mathcal{T}_\delta$ is a globally conforming mesh.

In [32], a residual a posteriori estimator was derived for the Poisson problem proving the equivalence between the estimator and the error with respect to a suitable polynomial projection of the VEM solution. The extension of this estimate to the case of a globally conforming discretization of the Laplace problem on a DFN is quite straightforward. Let us define the following error measure:

$$|||v||| = \sup_{w \in V} \frac{\sum_{i=1}^{N} a_i(v_i, w)}{\left(\sum_{i=1}^{N} \left\| \sqrt{\mathcal{K}} \nabla w \right\|^2_{F_i}\right)^{\frac{1}{2}}},$$

then, we denote $H_\delta^\pi = \Pi_k^\nabla(H_\delta)$ and define

$$\begin{aligned} est_\delta^2 = \sum_{i=1}^{N} \Bigg( & \sum_{E \in \mathcal{T}_{\delta,i}} \frac{h_E^2}{\mathcal{K}_i} \left\| \Pi^0_{k-1} f + \mathcal{K}_i \Delta H^\pi_{\delta i} \right\|^2_E + \sum_{e \in \mathcal{E}^{\mathrm{int}}_{\delta,i}} \frac{h_e}{\mathcal{K}_i} \left\| \left[\!\left[ \mathcal{K}_i \frac{\partial H^\pi_{\delta i}}{\partial \mathbf{n}} \right]\!\right]_e \right\|^2_e + \\ & + \sum_{e \in \mathcal{E}^{N}_{\delta,i}} \frac{h_e}{\mathcal{K}_i} \left\| \left[\!\left[ \mathcal{K}_i \frac{\partial H^\pi_{\delta i}}{\partial \mathbf{n}} \right]\!\right]_e - g_i^N \right\|^2_e + \sum_{E \in \mathcal{T}_\delta} \frac{h_E^2}{\mathcal{K}_i} \left\| f - \Pi^0_{k-1} f \right\|^2_E \Bigg) \\ & + \sum_{\substack{S \in \mathcal{S}: \\ \mathcal{I}(S) = \{i,j\}}} \left( \sum_{e \in \mathcal{E}_{\delta,S}} \frac{h_e}{\min\{\mathcal{K}_i, \mathcal{K}_j\}} \left\| \left[\!\left[ \mathcal{K}_i \frac{\partial H^\pi_{\delta i}}{\partial \mathbf{n}} \right]\!\right]_e + \left[\!\left[ \mathcal{K}_j \frac{\partial H^\pi_{\delta j}}{\partial \mathbf{n}} \right]\!\right]_e \right\|^2_e \right), \end{aligned}$$

where $\forall i = 1, \dots N$, $\mathcal{E}^{\mathrm{int}}_{\delta,i}$ is the set of edges of $\mathcal{T}_{\delta,i}$ such that, $\forall e \in \mathcal{E}^{\mathrm{int}}_{\delta,i}$, $e \cap S = \emptyset$ $\forall S \in \mathcal{S}_i$ and $e \cap \Gamma_{N_i} = \emptyset$, $\mathcal{E}^N_{\delta,i}$ is the set of edges $e$ of $\mathcal{T}_{\delta,i}$ such that $e \cap \Gamma_i^N \neq \emptyset$ and $\forall S \in \mathcal{S}$, $\mathcal{E}_{\delta,S}$ is the set of edges $e$ of $\mathcal{T}_\delta$ such that $e \cap S \neq \emptyset$. Then, there exist two constants $c, C > 0$ independent of $\delta$ such that

$$c \cdot est_\delta \leq |||H - H_\delta^\pi||| \leq C \cdot est_\delta . \tag{2}$$

In view of an adaptive approach, we define, for each cell $E$, a local estimator $est_{\delta,E}$, such that the estimators defined on the edges are split among neighbouring cells according to their areas.

## 4. DFN Minimal mesh construction

In this section we introduce the strategy used for the construction of the initial coarse polygonal mesh on the DFNs. This mesh is obtained by the construction of convex polygons representing sub-fractures, i.e. portion of fractures

not crossed by traces that can have traces or portion of traces only on the boundary. Given a fracture and the set of its traces there exist many partitions of the fracture in sub-fractures with a different number of sub-fractures and different quality of the produced elements. The construction of the minimal mesh is a complex trade off between the number of the elements and the quality of the produced elements. In our approach we aim at limiting the number of elements. Improvement of the mesh quality is transferred to a following suitable refining strategy.

The approach we follow is an iterative splitting of the leaves of a tree structure. We start from the original fracture that is the root cell of the structure, then we select a trace and we split the given cell along the trace or along a segment overlapped to the trace extending the trace contained in the cell up to the edges/points of the cell producing two children.

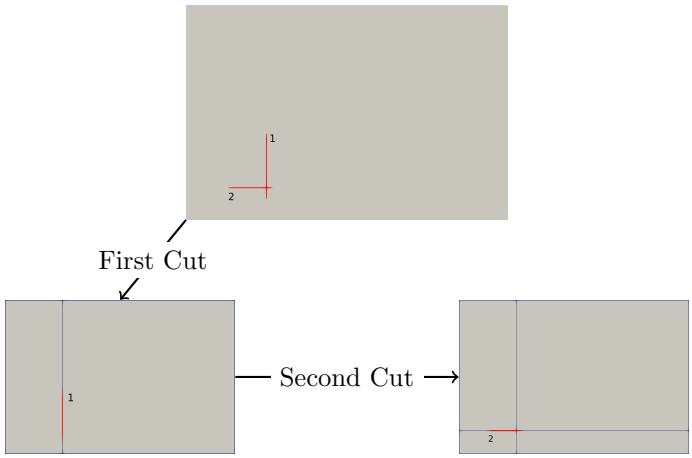



Figure 1: Minimal mesh generation on a fracture with two traces

Then we proceed iteratively choosing a new trace and cutting the leaves of the tree with the selected trace, exclusively in the case the trace is intersecting the internal part of the cell. At each iteration each cell can be split in two children cells or can be modified in a unique child cell that is the same polygon with one of the edges split in two aligned sub-edges in order to guarantee conformity of the global mesh. The algorithm is sketched in Algorithm 1. In Figure 1 we show the generation of a minimal polygonal mesh on a fracture with two traces. In this simple example two cuts to generate a polygonal mesh conforming to the traces are required. With the first cut we split the fracture along the vertical segment overlapped to the vertical trace, and with the second cut we split the two generated polygons (the two leaves of the current tree) along the segments overlapped to the horizontal trace. In Figure 2 we show the tree structure resulting from the first cut (level one) and the second cut (level

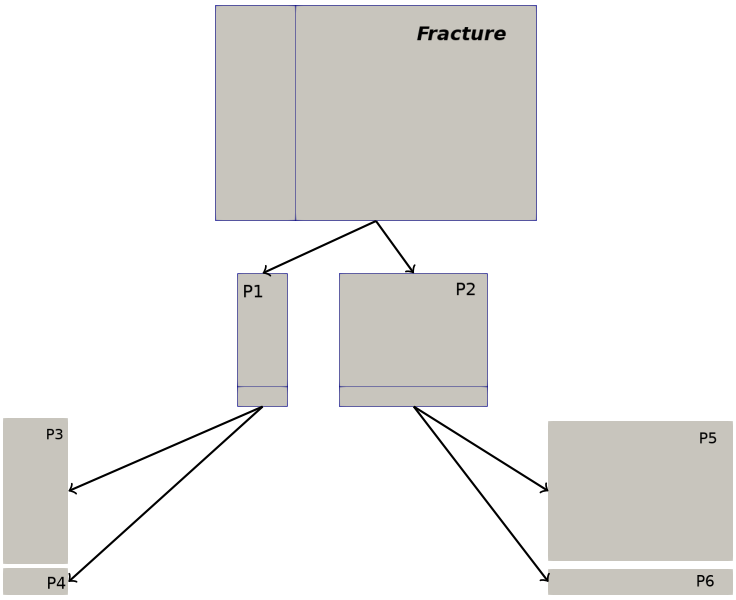



Figure 2: Second Cut

two). Each node in the tree structure represents a cell (or a temporary cell), the leaves represent the cells in the final mesh. The process starts setting the fracture as the root node of the tree (Fracture label in Figure 2). Taken the first trace, the intersections between the fracture cell and the segment generated by the trace one are computed. The intersection generates two cells, P1 and P2, that are inserted in the tree structure as children nodes of the root. Then we consider the second trace, all the leaves cells crossed by the trace are found (in this case both the level-one cells) and the intersections are computed. The intersections between the segment passing through the trace two and the cell P1 generates the cells P3 and P4, and the cell P2 generates the cells P5 and P6. The cells P3 and P4 and the cells P5 and P6 are inserted as children nodes of P1 and P2, respectively.

After the cut, all the leaves cells are updated adding the tips of the traces as vertices, introducing aligned edges and generating the minimal polygonal mesh on the fracture conforming to the traces.

In order to control the number of cells produced by the algorithm we cut the current cells with a suitable order of traces. We start considering the traces that cross the fracture intersecting two boundary edges of the fracture. Then we continue considering the remaining traces from the longest to the shortest. Considering the traces in this order usually yields to a smaller number of cells.

**Algorithm 1** Minimal mesh

Given a fracture and the set of traces create a tree structure with the fracture as root cell

```
for All the traces do
  for All the leaves cells of the tree do
    Compute the intersection of the trace with the cell
    if There is an intersection with the internal part of the cell then
      Split the cell in two children cells
    end if
    Update the neighbouring cell with the new edges
  end for
end for
```

## 5. Refinement and Marking algorithms

In this section we briefly introduce the algorithms used for marking the cells with largest estimators to be refined in order to reduce the discretization error and the different algorithms tested for the refinement of polygonal cells. This refinement step is part of the usual refinement process SOLVE-ESTIMATE-MARK-REFINE [40].

### *5.1. Marking Strategy*

The marking strategy (see Algorithm 2) of the cells to be refined is simply based on the selection of all the cells with the largest error estimators. We mark the cells starting from those with largest estimators $est^2_{\delta,E}$ up to when the cumulative error estimator $est^2_{\delta,\mathcal{T}_{\delta,mark}}$ of the marked cells is a given ratio $C$ of the total error estimator $est^2_{\delta,\mathcal{T}_\delta} = \sum_{E\in\mathcal{T}_\delta} est^2_{\delta,E}$. In this algorithm we accept the sorting cost for the estimator vector [40] in order to maximally contain the refinement iterations that, in practical applications to large scale DFN simulations, can be quite expensive in the last refinement steps.

### *5.2. Refinement algorithms*

In this section we introduce four different refinement algorithms used for cutting marked convex cells in two convex sub-cells. All the algorithms are based on a similar approach and differ for the choice of the cutting direction. The common approach is described in the Algorithm 3 and the four different approaches differ for the Step 3 of Algorithm 3.

In the following we compare four different refinement options for choosing the cutting direction denoted Maximun Momentum (MaxMom), Trace Direction (TrDir), Maximum Number of Points (MaxPnt) and Maximum Edge (MaxEdg). At the Step 8 of the Algorithm 3 we accept to slightly modify the chosen cutting direction in order to avoid the proliferation of edges and vertices of the new cells and consequently of VEM degrees of freedom that do not efficiently increase the

**Algorithm 2** Cells marking algorithm

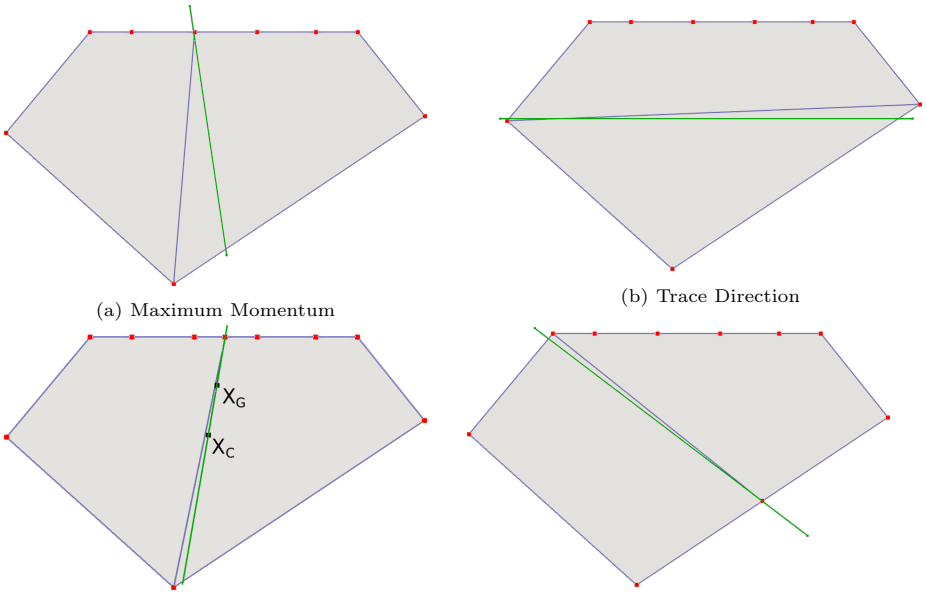
Given a convex polygon

Compute the cell error estimators $est^2_{\delta,E}$ and save them in a vector $Vest^2_\delta$.
Compute the total error estimator $est^2_{\delta,\mathcal{T}_\delta}$
Sort the cell estimators vector $Vest^2_\delta$
Choose a constant $0 < C < 1$
$i = 0$
$est^2_{\delta,\mathcal{T}_{\delta,mark}} = est^2_{\delta,\mathcal{T}_{\delta,mark}} + Vest^2_\delta(i)$
**repeat**
  $i++$
  $est^2_{\delta,\mathcal{T}_{\delta,mark}} = est^2_{\delta,\mathcal{T}_{\delta,mark}} + Vest^2_\delta(i)$
**until** $est^2_{\delta,\mathcal{T}_{\delta,mark}} \leq C * est^2_{\delta,\mathcal{T}_\delta}$
Mark the cells corresponding to the first $i$ positions of $Vest^2_\delta$

**Algorithm 3** Refinement algorithm for convex polygons

Given a convex polygon

Check aspect ratio (AR)
Compute the centroid $\mathbf{X}_c$
Choose the cutting direction
Build a straight line passing from the centroid with the chosen direction
Choose the collapsing tolerance CollapseToll
**for** Each edges of the cell **do**
  **if** There is an intersection **then**
    **if** The intersection point is near to the begin/end point of the edge according to the chosen tolerance CollapseToll **then**
      Change the cutting direction and set the intersection point as the begin/end point of the edge
    **else**
      The intersection point will be a new point in the mesh
    **end if**
  **end if**
**end for**
Create the two children cells
Update the neighbourhood




(a) Maximum Momentum

(b) Trace Direction

(c) Maximum Number of Points

(d) Maximum Edge

Figure 3: Different strategies for the selection of the cutting direction.

quality of the solution and to avoid the generation of very small edges that may induce stability problems [29, 41].

We collapse the points given by the intersection of the cutting direction with an edge to existing vertices of the cell when the distance of the intersection from the closest vertex are smaller than the tolerance CollapseToll multiplied by the length of the edge intersected.

For some of the proposed cutting directions we switch from the selected refinement criterion to the Maximum Momentum criterion in order to avoid the generation of cells with a huge aspect ratio (AR) defined as the ratio between the longest distance between the centroid and the vertices and the smallest distance between the centroid and the edges. In order to avoid the generation of elements with a huge aspect ratio, we define a fixed value denoted by MaxAR and if the aspect ratio of the cell to cut is larger than MaxAR, the cutting criterion is always MaxMom.

*5.2.1. Maximum Momentum (MaxMom)*

This cutting strategy of marked cells is based on the choice of cutting direction that is orthogonal to the direction of the eigenvector associated to the smallest eigenvalue of the inertia tensor of the cell (Figure 3a).

*5.2.2. Trace Direction (TrDir)*

This cutting strategy is based on the choice as cutting direction of the direction parallel to a trace (if any), see Figure 3c, where we have assumed that

the edge with several aligned points belongs to the trace. The rationale of this approach is related to the known property of the solution to the considered problem that displays strong gradient components in the direction orthogonal to the traces. If a cell intersects more than one trace we switch to MaxMom criterion.

*5.2.3. Maximum Number of Points (MaxPnt)*

This cutting strategy is based on the choice as cutting direction of the direction of the vector connecting the centroid of the cell $X_C$ to the center of mass of the vertices $X_G$, see Figure 3c. This vector points towards the region of the marked cell with the highest density of vertices and should split the cell balancing the vertices of the two new subcells. In this strategy it is mandatory to define the option MaxNP, to set how many points the cells can have. This strategy switches to the MaxMom strategy in two cases: the first when the number of points of the cell to cut is less than MaxNP, the second when the centroid and the center of mass have a distance under a fixed tolerance. We remark that this refinement strategy can be considered as a simple improvement of the MaxMom strategy being the refinement strategy different only for the cells with a large number of cells and considering that when the number of vertices of the cell is larger than MaxNP this refinement strategy aims at dropping the number of vertices of the two produced cells under MaxNP.

*5.2.4. Maximum Edge (MaxEdg)*

This cutting strategy is based on the choice as cutting direction of the direction that splits the longest edge of the cell in the middle, see Figure 3d, where the longest edge is the right-bottom edge. If a cell displays aligned edges, these are considered as one unique edge.

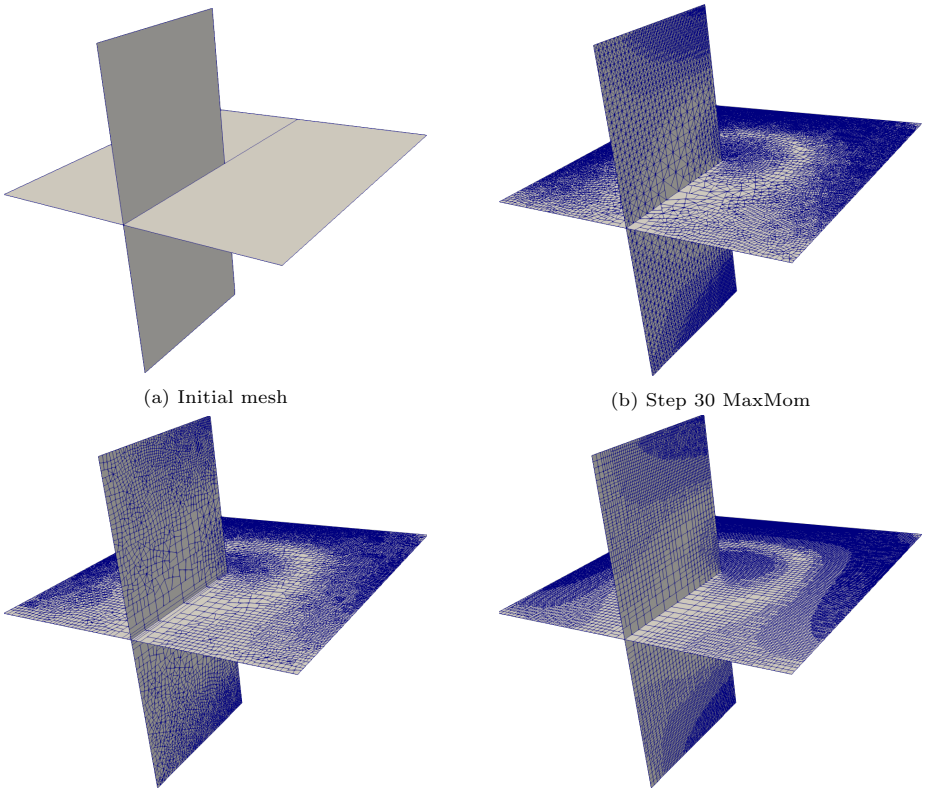
## 6. Numerical Results: optimality and effectivity index

In order to validate our refinement algorithm we test it on two simple DFNs for which an exact solution is known. We consider two DFNs with two and three fractures, labelled as Problem 1 and Problem 2, respectively.

This first set of tests aims at validating the equivalence relation stated in (2) between the error and the error estimator. We have tested this equivalence relation on the meshes produced during the adaptive process although this property holds true on any sufficiently refined mesh. We apply our adaptive algorithm and compare at each refinement iteration the error and the error estimator computing the *effectivity index*

$$\varepsilon = \frac{err}{est_\delta}$$

in order to verify that it is independent of the mesh size obtained by adaptive refinements. See [32] for the same analysis performed on uniformly refined meshes.



(a) Initial mesh

(b) Step 30 MaxMom

(c) Step 30 TrDir

(d) Step 30 MaxEdg

Figure 4: Problem 1: DFN with meshes at refining step 30.

As stopping criterion for the adaptive process, we require the following condition on the estimated relative error:

$$\frac{est_\delta}{\left(\sum_{i=1}^{N} \left\| \sqrt{\mathcal{K}_i} \nabla \Pi_k^\nabla H_{\delta i} \right\|_{F_i}^2 \right)^{\frac{1}{2}}} \leq 0.05\,. \tag{3}$$

All the simulations here presented are performed with the following methods and parameters: VEM orders from 1 to 4, Preconditioned Conjugate Gradient as linear solver with relative stopping residual $1.0e-15$, the preconditioner being an incomplete Cholesky factorization implemented as described in [42], CollapseToll = 0.2, MaxAR = 10, MaxNP = 12 and C = 0.50.

### *6.1. Problem* 1

The geometry and the parameters of this test problem are described in detail in [43, Section 5.1]. The DFN is composed by two fractures that are planar

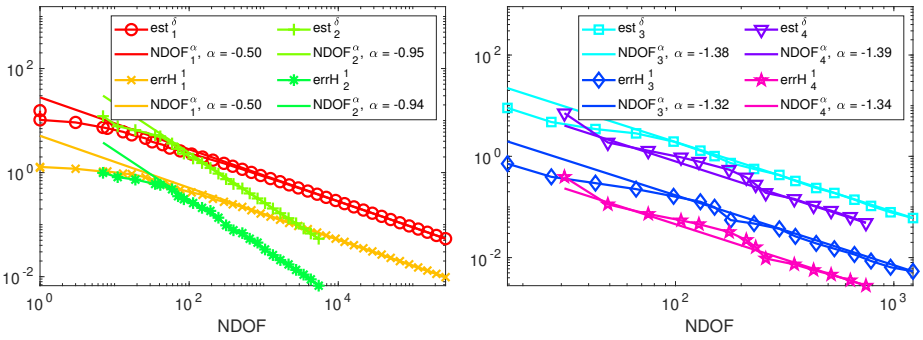



Figure 5: Problem 1: Rates of convergence ($\alpha$) for error and estimator with MaxMom refinement.

| TypeRef | Order 1 | Order 2 | Order 3 | Order 4 |
|---|---|---|---|---|
| MaxMom | $\alpha = -0.50$ | $\alpha = -0.95$ | $\alpha = -1.38$ | $\alpha = -1.39$ |
| | $\alpha_{err} = -0.50$ | $\alpha_{err} = -0.94$ | $\alpha_{err} = -1.32$ | $\alpha_{err} = -1.34$ |
| TrDir | $\alpha = -0.49$ | $\alpha = -0.93$ | $\alpha = -1.67$ | $\alpha = -1.56$ |
| | $\alpha_{err} = -0.50$ | $\alpha_{err} = -0.92$ | $\alpha_{err} = -1.63$ | $\alpha_{err} = -1.30$ |
| MaxEdg | $\alpha = -0.50$ | $\alpha = -0.93$ | $\alpha = -1.37$ | $\alpha = -1.55$ |
| | $\alpha_{err} = -0.50$ | $\alpha_{err} = -0.93$ | $\alpha_{err} = -1.41$ | $\alpha_{err} = -1.38$ |

Table 1: Problem 1: Rates of convergence for the estimator ($\alpha$) and the error ($\alpha_{err}$) with refinement criteria MaxMom, TrDir and MaxEdg.

rectangles defined as

$$F_1 = (-1,1) \times (-1,1) \times \{0\}\,; \quad F_2 = (-1,0) \times \{0\} \times (-1,1)\,.$$

Thus, the only trace of the DFN is $\Gamma_1 = (-1,0) \times \{0\} \times \{0\}$. The differential model is set up by choosing the following exact solutions:

$$h_1(x,y,z) = \left(x^2-1\right)\left(y^2-1\right)\left(x^2+y^2\right)\cos\left(\frac{1}{2}\text{arctan2}(x,y)\right)\,,$$
$$h_2(x,y,z) = -\left(z^2-1\right)\left(x^2-1\right)\left(x^2+z^2\right)\cos\left(\frac{1}{2}\text{arctan2}(z,x)\right)\,,$$

where $\text{arctan2}(x,y)$ is the four-quadrant inverse tangent, that is the arctan of $y/x$ in $[-\pi,\pi]$. The transmissivity of both fractures is set to 1.

In Figure 4 we report some of the meshes generated during the refinement process. We remark that in this test problem the presence of a known forcing function on the fractures induces a refinement in the whole domain, this will not be the same for the flow simulations in which the rock matrix surrounding the DFNs is considered impervious (Section 7).

In Figure 5 we report the convergence behaviour of the error ($\text{errH}^1$) and of the error estimator ($\text{est}_\delta$) and the final rates of convergence $\alpha$ with respect to the

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 4 | 1 | 0.0064 | 1 |
| 5 | 16 | 8 | 0.0221 | 5 |
| 10 | 58 | 44 | 0.0318 | 12 |
| 15 | 232 | 208 | 0.0326 | 23 |
| 20 | 958 | 910 | 0.0345 | 43 |
| 25 | 3997 | 3901 | 0.0343 | 88 |
| 30 | 16696 | 16214 | 0.0337 | 181 |
| 35 | 69827 | 67074 | 0.0334 | 370 |
| 39 | 219297 | 207112 | 0.0333 | 662 |
| 40 | 291581 | 274982 | 0.0332 | 777 |

Table 2: Problem 1: MaxMom refinement Order 1.

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 4 | 1 | 0.0064 | 1 |
| 5 | 14 | 5 | 0.0130 | 1 |
| 10 | 46 | 34 | 0.0441 | 12 |
| 15 | 167 | 147 | 0.0364 | 21 |
| 20 | 672 | 712 | 0.0356 | 46 |
| 25 | 2760 | 3232 | 0.0377 | 94 |
| 30 | 11528 | 13848 | 0.0357 | 188 |
| 35 | 48362 | 57738 | 0.0353 | 378 |
| 39 | 152185 | 178922 | 0.0349 | 695 |
| 40 | 202798 | 237727 | 0.0347 | 846 |

Table 3: Problem 1: TrDir refinement Order 1.

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 4 | 1 | 0.0064 | 1 |
| 5 | 16 | 7 | 0.0271 | 3 |
| 10 | 58 | 43 | 0.0347 | 12 |
| 15 | 239 | 223 | 0.0381 | 21 |
| 20 | 974 | 1007 | 0.0382 | 47 |
| 25 | 4101 | 4420 | 0.0382 | 100 |
| 30 | 17254 | 18980 | 0.0380 | 205 |
| 35 | 72401 | 80561 | 0.0386 | 414 |
| 37 | 128831 | 139214 | 0.0385 | 545 |

Table 4: Problem 1: MaxEdg refinement Order 1.

total number of degrees of freedom (NDOF): $est_\delta \sim (NDOF)^\alpha$, computed on the basis of the last five refinement iterations, considering the strategy MaxMom criterion. We can clearly appreciate a parallel behaviour of error and error estimator as well as the almost optimal asymptotic rate of convergence very close to $-0.5$ and $-1$ for the VEM orders 1 and 2, respectively. For higher VEM orders the sub-optimal rates of convergence are due to the bounded Besov regularity of the solution around the internal trace-tip. In Table 1 we report the rates of convergence for the estimator and for the error obtained by the refinement strategies MaxMom, TrDir and MaxEdg. For this problem we do not report results for the MaxPnt criterion that is always the MaxMom criterion being the number of vertices of the cells always smaller than MaxNP.

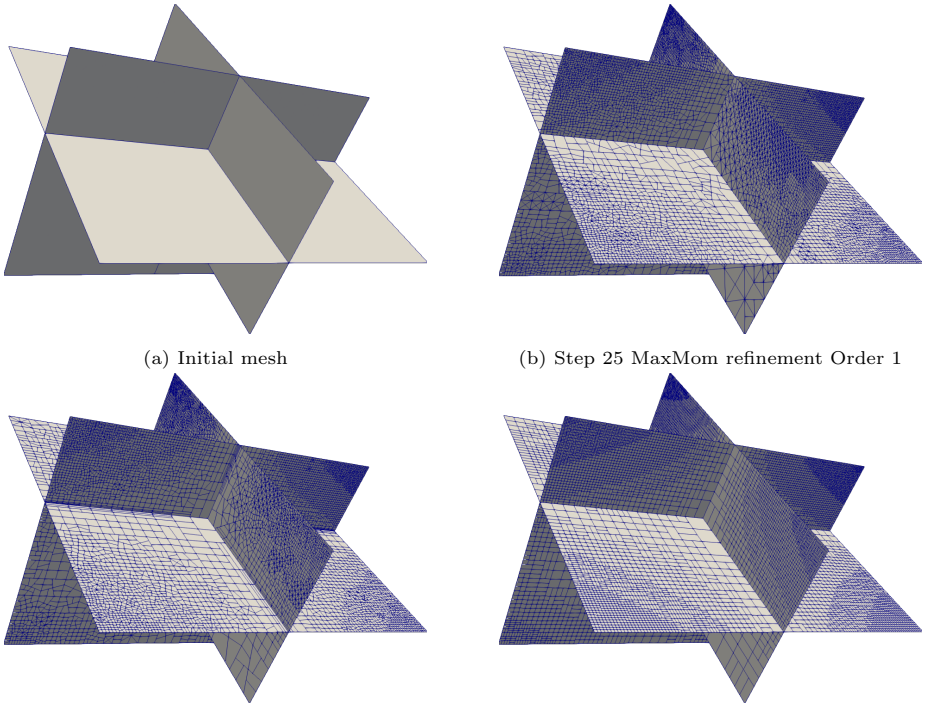
In Tables 2, 3 and 4 we report the most significant quantities to describe the refinement process for the MaxMom, TrDir and MaxEdge strategies: NCell is the total number of cells on the DFN, NDOF is the total numbers of degrees of freedom, $\varepsilon$ is the effectivity index, PCG-It is the number of conjugate gradient iterations performed. For all the strategies, we highlight the relatively small variations of $\varepsilon$ with respect to the large variations of number of cells and degrees of freedom, after the first iterations corresponding to very small number of NDOF. We also remark the weak growing of PCG-It with respect to the growing of NDOF. Observing the results concerning the number of degrees of freedom and the number of cells as well as the PCG iterations, the MaxEdg produces a lower NDOF and thus a lower PCG-It.

*6.2. Problem* 2

| TypeRef | Order 1 | Order 2 | Order 3 | Order 4 |
|---|---|---|---|---|
| MaxMom | $\alpha = -0.48$ | $\alpha = -0.93$ | $\alpha = -1.22$ | $\alpha = -1.49$ |
| | $\alpha_{err} = -0.49$ | $\alpha_{err} = -0.95$ | $\alpha_{err} = -1.23$ | $\alpha_{err} = -1.52$ |
| TrDir | $\alpha = -0.49$ | $\alpha = -0.95$ | $\alpha = -1.14$ | $\alpha = -1.49$ |
| | $\alpha_{err} = -0.49$ | $\alpha_{err} = -0.92$ | $\alpha_{err} = -1.20$ | $\alpha_{err} = -1.52$ |
| MaxEdg | $\alpha = -0.49$ | $\alpha = -0.92$ | $\alpha = -1.32$ | $\alpha = -1.47$ |
| | $\alpha_{err} = -0.47$ | $\alpha_{err} = -0.94$ | $\alpha_{err} = -1.35$ | $\alpha_{err} = -1.68$ |

Table 5: Problem 2: Rates of convergence for the estimator ($\alpha$) and the error ($\alpha_{err}$) with refinement criteria MaxMom, TrDir and MaxEdg.

The geometry and the parameters of this test problem are described in detail in [30, Section 6.1]. The DFN is composed by three fractures and three traces, defined as



(a) Initial mesh

(b) Step 25 MaxMom refinement Order 1

(c) Step 25 TrDir refinement Order 1

(d) Step 25 MaxEdg refinement Order 1

Figure 6: Problem 2: DFN with meshes at refining step 25

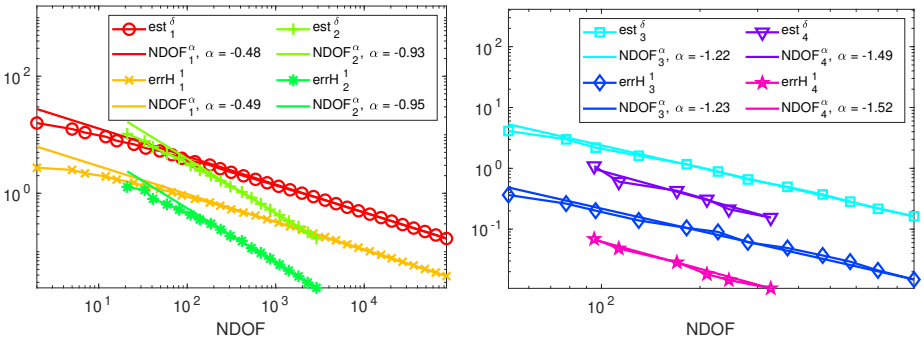



Figure 7: Problem 2: Rates of convergence ($\alpha$) for error and estimator with MaxMom refinement.

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 12 | 2 | 0.0296 | 1 |
| 5 | 30 | 16 | 0.0472 | 6 |
| 10 | 114 | 87 | 0.0541 | 13 |
| 15 | 455 | 432 | 0.0552 | 25 |
| 20 | 1875 | 1979 | 0.0557 | 55 |
| 25 | 7752 | 8622 | 0.0552 | 114 |
| 30 | 32221 | 36261 | 0.0531 | 234 |
| 32 | 57076 | 64484 | 0.0520 | 309 |

Table 6: Problem 2: MaxMom refinement Order 1.

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 12 | 2 | 0.0296 | 1 |
| 5 | 30 | 16 | 0.0441 | 5 |
| 10 | 113 | 92 | 0.0500 | 13 |
| 15 | 433 | 428 | 0.0531 | 32 |
| 20 | 1741 | 1933 | 0.0596 | 77 |
| 25 | 7047 | 8221 | 0.0581 | 152 |
| 30 | 28933 | 34475 | 0.0574 | 411 |
| 33 | 67557 | 80552 | 0.0562 | 542 |

Table 7: Problem 2: TrDir refinement Order 1.

| Step | # Cells | NDOF | $\varepsilon$ | PCG-It |
|---|---|---|---|---|
| 1 | 12 | 2 | 0.0296 | 1 |
| 5 | 30 | 16 | 0.0476 | 6 |
| 10 | 113 | 89 | 0.0555 | 12 |
| 15 | 450 | 418 | 0.0578 | 24 |
| 20 | 1847 | 1885 | 0.0614 | 55 |
| 25 | 7737 | 8203 | 0.0636 | 114 |
| 30 | 32385 | 34978 | 0.0679 | 243 |
| 31 | 43231 | 47014 | 0.0681 | 286 |

Table 8: Problem 2: MaxEdg refinement Order 1.

$$
\begin{aligned}
F_1 &= \left(-1, \frac{1}{2}\right) \times (-1,1) \times \{0\}\,, & \Gamma_1 &= F_1 \cap F_2 = \left(-1, \frac{1}{2}\right) \times \{0\} \times \{0\}\,,\\
F_2 &= (-1,0) \times \{0\} \times (-1,1)\,, & \Gamma_2 &= F_1 \cap F_3 = (-1,0) \times \{0\} \times (-1,1)\,,\\
F_3 &= \left\{\frac{1}{2}\right\} \times (-1,1) \times (-1,1)\,, & \Gamma_3 &= F_2 \cap F_3 = \left\{-\frac{1}{2}\right\} \times \{0\} \times (-1,1)\,.
\end{aligned}
$$

The chosen exact solution is defined on each fracture as follows:

$$
\begin{aligned}
h_1(x,y,z) &= -\frac{1}{10}\left(x+\frac{1}{2}\right)\left(8xy\left(x^2+y^2\right)\operatorname{arctan2}(y,x)+x^3\right)\,,\\
h_2(x,y,z) &= -\frac{1}{10}\left(x+\frac{1}{2}\right)x^3\left(1-8\pi\,|z|\right)\,,\\
h_3(x,y,z) &= y(y-1)(y+1)(z-1)z\,,
\end{aligned}
$$

and all fractures have transmissivities set to 1.

In Figure 6, we report some of the meshes generated by the refinement process. Similarly to the previous test, a refinement is induced also far from traces due to the presence of a non-null forcing term.

In Figure 7 we report the rates of convergence ($\alpha$) of the error and of the error estimator considering the strategy MaxMom. The rates of convergence for the strategies MaxMom, TrDir and MaxEdg are shown in the Table 5. The convergence rates are computed on the basis of the last five refinement iterations: again we can remark a very good agreement between the error and the estimator. The sub-optimal rates of convergence with higher VEM orders is still due to the bounded regularity of the solution.

In Tables 6-8 we report the same quantities reported in Tables 2-4 in order to describe the refinement process. We observe again that the effectivity index is almost independent of the meshsize since, as the mesh starts to have a sufficient number of DOFs, it displays small variations. As previously observed for Problem 1, the MaxEdg strategy produces a lower NDOF and thus a lower PCG-It.

## 7. Numerical Results on a realistic DFN

In this section we discuss the results obtained by the four presented refinement strategies when applied to more realistic DFNs. The geometry of the considered DFNs is fixed and is composed by 86 fractures and 159 traces, with a maximum number of traces per fracture equal to 11 and a mean value of traces per fracture equal to 1.85, see Figure 8. Working with this DFN we can easily produce several polygonal elements with a different number of edges, as well as, aligned edges. Moreover, the starting coarse mesh is commonly a poor quality mesh. All these configurations can be considered a good and challenging test

for the reliability and robustness of proposed approaches. The analysis for the quality of the elements generated by the algorithm can provide quite general results that can be extended to many different contexts.

We consider two test cases where the transmissivities of the fractures are sampled from two log-normal distributions having standard deviations equal to 10 and $10^4$, respectively. These two problems are tagged with the labels DFN86E01 and DFN86E04.

The problems considered have no forcing terms, and the flux is driven by the presence of two Dirichlet boundary conditions (10 on the boundaries at $x = 0$ and 0 at the boundaries at $x = 1000$), whereas homogeneous Neumann boundary conditions are imposed on all the other boundaries.

In the following analysis we display the convergence rates of the error estimate with respect to the number of degrees of freedom and we assess how the refinement strategies impact on the aspect ratio of the cells of a selected fracture (Fracture 72), and on the iterations of the preconditioned conjugate gradient method used to solve the linear system.

Figure 8: DFN with 86 fractures.

In Figures 9 and 10 we display the behaviour of the estimators with respect to the number of DOFs for the four refinement strategies considered and we report the slope of the estimator for each VEM order, for the two problems. We can observe that all the refinement strategies display an optimal asymptotic rate of convergence up to VEM order 2 ($-0.5$ for the VEM Order 1 and $-1.0$ for the VEM Order 2). For higher VEM orders the bounded regularity constraints the rates of convergence. The displayed behaviours suggest that the higher order VEM advisable for DFN flow simulations is $k = 2$. The trend of the

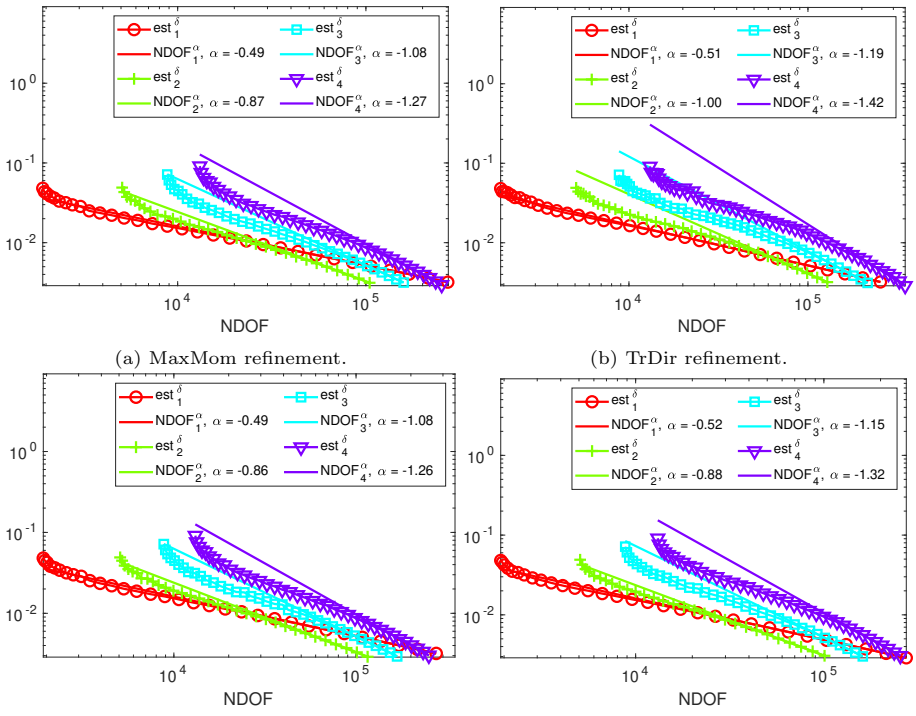
rate of convergence results similar even if the fluxes in the these two DFNs are completely different.

In Figure 11 we plot the behaviour of the ratio PCG-It/NDOF. After the initial noisy behaviour we can observe a decreasing asymptotical trend for all the considered refinement strategies. These plots highlight the advantages of a suitably refined mesh also on the performances of the linear solver.

In Figure 12 we report the minimal mesh on Fracture 72: this is the common initial mesh for all the refinement strategies and for both DFN86E01 and DFN86E04. In Figures 13 and 14 we display the meshes produced by all the considered refinement strategies. The TrDir strategy produces a stronger and sharper refinement along the traces. The switch from TrDir to MaxMom for elements with an aspect ratio over MaxAR prevents the generation of badly shaped elements parallel to the traces. The MaxPnt strategy produces a mesh quite similar to the mesh produced by MaxMom due to the fact that we fave few cells with a number of vertices larger than MaxNP during the refinement process. In Figures 15 and 16 we report the minimum, the mean and the maximum aspect ratios of the cells on Fracture 72 along the refinement process. We remark that in all the strategies we use MaxMom strategy to refine the elements with large aspect ratio. The MaxMom and the MaxEdg strategies produce a decreasing mean aspect ratio, whereas TrDir and MaxPnt have a different behaviour. A slight difference from the two figures can be seen in the TrDir plots. The average AR grows in the DFN86E04 test case because the MaxMom strategy is less used due to the weaker refinement around the traces due to the different transmissivities on the intersecting fractures that justify a smaller flux (notice the more coarse mesh for DFN86E04 comparing Figures 13 and 14).

## 8. Conclusions

In this paper we propose several refinement algorithms for polygonal meshes, suitable for polygonal numerical methods such as Virtual Element methods (VEM), Discontinuous Galerkin methods, Hybrid High Order methods or Mimetic Finite Differences. The proposed strategies are based on selecting proper cutting directions, aimed at preserving the regularity of the resulting polygons and the quality of the resulting mesh. A suitable collapsing of intersection points is also discussed, in order to avoid the generation of too small new edges. A detailed discussion of the overall resulting mesh quality and corresponding rates of convergence is reported. An adaptive algorithm for the simulation of flow in Discrete Fracture Networks (DFN) using VEM is used to test the proposed refinement strategies in complex geometrical configurations. Such algorithm is based on the generation of a minimal mesh on the given DFN, that is refined using a residual based *a posteriori* error estimate to mark the cells to be refined and the proposed refining strategies to refine the mesh. Numerical results show the effectiveness and the robustness of the proposed refinement strategies, and confirm the viability of the approach. A comparison of the results provided by the four proposed refinement strategies on the test problems yields us to infer that all the methods behave correctly, improving the representation of




(a) MaxMom refinement.

(b) TrDir refinement.

(c) MaxPnt refinement.

(d) MaxEdg refinement.

Figure 9: DFN86E01: Estimator vs. NDOF

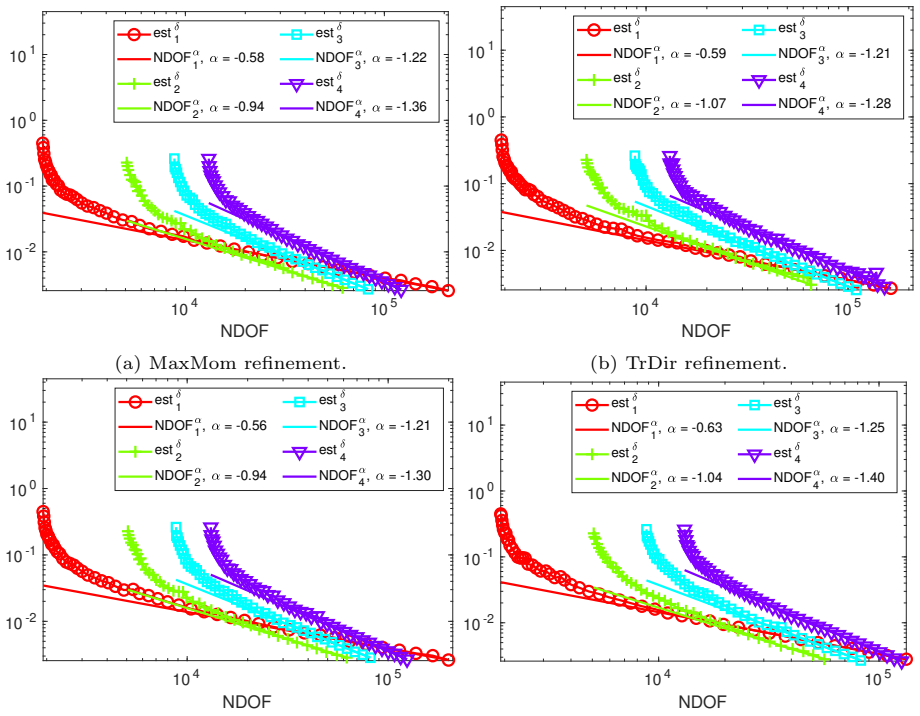
the solution. From the mesh quality point of view, the MaxMom strategy and the MaxEdg strategy work better in reducing the aspect ratio of the elements, whereas MaxPnt and TrDir display sometimes some oscillations of the maximum aspect ratio. Moreover, the MaxMom strategy requires a lower computational cost and a less involved implementation, dealing with all the considered situations in a unified approach, in contrast to the MaxEdg that requires a slightly more complex implementation to deal with elements displaying aligned edges. On the other hand, when the mesh is composed mainly by simple polygons, the MaxEdg strategy performs slightly better, as can be seen in Problem 1 and 2.

Finally, an extension of the MaxMom refinement strategy has been applied to an anisotropic mesh refinement procedure in [35].

## References

[1] W. S. Dershowitz, H. H. Einstein, Characterizing rock joint geometry with joint system models, Rock Mechanics and Rock Engineering 1 (1988) 21–51.

[2] C. Fidelibus, G. Cammarata, M. Cravero, Hydraulic characterization of fractured rocks. In: Abbie M, Bedford JS (eds) Rock mechanics: new research., Nova Science Publishers Inc., New York, 2009.




(a) MaxMom refinement.

(b) TrDir refinement.

(c) MaxPnt refinement.

(d) MaxEdg refinement.

Figure 10: DFN86E04: Estimator vs. NDOF

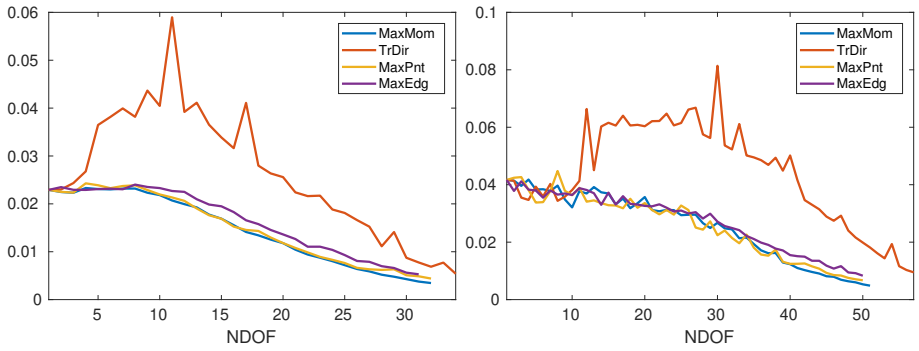

[3] M. C. Cacas, E. Ledoux, G. de Marsily, B. Tillie, A. Barbreau, E. Durand, B. Feuga, P. Peaudecerf, Modeling fracture flow with a stochastic discrete fracture network: calibration and validation: 1. the flow model, Water Resour. Res. 26 (1990) 479–489. doi:10.1029/WR026i003p00479.

[4] T. T. Garipov, M. Karimi-Fard, H. A. Tchelepi, Discrete fracture model for coupled flow and geomechanics, Computational Geosciences 20 (1) (2016) 149–160.

[5] G. Pichot, J. Erhel, J. de Dreuzy, A mixed hybrid mortar method for solving flow in discrete fracture networks, Applicable Analysis 89 (2010) 1629 – 643. doi:10.1080/00036811.2010.495333.

[6] G. Pichot, J. Erhel, J. de Dreuzy, A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks, SIAM Journal on scientific computing 34 (2012) B86 – B105. doi:10.1137/100804383.

[7] M. Karimi-Fard, L. J. Durlofsky, Unstructured adaptive mesh refinement for flow in heterogeneous porous media, in: ECMOR XIV-14th Euro-




(a) DFN86E01 (b) DFN86E04

Figure 11: Ratio between PCG-It and NDOF

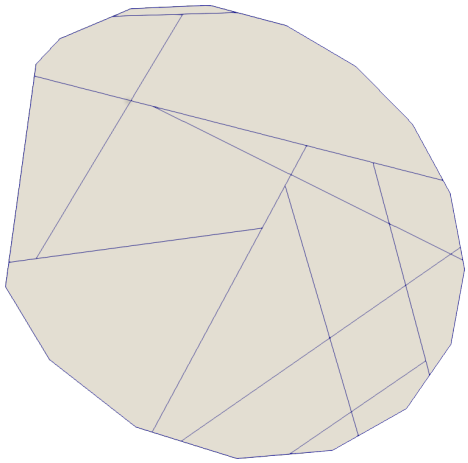
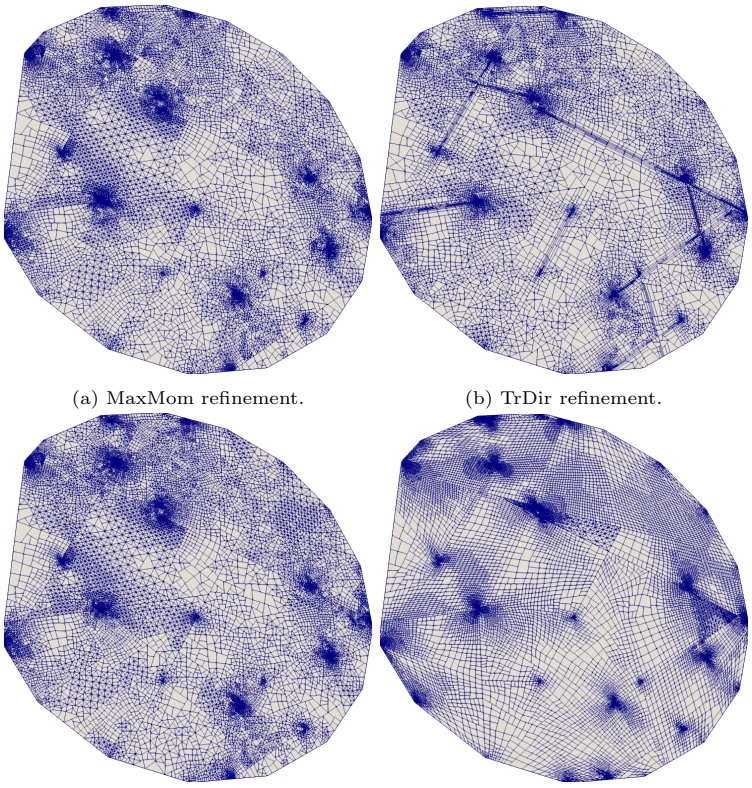


Figure 12: DFN86E01 - DFN86E04: initial mesh on Fracture 72.

pean conference on the mathematics of oil recovery, 2014. doi:10.3997/2214-4609.20141856.

[8] J. Hyman, C. Gable, S. Painter, N. Makedonska, Conforming delaunay triangulation of stochastically generated three dimensional discrete fracture networks: A feature rejection algorithm for meshing strategy, SIAM Journal on Scientific Computing 36 (2014) A1871–A1894. doi:10.1137/130942541.

[9] A. Fourno, T.-D. Ngo, B. Noetinger, C. L. Borderie, FraC: A new conforming mesh method for discrete fracture networks, Journal of Computational Physics 376 (2019) 713 – 732. doi:10.1016/j.jcp.2018.10.005.
URL http://www.sciencedirect.com/science/article/pii/S0021999118306624



(a) MaxMom refinement. (b) TrDir refinement.

(c) MaxPnt refinement. (d) MaxEdg refinement.

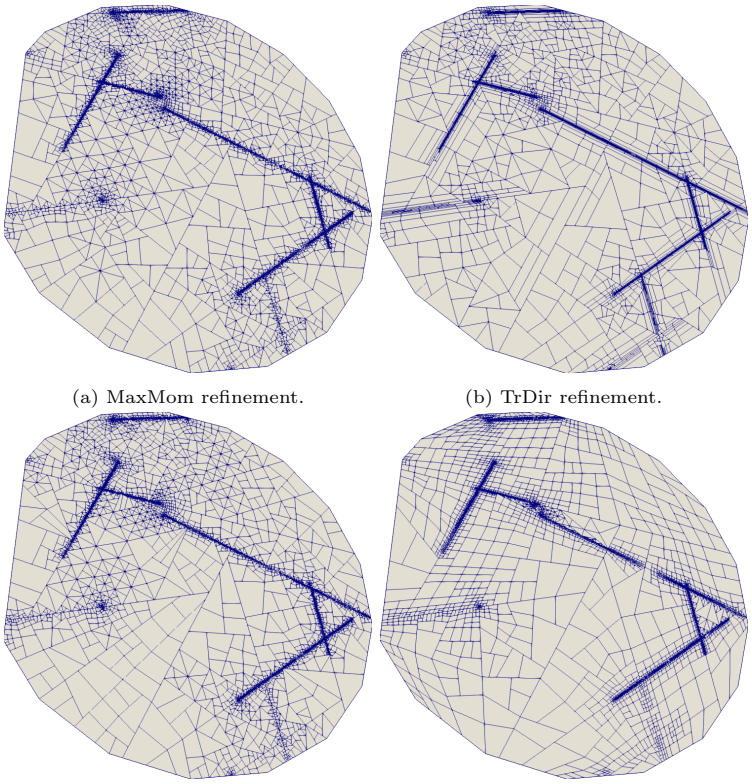
Figure 13: DFN86E01: final mesh on Fracture 72.


[10] T.-D. Ngo, A. Fourno, B. Noetinger, Modeling of transport processes through large-scale discrete fracture networks using conforming meshes and open-source software, Journal of Hydrology 554 (2017) 66 – 79. doi:10.1016/j.jhydrol.2017.08.052.
URL http://www.sciencedirect.com/science/article/pii/S0022169417305899

[11] M. F. Benedetto, A. Borio, S. Scialò, Mixed virtual elements for discrete fracture network simulations, Finite Elements in Analysis & Design 134 (2017) 55–67. doi:10.1016/j.finel.2017.05.011.

[12] A. Fumagalli, A. Scotti, A numerical method for two-phase flow in fractured porous media with non-matching grids, Advances in Water Resources 62 (2013) 454 – 464. doi:10.1016/j.advwatres.2013.04.001.

[13] A. Fumagalli, A. Scotti, An efficient XFEM approximation of Darcy flows in arbitrarily fractured porous media, Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles 69 (4) (2014) 555–564. doi:10.2516/ogst/2013192.



(a) MaxMom refinement. (b) TrDir refinement.

(c) MaxPnt refinement. (d) MaxEdg refinement.

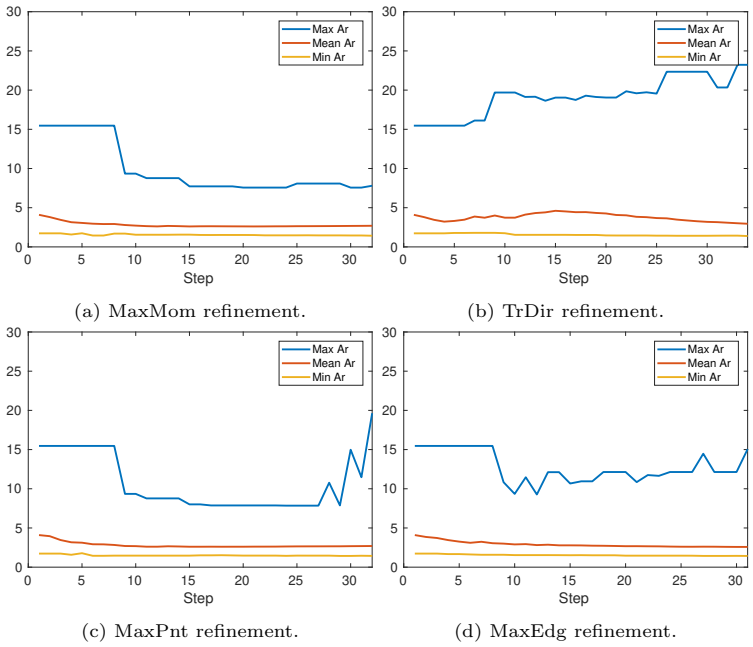
Figure 14: DFN86E04: final mesh on Fracture 72.


[14] P. F. Antonietti, L. Formaggia, A. Scotti, M. Verani, N. Verzott, Mimetic finite difference approximation of flows in fractured porous media, ESAIM: M2AN 50 (3) (2016) 809–832. doi:10.1051/m2an/2015087.

[15] A. Fumagalli, E. Keilegavlen, S. Scialò, Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations, J. Comput. Phys. 376 (2019) 694–712. doi:10.1016/j.jcp.2018.09.048.

[16] V. Lenti, C. Fidelibus, A *BEM* solution of steady-state flow problems in discrete fracture networks with minimization of core storage, Computers & Geosciences 29 (9) (2003) 1183 – 1190. doi:10.1016/S0098-3004(03)00140-7.

[17] W. S. Dershowitz, C. Fidelibus, Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method, Water Resource Res. 35 (1999) 2685–2691. doi:10.1029/1999WR900118.




(a) MaxMom refinement.

(b) TrDir refinement.

(c) MaxPnt refinement.

(d) MaxEdg refinement.

Figure 15: DFN86E01: aspect ratio statistics of the mesh on Fracture 72.

[18] O. Al-Hinai, M. F. Wheeler, I. Yotov, A generalized mimetic finite difference method and two-point flux schemes over Voronoi diagrams, ESAIM: Mathematical Modelling and Numerical Analysis 51 (2) (2017) 679–706.

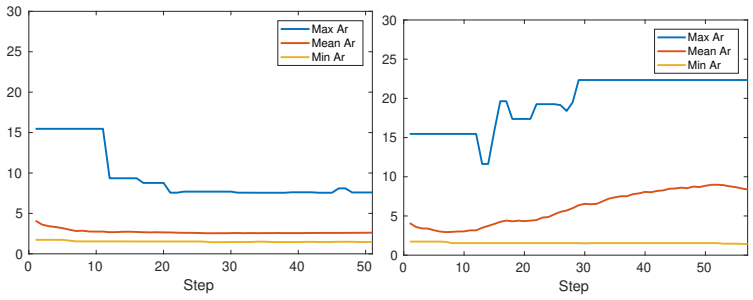
[19] B. Flemisch, I. Berre, W. Boon, A. Fumagalli, N. Schwenck, A. Scotti, I. Stefansson, A. Tatomir, Benchmarks for single-phase flow in fractured porous media, Advances in Water Resources 111 (2018) 239–258.

[20] K. Brenner, J. Hennicker, R. Masson, P. Samier, Gradient discretization of hybrid-dimensional Darcy flow in fractured porous media with discontinuous pressures at matrix fracture interfaces, IMA Journal of Numerical Analysis 37 (3) (2017) 1551–1585.

[21] R. Ahmed, M. G. Edwards, S. Lamine, B. A. Huisman, M. Pal, CVD-MPFA full pressure support, coupled unstructured discrete fracture matrix Darcy-flux approximations, Journal of Computational Physics 349 (2017) 265 – 299.

[22] S. Berrone, S. Pieraccini, S. Scialò, A PDE-constrained optimization formulation for discrete fracture network flows, SIAM J. Sci. Comput. 35 (2) (2013) B487–B510. doi:10.1137/120865884.

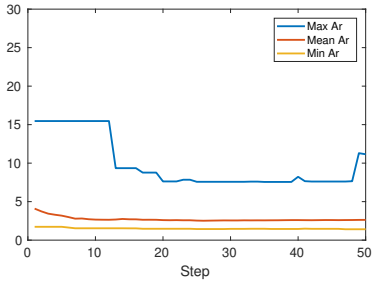
[23] S. Berrone, S. Pieraccini, S. Scialò, An optimization approach for large scale simulations of discrete fracture network flows, J. Comput. Phys. 256 (2014) 838–853. doi:10.1016/j.jcp.2013.09.028.



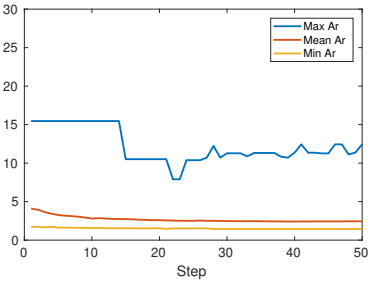

(a) MaxMom refinement.




(b) TrDir refinement.




(c) MaxPnt refinement.




(d) MaxEdg refinement.

Figure 16: DFN86E04: aspect ratio statistics of the mesh on Fracture 72.

[24] S. Berrone, C. Fidelibus, S. Pieraccini, S. Scialò, Simulation of the steady-state flow in discrete fracture networks with non-conforming meshes and extended finite elements, Rock Mechanics and Rock Engineering 47 (6) (2014) 2171–2182. doi:10.1007/s00603-013-0513-5.

[25] M. Benedetto, S. Berrone, S. Pieraccini, S. Scialò, The virtual element method for discrete fracture network simulations, Comput. Methods Appl. Mech. Engrg. 280 (0) (2014) 135 – 156. doi:10.1016/j.cma.2014.07.016.

[26] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, A. Russo, Basic principles of virtual element methods, Mathematical Models and Methods in Applied Sciences 23 (01) (2013) 199–214. doi:10.1142/S0218202512500492.
URL http://www.worldscientific.com/doi/abs/10.1142/S0218202512500492

[27] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, A. Russo, Equivalent projectors for virtual element methods, Computers & Mathematics with Applications 66 (2013) 376–391.

[28] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, Virtual element methods for general second order elliptic problems on polygonal meshes, Mathematical Models and Methods in Applied Sciences 26 (04) (2015) 729–750. doi:10.1142/S0218202516500160.

[29] L. Beirão da Veiga, C. Lovadina, A. Russo, Stability analysis for the virtual element method, Mathematical Models and Methods in Applied Sciences 27 (13) (2017) 2557–2594.
URL www.scopus.com

[30] M. 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. doi:10.1016/j.jcp.2015.11.034.

[31] M. 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. doi:10.1016/j.finel.2015.10.003.

[32] S. Berrone, A. Borio, A residual a posteriori error estimate for the virtual element method, Mathematical Models and Methods in Applied Sciences 27 (08) (2017) 1423–1458. doi:10.1142/S0218202517500233.
URL http://www.worldscientific.com/doi/abs/10.1142/S0218202517500233

[33] S. Berrone, C. Canuto, S. Pieraccini, S. Scialò, Uncertainty quantification in discrete fracture network models: stochastic fracture transmissivity, Comput. Math. Appl. 70 (4) (2015) 603–623. doi:10.1016/j.camwa.2015.05.013.

[34] S. Berrone, C. Canuto, S. Pieraccini, S. Scialò, Uncertainty quantification in discrete fracture network models: Stochastic geometry, Water Resources Research 54 (2) (2018) 1338–1352. doi:10.1002/2017WR021163.
URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017WR021163

[35] P. F. Antonietti, S. Berrone, A. Borio, A. D'Auria, M. Verani, S. Weisser, Anisotropic a posteriori error estimate for the virtual element method (2020). arXiv:2001.00381.

[36] L. Beirão da Veiga, G. Manzini, L. Mascotto, A posteriori error estimation and adaptivity in hp virtual elements, Numerische Mathematik 143 (1) (2019) 139–175.

[37] E. Artioli, L. Beirão da Veiga, M. Verani, An adaptive curved virtual element method for the statistical homogenization of random fibre-reinforced composites, Finite elements in analysis and design 177 (2020) 103418.

[38] F. Dassi, J. Gedicke, L. Mascotto, Adaptive virtual element methods with equilibrated flux (2020). arXiv:2004.11220.

[39] L. Beirão da Veiga, F. Brezzi, L. D. Marini, A. Russo, The hitchhiker's guide to the virtual element method, Mathematical Models and Methods in Applied Sciences 24 (08) (2014) 1541–1573. doi:10.1142/

S021820251440003X.
URL https://doi.org/10.1142/S021820251440003X

[40] W. Dörfler, A convergent adaptive algorithm for Poisson's equation, SIAM Journal on Numerical Analysis 33 (3) (1996) 1106–1124. doi:10.1137/0733054.

[41] S. Berrone, A. Borio, Orthogonal polynomials in badly shaped polygonal elements for the virtual element method, Finite Elements in Analysis & Design 129 (2017) 14–31. doi:10.1016/j.finel.2017.01.006.

[42] Eigen conjugate gradient, http://eigen.tuxfamily.org/dox/classEigen_1_1ConjugateGradient.html, URL Accessed: 12-09-2019.

[43] S. Berrone, S. Pieraccini, S. Scialò, On simulations of discrete fracture network flows with an optimization-based extended finite element method, SIAM J. Sci. Comput. 35 (2) (2013) A908–A935. doi:10.1137/120882883.
URL http://dx.doi.org/10.1137/120882883