POLITECNICO DI TORINO
Repository ISTITUZIONALE

Reliable a posteriori mesh adaptivity in discrete fracture network flow simulations

Original
Reliable a posteriori mesh adaptivity in discrete fracture network flow simulations / Berrone, S.; Borio, A.; Vicini, F.. - In: COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING. - ISSN 0045-7825. - ELETTRONICO. - 354:(2019), pp. 904-931. [10.1016/j.cma.2019.06.007]

Availability:
This version is available at: 11583/2725010 since: 2020-12-07T12:40:13Z

Publisher:
Elsevier

Published
DOI:10.1016/j.cma.2019.06.007

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)

19 April 2024

Reliable a posteriori mesh adaptivity in discrete fracture network flow simulations ☆

S. Berrone[a,b,*], A. Borio[a,b], F. Vicini[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

A new approach for the flow simulation in very complex Discrete Fracture Networks (DFNs) based on PDE-constrained optimization has been recently proposed in [1, 2] with the aim of improving robustness with respect to geometrical complexities. In [3] a rigorous derivation of "a posteriori" error estimates has been performed, proving an equivalence relation between the discretization error and suitable quantities that are computable from the discrete solution and the problem data.

In this paper, the previous results are applied to several fracture networks in order to investigate mesh adaptivity applied to complex realistic fracture networks.

Keywords: Discrete Fracture Network flow simulations, Simulations in complex geometries, Mesh adaptivity, A posteriori error estimates
2010 MSC: 65N30, 65N50, 68U20, 86-08, 86A05

1. Introduction

The simulation of underground phenomena in fractured media is a key aspect in many applications such as aquifers monitoring, nuclear waste disposal or the control of contaminant dispersion in the subsoil. In practical applications, many simulations need to be performed on randomly generated configurations of fractures generated from experimentally deduced probability distributions of the physical properties of the soil [4, 5]. The reliability of these simulations is of crucial importance, for instance, for the subsequent application of uncertainty quantification techniques for the estimation of relevant quantities of interest [6, 7].

In those situations where the rock matrix can be considered perfectly impervious, the Discrete Fracture Network model is often used, representing fractures as planar

☆This research has been partially supported by INdAM-GNCS Projects 2017 and 2018, and by the MIUR project "Dipartimenti di Eccellenza 2018-2022". Computational resources were partially provided by HPC@POLITO (`http://hpc.polito.it`) and by CINECA Project IsC58 HP10CDFLWH

*Corresponding author
Email addresses: `stefano.berrone@polito.it` (S. Berrone), `andrea.borio@polito.it` (A. Borio), `fabio.vicini@polito.it` (F. Vicini)

polygons intersecting randomly in the 3D space [8, 9]. These kind of domains usually present many geometrical challenges, in particular when one needs to mesh the domain while respecting some kind of conformity requirements, for example in order to apply standard finite element approaches. To circumvent these difficulties, many approaches have been devised, relying on domain decomposition techniques [10, 11, 12], or devising particular meshing strategies [13, 14, 15, 16, 17], or using extensions of the finite element method [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33].

In [1, 34, 2], a PDE-constrained optimization approach was devised, enabling for the use of completely non-conforming meshes, where finite element, extended finite element [35] or virtual element [36] spaces can be used. A scalable parallel implementation of such approach is proposed in [37]. In [3], a residual a posteriori error estimate is devised, taking into account both the discretization error and the error introduced by the non-conformity of the mesh with respect to intersections. In the present work, the estimate is extended in order to deal with general boundary conditions and is used to devise an adaptive meshing algorithm [38, 39, 40, 41]. In sections 2 and 3 we introduce the DFN model and define some useful notations, in section 4 we introduce the variational problem and discretize it as an optimization problem for a suitably defined functional, in section 5 we define the error estimators and state the a posteriori error estimate (Theorem 1), in section 6 we describe the adaptive algorithm and in section 7 we validate its application to some test cases with known solution and analyse some results on some more realistic DFNs.

2. Discrete Fracture Networks

A DFN is a model representing fractures inside an impervious rock matrix as planar polygons. We assume that the flow occurs only through fractures and across fracture-intersections in the normal direction. Let us denote by Ω the DFN, composed of N intersecting fractures. Each fracture F_i, $i \in \mathcal{I} = \{1, \dots, N\}$ has boundary ∂F_i and the boundary of Ω is $\partial\Omega = \bigcup_{i\in\mathcal{I}} \partial F_i$. Ω is supposed to be connected and we call *traces* the intersections between exactly two fractures, denoted by Γ_m, with $m \in \mathcal{M} = \{1, \dots, M\}$. For each $m \in \mathcal{M}$, let $\mathcal{I}_m = (i, j) \in \mathbb{N}^2$ be the ordered pair of indices of fractures such that $\Gamma_m = F_i \cap F_j$. For each fracture F_i, let $\mathcal{M}_i$ be the ordered set of indices of traces belonging to $\bar{F}_i$, and denote $M_i = \#\mathcal{M}_i$. $\mathcal{M}_i(k)$, for $k = 1, \dots, M_i$, denotes the k-th index of a trace in $\mathcal{M}_i$. For each $i \in \mathcal{I}$ and each $m \in \mathcal{M}_i$, $\mathbf{n}_{m,i}$ denotes a fixed unit vector normal to the trace Γ_m on F_i.

3. Notation

In this section we define some useful notations very close to those used in [3].

For any bounded open set ω, we denote by $(\cdot,\cdot)_\omega$ and by $\|\cdot\|_\omega$ the $\mathrm{L}^2(\omega)$ scalar product and norm, respectively, and by $|||\cdot|||_\omega$ the norm in $\mathrm{H}^1_0(\omega)$. Further, $(\cdot,\cdot)_{\alpha,\omega}$ and $\|\cdot\|_{\alpha,\omega}$ denote the scalar product and the norm in $\mathrm{H}^\alpha(\omega)$, respectively. For any given segment $\sigma \subset F_i$, $i \in \mathcal{I}$, $\gamma^i_\sigma : \mathrm{H}^1(F_i) \to \mathrm{H}^{\frac{1}{2}}(\sigma)$ is the trace operator and

$$\langle \mu, \beta \rangle_\sigma = {}_{\mathrm{H}^{-\frac{1}{2}}(\sigma)}\langle \mu, \beta \rangle_{\mathrm{H}^{\frac{1}{2}}(\sigma)} \qquad \forall \mu \in \mathrm{H}^{-\frac{1}{2}}(\sigma),\ \forall \beta \in \mathrm{H}^{\frac{1}{2}}(\sigma)$$

is the duality between $\mathrm{H}^{\frac{1}{2}}(\sigma)$ and $\mathrm{H}^{-\frac{1}{2}}(\sigma)$. For any given function $v \in \mathrm{H}^1(F_i)$, $\gamma_{\mathcal{M}_i}(v) \in \prod_{m\in\mathcal{M}_i} \mathrm{H}^{\frac{1}{2}}(\Gamma_m)$ is the tuple of the functional traces on the geological traces of the fracture f_i, $\gamma^i_{\Gamma_m}(v)$, $m \in \mathcal{M}_i$ sorted by increasing trace index m, and we denote

$$\forall \mu \in \prod_{m\in\mathcal{M}_i} \mathrm{H}^{-\frac{1}{2}}(\Gamma_m),\ \forall \beta \in \prod_{m\in\mathcal{M}_i} \mathrm{H}^{\frac{1}{2}}(\Gamma_m),\ \ \langle \mu, \beta\rangle_{\mathcal{M}_i} := \sum_{m\in\mathcal{M}_i} \langle \mu_m, \beta_m\rangle_{\Gamma_m}\ .$$

Moreover, we denote by

$$\gamma_{\mathcal{M}}(\mathbf{v}) = \left(\gamma_{\mathcal{M}_1}(v_1), \ldots, \gamma_{\mathcal{M}_N}(v_N)\right) \quad \forall \mathbf{v} \in V,$$

the tuple of the tuples of traces on each fracture. We introduce the functional space $V := V_1 \times \ldots \times V_N$, with $V_i := \mathrm{H}^1(F_i)$, $\forall i \in \mathcal{I}$. For any function $\mathbf{g} \in V$, we define, $\forall m \in \mathcal{M}$, and $\mathcal{I}_m = (i, j)$ the jump operator across a trace Γ_m, $[\![\mathbf{g}]\!]_{\Gamma_m} := \gamma^i_{\Gamma_m}(g_i) - \gamma^j_{\Gamma_m}\left(g_j\right)$. $\forall i \in \mathcal{I}$, $[\![\mathbf{g}]\!]_{\mathcal{M}_i}$ is the vector of jumps of $\mathbf{g}$ across the traces in $\mathcal{M}_i$, sorted according to trace index:

$$[\![\mathbf{g}]\!]_{\mathcal{M}_i} := \left([\![\mathbf{g}]\!]_{\Gamma_{\mathcal{M}_i(1)}}, \ldots, [\![\mathbf{g}]\!]_{\Gamma_{\mathcal{M}_i(M_i)}}\right) \in \prod_{m\in\mathcal{M}_i} \mathrm{H}^{\frac{1}{2}}(\Gamma_m)\ .$$

Similarly, given the diffusion parameter $\mu_i \in \mathrm{L}^\infty(F_i)$ uniformly positive and a function $g_i \in V_i$, $\left[\!\left[\mu_i \frac{\partial g_i}{\partial \mathbf{n}_{m,i}}\right]\!\right]_{\Gamma_m}$ is the jump of the co-normal derivative across Γ_m on F_i, and

$$\left[\!\left[\boldsymbol{\mu}\frac{\partial g_i}{\partial \mathbf{n}}\right]\!\right]_{\mathcal{M}_i} := \left(\left[\!\left[\mu_i \frac{\partial g_i}{\partial \mathbf{n}_{\mathcal{M}_i(1),i}}\right]\!\right]_{\Gamma_{\mathcal{M}_i(1)}}, \ldots, \left[\!\left[\mu_i \frac{\partial g_i}{\partial \mathbf{n}_{\mathcal{M}_i(M_i),i}}\right]\!\right]_{\Gamma_{\mathcal{M}_i(M_i)}}\right) \in \prod_{m\in\mathcal{M}_i} \mathrm{H}^{-\frac{1}{2}}(\Gamma_m)\ ,$$

is the tuple of the jumps of co-normal derivatives on the fracture F_i.

4. Problem Formulation

In this section, we briefly state the formulation of our optimization approach as proposed in [1, 2]. We are interested in computing the hydraulic head distribution within the DFN, that means we aim to solve local Darcy problems on each fracture, coupled together by suitable matching conditions.

We denote the hydraulic head in Ω as $\mathbf{h} = (h_1, \ldots, h_N) \in V$, where h_i is the hydraulic head on each fracture F_i. Let $\Gamma_D \subset \partial\Omega$ be the portion of the boundary of Ω where a Dirichlet condition $\mathbf{g}_D \in \prod_{i\in\mathcal{I}} \mathrm{H}^{\frac{1}{2}}(\Gamma_{D,i})$ is set, which we suppose to be not empty, where $\Gamma_{D,i} = \Gamma_D \cap F_i$. Let Γ_N be the boundary with Neumann conditions $\mathbf{g}_N \in \prod_{i\in\mathcal{I}} \mathrm{H}^{-\frac{1}{2}}(\Gamma_{N,i})$, where $\Gamma_{N,i} = \Gamma_N \cap F_i$. We assume $\Gamma_N \cap \Gamma_D = \emptyset$ and $\Gamma_N \cup \Gamma_D = \partial\Omega$. Furthermore, we denote by $V^D \subset V$ the sub-space of V of functions whose value on Γ_D is g_D, V^0 the sub-space of functions with zero value on Γ_D.

For each $i \in \mathcal{I}$, the local hydraulic head h_i is the solution of the following variational problem: *find* $\mathbf{h} \in V^D$ *such that,* $\forall i \in \mathcal{I}$

$$(\mu_i \nabla h_i, \nabla v)_{F_i} = (f_i, v)_{F_i} + \left\langle \left[\!\left[\mu_i \frac{\partial h_i}{\partial \mathbf{n}}\right]\!\right]_{\mathcal{M}_i}, \gamma_{\mathcal{M}_i}(v)\right\rangle_{\mathcal{M}_i} + \left\langle g_{N,i}, \gamma_{\Gamma_{N,i}}(v)\right\rangle_{\Gamma_{N,i}} \quad \forall v \in V^0_i, \tag{1}$$

where $f_i \in \mathrm{L}^2(F_i)$ and $\mu_i \in \mathrm{L}^\infty(F_i)$, $\forall i \in \mathcal{I}$. All the results are applicable to the case of uniformly positive definite tensors μ_i. The local problems are coupled together by some natural matching conditions, enforcing continuity of the hydraulic head and conservation of fluxes: $\forall m \in \mathcal{M}$, $\mathcal{I}_m = (i, j)$

$$\gamma^i_{\Gamma_m}(h_i) - \gamma^j_{\Gamma_m}\left(h_j\right) = 0\,, \tag{2}$$

$$\left[\!\left[\mu_i \frac{\partial h_i}{\partial \mathbf{n}_{m,i}}\right]\!\right]_{\Gamma_m} + \left[\!\left[\mu_j \frac{\partial h_j}{\partial \mathbf{n}_{m,j}}\right]\!\right]_{\Gamma_m} = 0\,. \tag{3}$$

4.1. Formulation as an optimization problem

Following [1, 2], problem (1)-(3) can be written equivalently as a PDE-constrained minimization problem of a suitable functional representing the matching conditions.

Let $\alpha \in \mathbb{R}$ be fixed and let, $\forall m \in \mathcal{M}$, $\Lambda_m \colon \mathrm{H}^{\frac{1}{2}}(\Gamma_m) \to \mathrm{H}^{-\frac{1}{2}}(\Gamma_m)$ be the natural embedding from $\mathrm{H}^{\frac{1}{2}}(\Gamma_m)$ to $\mathrm{H}^{-\frac{1}{2}}(\Gamma_m)$. We define

$$\begin{aligned}
\forall i \in \mathcal{I}\,, \forall m \in \mathcal{M}_i\,, u_i^m &:= \alpha\Lambda_m\left(\gamma^i_m(h_i)\right) + \left[\!\left[\mu_i \frac{\partial h_i}{\partial \mathbf{n}_{m,i}}\right]\!\right]_{\Gamma_m} \in \mathrm{H}^{-\frac{1}{2}}(\Gamma_m)\ , \\
\forall i \in \mathcal{I}\,, \mathbf{u}_i &:= \left(u_i^{\mathcal{M}_i(1)}, \ldots, u_i^{\mathcal{M}_i(M_i)}\right) \in U_i := \prod_{m\in\mathcal{M}_i} \mathrm{H}^{-\frac{1}{2}}(\Gamma_m)\ , \\
\mathbf{u} &:= (\mathbf{u}_1, \ldots, \mathbf{u}_N) \in U := \prod_{i\in\mathcal{I}} U_i\ .
\end{aligned}$$

For any $\mathbf{w} \in U$ we set $\{\!\{\mathbf{w}\}\!\}_{\Gamma_m} = w_i^m + w_j^m$, $\forall m \in \mathcal{M}$ with $\mathcal{I}_m = (i, j)$ and denote by $\{\!\{\mathbf{w}\}\!\}_{\mathcal{M}_i}$ the vector whose k-th component is $\{\!\{\mathbf{w}\}\!\}_{\Gamma_{\mathcal{M}_i(k)}}$. Finally, we denote by U_i^* and U^* the dual spaces of U_i and U, respectively and by $\Lambda \colon U^* \to U$ the vectorized embedding such that

$$\begin{aligned}
\Lambda(\mathbf{v}) &= \left(\Lambda_{\mathcal{M}_1}(\mathbf{v}_1), \ldots, \Lambda_{\mathcal{M}_N}(\mathbf{v}_N)\right) \\
&\text{where}\,, \forall i \in \mathcal{I}\,, \Lambda_{\mathcal{M}_i}(\mathbf{v}_i) = \left(\Lambda_{\mathcal{M}_i(1)}(v_i), \ldots, \Lambda_{\mathcal{M}_i(M_i)}(v_i)\right) \in U_i\,.
\end{aligned}$$

To define our functional, we first need to introduce the operator $\mathcal{H} \colon U \to V^D$, that associates to each vector $\mathbf{w} \in U$ a vector $\mathcal{H}(\mathbf{w}) = \left(h_1^w, \ldots, h_N^w\right) \in V^D$ such that

$$\begin{aligned}
\left(\mu_i \nabla h_i^w, \nabla v_i\right)_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}(h_i), \gamma_{\mathcal{M}_i}(v_i)\right)_{\mathcal{M}_i} &= (f_i, v_i)_{F_i} + \left\langle \mathbf{w}_i, \gamma_{\mathcal{M}_i}(v_i)\right\rangle_{\mathcal{M}_i} \\
&+ \left\langle g_{N,i}, \gamma_{\Gamma_{N,i}}(v_i)\right\rangle_{\Gamma_{N,i}} \quad \forall i \in \mathcal{I}\,,\ \forall \mathbf{v} \in V^0\,.
\end{aligned} \tag{4}$$

Using the above operator, we define, for each $m \in \mathcal{M}$, the functional $J_m \colon U \to \mathbb{R}$

$$J_m(\mathbf{w}) = \left\|[\![\mathcal{H}(\mathbf{w})]\!]_{\Gamma_m}\right\|^2_{\frac{1}{2},\Gamma_m} + \left\|\left\{\!\!\left\{\mathbf{w} - \alpha\Lambda\left(\gamma_{\mathcal{M}}(\mathcal{H}(\mathbf{w}))\right)\right\}\!\!\right\}_{\Gamma_m}\right\|^2_{-\frac{1}{2},\Gamma_m}\,. \tag{5}$$

Then, problem (1)-(3) can be seen as a constrained optimization problem:

$$\text{find } \mathbf{u} \in U \text{ such that } \ \mathbf{u} = \arg\min_{\mathbf{w}\in U} J(\mathbf{w})\,, \quad \text{where } J(\mathbf{u}) = \sum_{m\in\mathcal{M}} J_m(\mathbf{u})\,. \tag{6}$$

The well-posedness of (6) and its equivalence with problem (1)-(3) is proved in [2]. Introducing the Lagrange multiplier $\mathbf{p} \in V$, it can be proved that the solution $\mathbf{u}$ to (6), satisfies the following system:

$$\left\langle \left\{\!\!\left\{ \mathbf{u} - \alpha\Lambda\left(\gamma_{\mathcal{M}}(\mathbf{h})\right)\right\}\!\!\right\}_{\mathcal{M}_i}, \mu_i \right\rangle_{\mathcal{M}_i} = -\left(\gamma_{\mathcal{M}_i}(p_i), \mu_i\right)_{\mathcal{M}_i} \quad \forall \mu_i \in U_i^*, \tag{7}$$

$$\begin{aligned}(\mu_i \nabla p_i, \nabla v_i)_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}(p_i), \gamma_{\mathcal{M}_i}(v_i)\right)_{\mathcal{M}_i} &= \left(\left[\!\left[\mathbf{h}\right]\!\right]_{\mathcal{M}_i}, \gamma_{\mathcal{M}_i}(v_i)\right)_{\mathcal{M}_i} \\ &+ \left\langle \alpha\left\{\!\!\left\{\alpha\Lambda\left(\gamma_{\mathcal{M}}(\mathbf{h})\right) - \mathbf{u}\right\}\!\!\right\}_{\mathcal{M}_i}, \gamma_{\mathcal{M}_i}(v_i)\right\rangle_{\mathcal{M}_i} \quad \forall v_i \in V_i^0,\end{aligned} \tag{8}$$

$$\begin{aligned}(\mu_i \nabla h_i, \nabla v_i)_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}(h_i), \gamma_{\mathcal{M}_i}(v_i)\right)_{\mathcal{M}_i} &= (f_i, v_i)_{F_i} + \left\langle \mathbf{u}_i, \gamma_{\mathcal{M}_i}(v_i)\right\rangle_{\mathcal{M}_i} \\ &+ \left\langle g_{N,i}, \gamma_{\Gamma_{N,i}}(v_i)\right\rangle_{\Gamma_{N,i}} \quad \forall v_i \in V_i^0.\end{aligned} \tag{9}$$

From (2) - (3), we see that $\left[\!\left[\mathbf{h}\right]\!\right]_{\mathcal{M}}$ is the null vector as well as $\left\{\!\!\left\{ \mathbf{u} - \alpha\Lambda\left(\gamma_{\mathcal{M}}(\mathbf{h})\right)\right\}\!\!\right\}_{\mathcal{M}}$, therefore, the exact solution of (8) corresponds to $p_i = 0\ \forall i \in \mathcal{I}$.

4.2. Discretization of the optimization problem

The optimization problem (6) can be discretized using local finite element spaces on each fracture, defined on local triangulations that are completely independent of trace positions.

Let $\mathcal{T}_{\delta i}$ be a regular triangulation of F_i, and $\mathcal{T}_\delta = \bigcup_{i\in\mathcal{I}} \mathcal{T}_{\delta i}$, δ being a mesh parameter characterizing the mesh size. Let us define the finite dimensional subspaces $V_{\delta,i}^D \subset V_i^D$ and $V_{\delta,i}^0 \subset V_i^0$, $\forall i \in \mathcal{I}$. Moreover, given a partition of each Γ_m in subintervals, let $U_{\delta i}^m \subset \mathrm{L}^2(\Gamma_m) \subset \mathrm{H}^{-\frac{1}{2}}(\Gamma_m) = U_i^m$, $\forall i \in \mathcal{I}$, $\forall m \in \mathcal{M}_i$ be given and let us set $U_{\delta i} := \prod_{m\in\mathcal{M}_i} U_{\delta i}^m$, $\forall i \in \mathcal{I}$, $U_\delta := \prod_{i\in\mathcal{I}} U_{\delta i}$, $V_\delta^D := \prod_{i\in\mathcal{I}} V_{\delta i}^D$, $V_\delta^0 := \prod_{i\in\mathcal{I}} V_{\delta i}^0$.

Let $\mathcal{H}_\delta \colon U_\delta \to V_\delta^D$ such that the components of $\mathcal{H}_\delta(\mathbf{w}) = (h_{\delta 1}^w, \dots, h_{\delta N}^w)$ are the solutions of

$$\begin{aligned}\left(\mu_i \nabla h_{\delta i}^w, \nabla v_{\delta i}\right)_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}\left(h_{\delta i}^w\right), \gamma_{\mathcal{M}_i}(v_{\delta i})\right)_{\mathcal{M}_i} &= (f_i, v_{\delta i})_{F_i} + \left(\mathbf{w}_i, \gamma_{\mathcal{M}_i}(v_{\delta i})\right)_{\mathcal{M}_i} \\ &+ \left\langle g_{N,i}, \gamma_{\Gamma_{N,i}}(v_{\delta i})\right\rangle_{\Gamma_{N,i}} \quad \forall i \in \mathcal{I},\ \forall \mathbf{v}_\delta \in V_\delta^0.\end{aligned} \tag{10}$$

Then, we can define the following "discrete" functional $J_{\delta m} \colon U_\delta \to \mathbb{R}$ that approximates (5):

$$J_{\delta m}(\mathbf{w}_\delta) = \left\| \left[\!\left[\mathcal{H}_\delta(\mathbf{w}_\delta)\right]\!\right]_{\Gamma_m} \right\|_{\Gamma_m}^2 + \left\| \left\{\!\!\left\{ \mathbf{w}_\delta - \alpha\Lambda\left(\gamma_{\mathcal{M}}(\mathcal{H}_\delta(\mathbf{w}_\delta))\right)\right\}\!\!\right\}_{\Gamma_m} \right\|_{\Gamma_m}^2, \tag{11}$$

and write the following discrete version of problem (6)

$$\text{find } \mathbf{u}_\delta \in U_\delta \text{ such that } \mathbf{u}_\delta = \underset{\mathbf{w}_\delta \in U_\delta}{\arg\min}\, J_\delta(\mathbf{w}_\delta), \quad \text{where } J_\delta(\mathbf{u}_\delta) = \sum_{m\in\mathcal{M}} J_{\delta m}(\mathbf{u}_\delta). \tag{12}$$

The well-posedness of problem (12) is proved in [2], where it is also seen to be equiv-

alent to the following problem: *find* $\mathbf{h}_\delta \in V_\delta^D$, $\mathbf{p}_\delta \in V_\delta^0$, $\mathbf{u}_\delta \in U_\delta$ *such that,* $\forall i \in \mathcal{I}$,

$$\left(\left\{\!\!\left\{\mathbf{u}_\delta - \alpha\gamma_{\mathcal{M}}(\mathbf{h}_\delta)\right\}\!\!\right\}_{\mathcal{M}_i}, \mu_{\delta i}\right)_{\mathcal{M}_i} = -\left(\gamma_{\mathcal{M}_i}(p_{\delta i}), \mu_{\delta i}\right)_{\mathcal{M}_i} \quad \forall \mu_{\delta i} \in U_{\delta i}, \tag{13}$$

$$(\mu_i \nabla p_{\delta i}, \nabla v_{\delta i})_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}(p_{\delta i}), \gamma_{\mathcal{M}_i}(v_{\delta i})\right) = \left(\left[\!\left[\mathbf{h}_\delta\right]\!\right]_{\mathcal{M}_i} + \alpha\left\{\!\!\left\{\alpha\mathbf{h}_\delta - \mathbf{u}_\delta\right\}\!\!\right\}_{\mathcal{M}_i}, \gamma_{\mathcal{M}_i}(v_{\delta i})\right)_{\mathcal{M}_i} \quad \forall v_{\delta i} \in V_{\delta i}^0, \tag{14}$$

$$(\mu_i \nabla h_{\delta i}, \nabla v_{\delta i})_{F_i} + \alpha\left(\gamma_{\mathcal{M}_i}(h_{\delta i}), \gamma_{\mathcal{M}_i}(v_{\delta i})\right)_{\mathcal{M}_i} = (f_i, v_{\delta i})_{F_i} + \left(\mathbf{u}_{\delta i}, \gamma_{\mathcal{M}_i}(v_{\delta i})\right)_{\mathcal{M}_i} + \left\langle g_{N,i}, \gamma_{\Gamma_{N,i}}(v_{\delta i})\right\rangle_{\Gamma_{N,i}} \quad \forall v_{\delta i} \in V_{\delta i}^0. \tag{15}$$

The above equivalent formulation is then used in our a posteriori error analysis.

5. Error estimates

Before proceeding to the error estimate, we need to establish some useful notation. Following [3], we denote by $\mathcal{E}_\delta$ the set of edges of the triangles in $\mathcal{T}_\delta$. $\mathcal{E}_{\delta i}$ is the subset of $\mathcal{E}_\delta$ containing the edges defined on fracture F_i. For all $T \in \mathcal{T}_\delta$, let $\mathring{T}$ be the interior of T, and we denote by $\mathcal{M}_T$ the set of indices of those traces having non empty intersection with $\mathring{T}$, and define $\ell_T^m = \Gamma_m \cap \mathring{T}$, $\forall m \in \mathcal{M}_T$. Moreover, $\forall \sigma \in \mathcal{E}_\delta$ we indicate by $\mathcal{M}_\sigma$ the set of those $m \in \mathcal{M}$ such that $|\Gamma_m \cap \sigma| \neq \emptyset$. On each trace Γ_m, let $\mathcal{I}_m = (i, j)$ be the couple of the two fracture indices sharing Γ_m, we fix two discretizations $\Lambda_{m,i}$ and $\Lambda_{m,j}$ associated to the two fractures F_i and F_j, respectively. Finally, the symbol $h_\sharp$ denotes the diameter of an arbitrary geometrical object $\sharp$.

To introduce our error measure, we first define the quantities

$$e_{\mathbf{h}} = \mathbf{h} - \mathbf{h}_\delta, \qquad e_{\mathbf{p}} = \mathbf{p}_\delta, \qquad e_{\mathbf{u}} = \mathbf{u} - \mathbf{u}_\delta. \tag{16}$$

Then, the error we want to estimate is the quantity

$$err := \left[\sum_{i \in \mathcal{I}}\left(\|\mu_i \nabla e_{\mathbf{h}}\|_{F_i}^2 + \left\|\mu_i \nabla e_{\mathbf{p}}\right\|_{F_i}^2 + \|e_{\mathbf{u}}\|_U^2\right)\right]^{\frac{1}{2}}. \tag{17}$$

In [3], upper and lower bounds for this error measure were proved in the case where each fracture has Dirichlet boundary conditions that allows $\alpha = 0$. The extension to the more general case presented here is straightforward and leads to the definition of the following error fracture-based **local** estimators $\forall i \in \mathcal{I}$:

Residual estimator for the Darcy problem: $\forall T \in \mathcal{T}_{\delta i}$,

$$\eta_{H,T} := h_T \left\|f_i + \nabla \cdot (\mu_i \nabla h_{\delta i})\right\|_T, \tag{18}$$

Estimator for the approximation of the flux through edges: $\forall \sigma \in \mathcal{E}_{\delta i}$, if $\sigma \notin \Gamma_{N,i}$,

$$\xi_{F,\sigma} := (h_\sigma)^{\frac{1}{2}} \left\| \left[\!\left[\mu_i \frac{\partial h_{\delta i}}{\partial \mathbf{n}_\sigma}\right]\!\right]_\sigma - \tilde{u}_{\delta i,\sigma} \right\|_\sigma$$

$$\text{where} \quad \tilde{u}_{\delta i,\sigma} := \begin{cases} u_{\delta i}^m - \alpha\gamma_{\Gamma_m}^i(h_{\delta i}) & \text{on } \Gamma_m \cap \sigma, \forall m \in \mathcal{M}_\sigma, \\ 0 & \text{elsewhere}. \end{cases} \tag{19}$$

If $\sigma \subset \Gamma_{N,i}$,

$$\xi_{F,\sigma} := (h_\sigma)^{\frac{1}{2}} \left\| \left[\left[\mu_i \frac{\partial h_{\delta i}}{\partial \mathbf{n}_\sigma} \right]\right]_\sigma - g_{N,i} \right\|_\sigma . \tag{20}$$

Estimator for the nonconformity of the discretization: $\forall T \in \mathcal{T}_{\delta i}$, $\forall m \in \mathcal{M}_T$,

$$\xi^m_{NC,T} := \left(h_{\ell^m_T}\right)^{\frac{1}{2}} \left\| u^m_{\delta i} - \alpha \gamma^i_m(h_{\delta i}) \right\|_{\ell^m_T} . \tag{21}$$

Estimator for the pressure induced by discontinuity: $\forall T \in \mathcal{T}_{\delta i}$,

$$\eta_{P,T} := \|\mu_i \nabla p_{\delta i}\|_T . \tag{22}$$

Estimator for the pressure induced by the unbalancing of fluxes: $\forall m \in \mathcal{M}_i$, $\forall \lambda \in \Lambda_{m,i}$,

$$\xi^i_{P,\lambda} := h_\lambda^{\frac{1}{2}} \left\| \gamma^i_{\Gamma_m}(p_{\delta i}) \right\|_\lambda . \tag{23}$$

Estimator of the minimization error: $\forall m \in \mathcal{M}_i$, $\lambda \in \Lambda_{m,i}$,

$$J_{\delta,\lambda}(\mathbf{u}_\delta) := h_\lambda \left((1+\alpha)^2 \left\| \{\{\mathbf{u}_\delta - \alpha \mathbf{h}_\delta\}\}_{\Gamma_m} \right\|^2_\lambda + \left\| [\![\mathbf{h}_\delta]\!]_{\Gamma_m} \right\|^2_\lambda \right) . \tag{24}$$

Using the above quantities, we define our error estimate as follows:

$$est_\delta := \left\{ \sum_{i \in \mathcal{I}} \left[\sum_{T \in \mathcal{T}_{\delta i}} \left(\eta^2_{H,T} + \eta^2_{P,T} + \sum_{m \in \mathcal{M}_T} \left(\xi^m_{NC,T}\right)^2 \right) + \sum_{\sigma \in \mathcal{E}_\delta} \left(\xi^m_{F,\sigma}\right)^2 \right.\right.$$
$$\left.\left. + \sum_{m \in \mathcal{M}_i} \sum_{\lambda \in \Lambda_{m,i}} \left(\left(\xi^i_{P,\lambda}\right)^2 + J_{\delta,\lambda}(\mathbf{u}_\delta) \right) \right] \right\}^{\frac{1}{2}} . \tag{25}$$

Then, we can prove the following results, stating the equivalence between the error and the error estimator.

Theorem 1. *Let err be defined by* (17) *and est_δ by* (25)*. Then, there exists a constant $C^* > 0$ independent of δ such that*

$$err \leq C^* \, est_\delta .$$

Moreover, there exists a constant $C_* > 0$ independent of δ such that

$$est_\delta \leq C_* \left\{ \left| \left| \left| e_{\mathbf{p}} \right| \right| \right|^2 + C_{NC} \left[\max_{\sigma \in \mathcal{E}_\delta} \{1, h_\sigma\} \left(|||e_{\mathbf{h}}|||^2 + \|e_{\mathbf{u}}\|^2_U \right) + \max_{\substack{i \in \mathcal{I} \\ T \in \mathcal{T}_{\delta i}}} h^2_T \, \|f_i - f_T\|^2_T \right] \right\}^{\frac{1}{2}} ,$$

where f_T is the integral mean of f_i on a triangle $T \in \mathcal{T}_{\delta i}$ and

$$C_{NC} := \max_{T \in \mathcal{T}_\delta} \left(\frac{1 + h_{NC,T}}{1 - h_{NC,T}} \right) .$$

Proof. The proof is omitted since it follows the same techniques used in [3] with some straightforward extensions due to presence of the additional terms involved by the more general boundary conditions that require $\alpha \neq 0$. □

6. Adaptive algorithm

In this section we describe an adaptive algorithm exploiting the numerical estimate obtained above. The aim of the procedure is to reduce the error estimator under a given percentage τ_p of the initial estimator.

For each $i \in \mathcal{I}$, and for any triangle $T \in \mathcal{T}_{\delta,i}$, we define the local estimator $est_{\delta,T}$ as follows:

$$est_{\delta,T} := \left[\eta^2_{H,T} + \eta^2_{P,T} + \sum_{\sigma \in \mathcal{E}_\delta} (\xi_{F,\sigma})^2 + \sum_{m \in \mathcal{M}_T} \sum_{\lambda \in \Lambda_{m,i}} \left(\left(\xi^m_{NC,T} \right)^2 + \left(\xi^i_{P,\ell^m_T} \right)^2 + J_{\delta,\lambda \cap T}(\mathbf{u}_\delta) \right) \right]^{\frac{1}{2}}. \tag{26}$$

The adaptive algorithm we chose is based on the equidistribution [42] of the error on the triangles and is as follows:

1. choose an estimate reduction threshold $\tau_p \in (0.0, 1.0]$;
2. set $k = 0$, choose an initial mesh parameter δ_k and build $\mathcal{T}_{\delta_k}$ independently on each fracture and independently of the traces;
3. do
   SOLVE the problem on $\mathcal{T}_{\delta_k}$;
   ESTIMATE the error by computing est_{δ_k};
   MARK triangles $T \in \mathcal{T}_{\delta_k}$ such that $est_{\delta_k,T} > \dfrac{est_{\delta_k}}{\#\mathcal{T}_\delta}$ for refinement;
   REFINE marked triangles by a chosen constant factor c, obtaining $\mathcal{T}_{\delta_{k+1}}$;
   Increase k;
   while $est^2_{\delta_k} > \tau_p est^2_{\delta_0}$ and $0 \le k \le k_{\max}$.

The reader may refer to [2, 37] for a detailed description of how the minimization process can be performed using the conjugate gradient method, also in parallel form.

7. Numerical results

In the following sections we apply the adaptive algorithm previously described to some DFNs with variable complexity in order to assess the reliability of the numerical solution and its viability to realistic DFNs. First, we consider two test cases with known solution and compute the effectivity indices at each refinement step to verify it is independent of the mesh size, then we investigate the behaviour of the adaptive method on some more realistic test case.

In all the following tests, the partitions $\Lambda_{m,i}$ and $\Lambda_{m,j}$ for the discretization of $u_{\delta i}$ and $u_{\delta j}$ on trace Γ_m such that $\mathcal{I}_m = (i, j)$ are staggered uniform discretizations with a number of intervals that is the mean value of the number of intersections between Γ_m and the triangulations $\mathcal{T}_{\delta,i}$ and $\mathcal{T}_{\delta,j}$. The discrete spaces used for $\mathbf{h}_\delta$ and $\mathbf{p}_\delta$ are linear finite elements, and a discontinuous linear finite element discretization is used for $\mathbf{u}_\delta$ on the traces. Table 1 reports the parameters used for each of the following simulations: the parameter δ_0 is the minimum number of cells required on each fracture at the beginning of the refinement process, `CG-MaxIt` is the maximum number of iterations allowed for the conjugate gradient and `CG-ResTol` is the tolerance set on the relative residual.

Table 1: Parameters used for the simulations

	τ_p	δ_0	CG-MaxIt	CG-ResTol	c	k_{max}
Problem 1	0.025	100	80000	10^{-12}	3	20
Problem 2	0.025	100	80000	10^{-12}	3	20
Frac6	0.025	100	80000	10^{-12}	3	20
Frac86	0.025	100	80000	10^{-6}	3	20
Frac15K	0.025	100	80000	10^{-6}	3	30

7.1. Numerical Results for Problems with known solutions

The first set of tests aims at validating the equivalence relation between the error and the error estimator during the adaptive process. In order to compute the error, we consider two simple DFNs on which we can define a problem whose exact solution is known. We apply to these problems our adaptive algorithm and compare at each refinement iteration the error and the error estimator computing the *effectivity index*

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

and verify that it is independent of the mesh obtained by adaptive refinements. See [3, Section 7] for the same analysis performed with meshes obtained by uniform refinement.

7.1.1. Problem 1

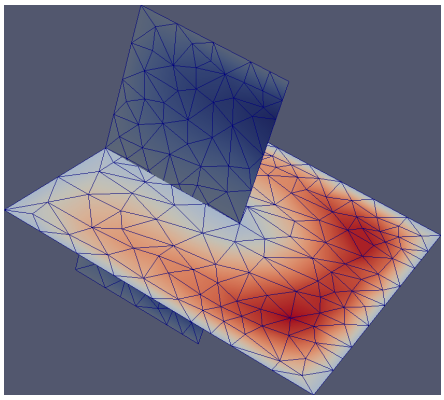
The geometry and the parameters of this test problem are described in detail in [34, Section 5.1]. In Figure 1 we report some of the meshes generated by the refinement algorithm, with the parameters reported in Table 1. We remark that in this test problem the presence of a known forcing function on the fractures induces a refinement in the whole domain, this will not be the same for the flow simulations in which the DFNs is considered impervious (Section 7.2).

In Figure 2a we report the convergence behaviour of the error and of the error estimator and the final rates of convergence α with respect to the total number of degrees of freedom $est_\delta \sim (\#\text{Dofs})^\alpha$ computed on the basis of the last three refinement iterations. We can clearly appreciate a parallel behaviour of error and error estimator. In Figure 2b we report the behaviour of the standard deviation of the local error estimators $est_{\delta,T}$, defined by

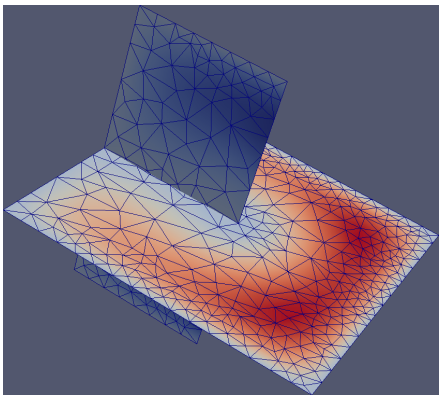
$$\sigma = \sqrt{\frac{\sum_{T\in\mathcal{T}_\delta}\left(est_{\delta,T} - \overline{est_\delta}\right)^2}{\#\mathcal{T}_\delta}}, \quad \text{being } \overline{est_\delta} = \frac{est_\delta}{\#\mathcal{T}_\delta}. \tag{27}$$

From the decreasing behaviour of this plot we can appreciate that the error has been efficiently equidistributed by the refinement algorithm. In Table 2 we report the most significant quantities to describe the refinement process: Ω-Cells is the total number of triangles on the fractures, Γ-Cells is the total number of sub-intervals on the traces for the discretization of the control variable $\mathbf{u}_\delta$, Dofs is the total numbers of degrees of freedom of the problem, ε is the effectivity index, σ is the standard deviation of the local estimator, CG-It is the number of conjugate gradient iterations performed and J

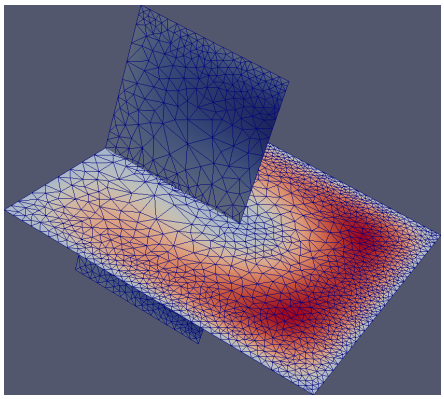
Figure 1: Problem 1. DFN with meshes at some refining steps.



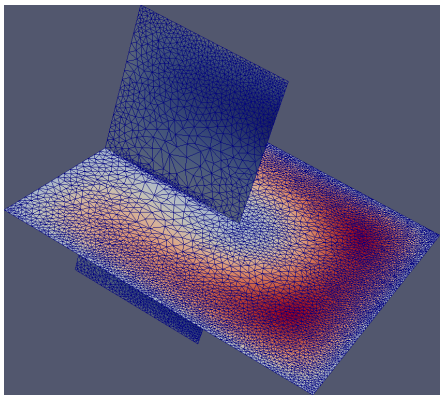
(a) Initial mesh



(b) Step 3

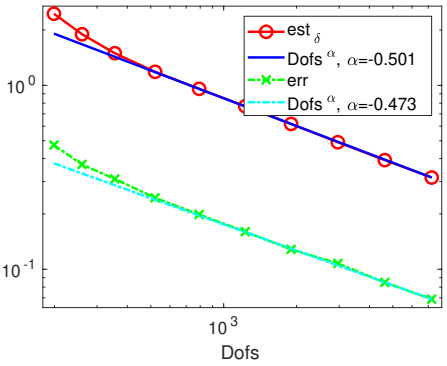


(c) Step 6

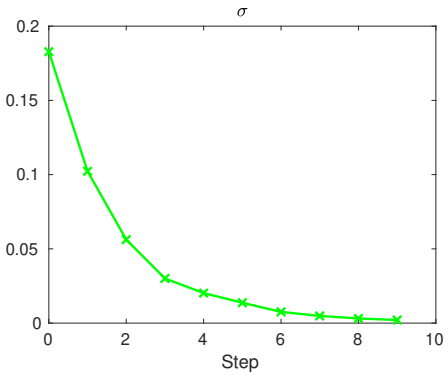


(d) Step 9

Figure 2: Problem 1. Plots



(a) Rates of convergence for error and error estimator



(b) Standard deviation of the estimator

Table 2: Problem 1. Data during the refinement process

Step	Ω-Cells	Γ-Cells	Dofs	ε	σ	CG-It	J
0	305	10	199	0.1933	1.83E-01	27	2.41E-05
1	417	10	260	0.1959	1.02E-01	27	3.13E-05
2	589	10	355	0.2073	5.62E-02	27	4.79E-05
3	891	12	521	0.2065	3.01E-02	33	2.99E-05
4	1393	15	792	0.2073	2.02E-02	43	1.93E-05
5	2215	19	1228	0.2081	1.37E-02	55	5.92E-06
6	3500	26	1899	0.2081	7.56E-03	79	1.61E-06
7	5556	28	2962	0.2193	4.84E-03	85	1.54E-06
8	8790	34	4628	0.2163	3.14E-03	106	7.87E-07
9	13871	52	7234	0.2180	2.09E-03	177	1.94E-07

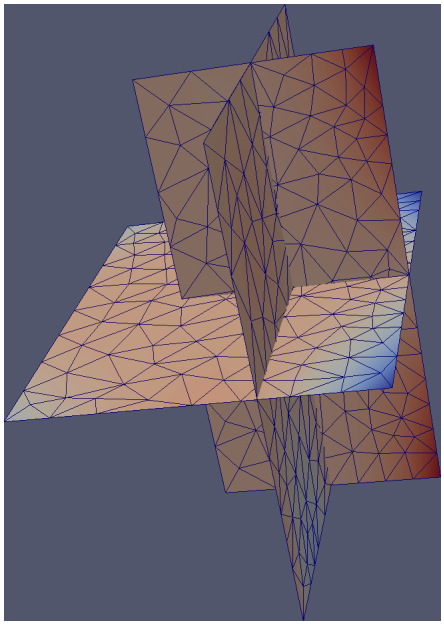
if the value of the functional at the last iteration. We highlight the very small variations of ε with respect to the large variations of number of cells and degrees of freedom.

7.1.2. Problem 2

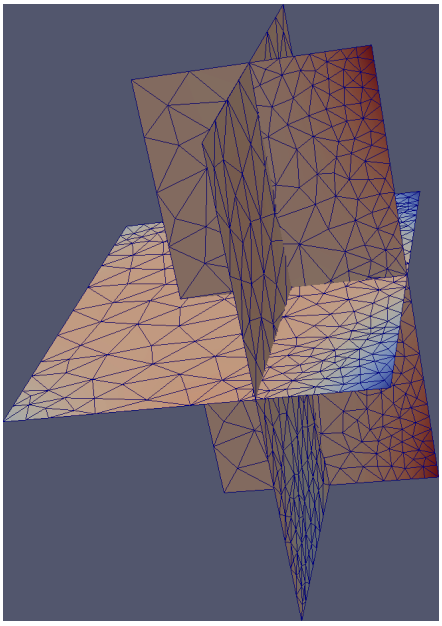
The geometry and the parameters of this test problem are described in detail in [18, Section 6.1]. In Figure 3, we report some of the meshes generated by the refinement process, with the parameters reported in Table 1. Similarly to the previous test, a refinement is induced also far from traces due to the presence of a non null forcing term.

In Figure 4a we report the convergence behaviour of the error and of the error estimator with the final rates of convergence (α), computed on the basis of the last three refinement iterations: again we have a very good agreement between them. In Figure 4b we report the behaviour of the standard deviation of the elemental error estimator, defined by (27). Like for the previous example, also in this case we see that the error is efficiently equidistributed. In Table 3 we report the same quantities reported

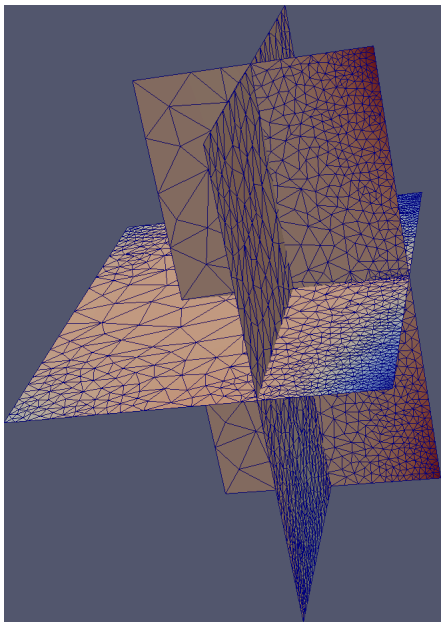
Figure 3: Problem 2. DFN with meshes at some refining steps



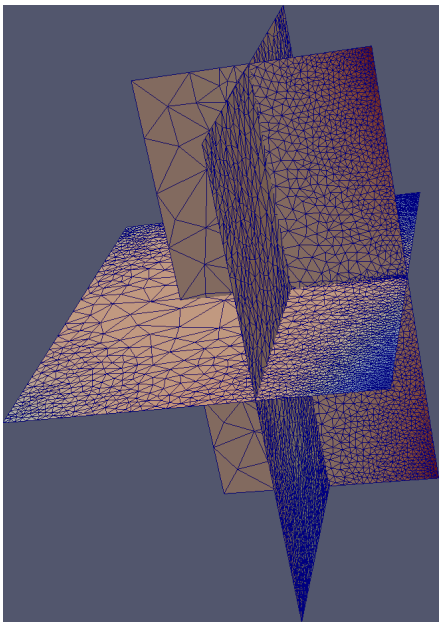
(a) Initial mesh



(b) Step 3

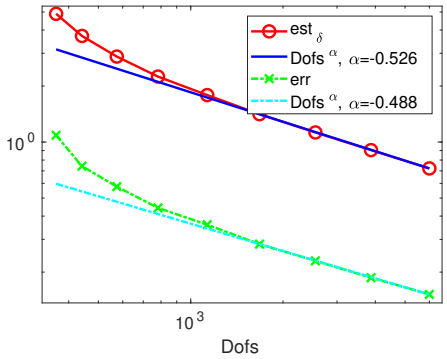


(c) Step 6

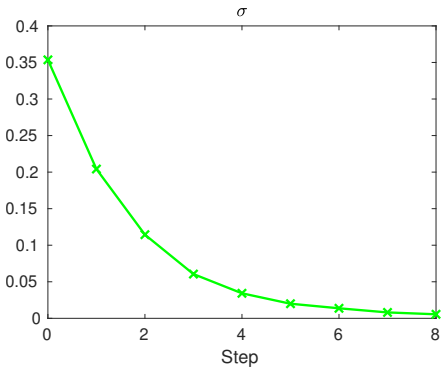


(d) Step 8

Figure 4: Problem 2. Plots



(a) Rates of convergence for error and error estimator



(b) Standard deviation of the estimator

Table 3: Problem 2. Data during the refinement process

Step	Ω-Cells	Γ-Cells	Dofs	ε	σ	CG-It	J
0	458	47	364	0.2225	3.54E-01	151	8.45E-04
1	594	47	442	0.1997	2.04E-01	150	4.50E-04
2	842	47	574	0.2000	1.14E-01	150	6.81E-04
3	1212	49	782	0.1970	6.05E-02	160	3.27E-04
4	1867	51	1132	0.2011	3.42E-02	164	3.19E-04
5	2902	54	1676	0.2003	2.00E-02	171	3.39E-04
6	4566	58	2550	0.2041	1.37E-02	194	3.04E-04
7	7095	70	3873	0.2063	8.07E-03	232	3.02E-04
8	11209	82	6001	0.2104	5.52E-03	283	2.68E-04

in Table 2 to describe the refinement process. We observe again that the effectivity index is independent of the meshsize.

7.2. Numerical Results on DFNs

This second set of tests is used to show the behaviour of the adaptive algorithm when applied to DFN flow simulations with increasing geometrical complexities. In Table 4 we summarize the geometrical features of the networks that we consider. In these cases, we do not impose any forcing term, and the flux is driven by the presence of two Dirichlet boundary conditions imposed on two sides of two different fractures, while homogeneous Neumann boundary conditions are imposed on all the other boundaries.

7.2.1. Frac6

The first DFN we consider is made up of six fractures that intersect each other orthogonally, as depicted in Figure 5, where we display the meshes generated by the

Table 4: Number of fractures and traces for each DFN

Problem	#Fractures	#Traces	Min #Tr	Max #Tr	$\frac{\#Traces}{\#Fractures}$
Frac6	6	6	1	4	1
Frac86	86	159	1	11	1.85
Frac15K	15102	27952	1	243	1.85

Table 5: Frac6. Data during the refinement process

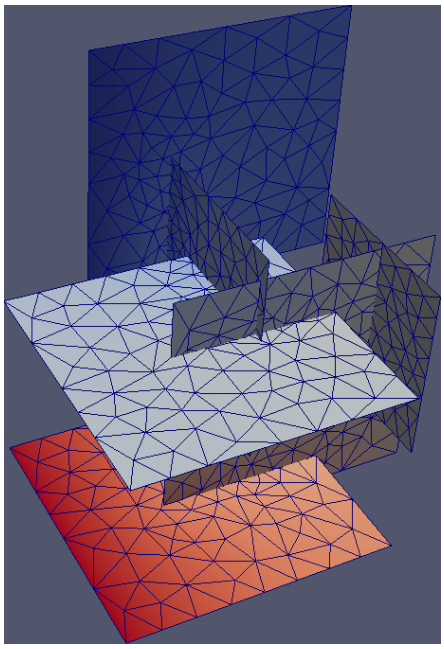
Step	Ω-Cells	Γ-Cells	Dofs	σ	CG-It	J
0	873	68	653	3.91E-01	136	1.04E-01
1	1097	76	783	2.68E-01	156	5.75E-02
2	1461	90	998	2.04E-01	192	3.47E-02
3	1956	112	1297	1.43E-01	225	2.05E-02
4	2689	140	1728	1.00E-01	295	1.38E-02
5	3783	173	2353	7.92E-02	388	1.14E-02
6	5339	237	3271	6.37E-02	548	7.28E-03
7	7836	295	4657	4.16E-02	605	5.37E-03
8	11111	402	6525	2.96E-02	922	4.18E-03
9	16161	512	9295	1.96E-02	1100	2.77E-03
10	23683	689	13437	1.45E-02	1365	1.79E-03
11	34569	977	19506	1.00E-02	1905	1.23E-03
12	50711	1295	28256	7.08E-03	2409	7.78E-04

adaptive process, with the parameters reported in Table 1. As expected, since the forcing term is null, the refinement is mainly concentrated in the neighbourhood of traces, while in the interior of fractures the refinement follows the flux stream: the reader can observe a non uniform refinement on the highest horizontal fracture, being the flux essentially flowing between the two intersecting fractures. In Figure 6a we report the convergence behaviour of the error estimator with the final rate of convergence (α), computed on the basis of the last three refinement steps. In Figure 6b we report the behaviour of the standard deviation of the elemental error estimator. In Table 5 we report the characteristic quantities of the refinement process as in the previous cases.

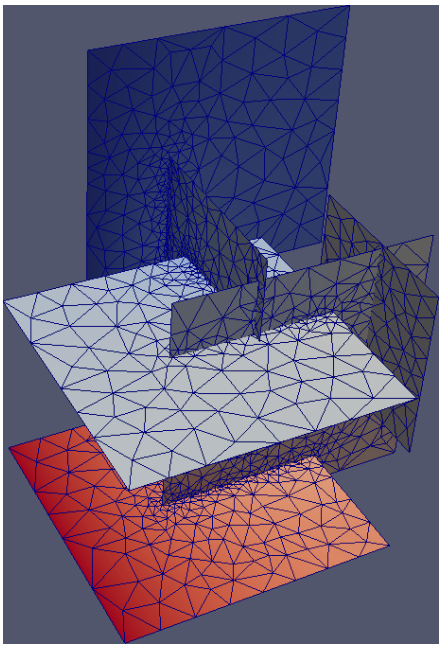
7.2.2. Frac86

This second network has 86 fracture and is more dense than the previous one, with an average of 1.85 traces per fracture (see Table 4). As for the previous tests, we show in Figure 7 the convergence graphs and in Table 6 the parameters describing the behaviour of the adaptive process. The results are in agreement with the previous considerations, moreover, from Table 6, we can also appreciate a good effect of the adaptively refined meshes on the conjugate gradient solver, being the number of iterations required to satisfy the relative tolerance (column CG-It) quite stable after the first faur iterations also with an increasing number of degrees of freedom of the control variable (column Γ-Cells).

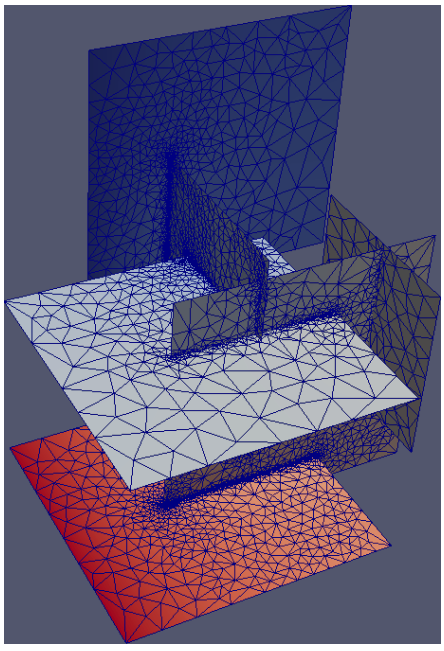
Figure 5: Frac6. DFN with meshes at some refining steps



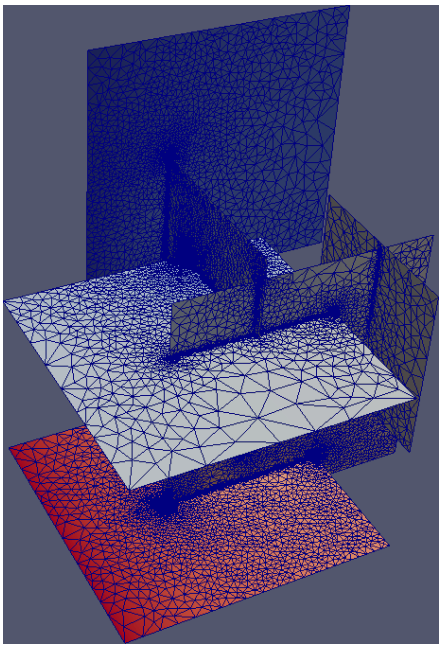
(a) Initial mesh



(b) Step 4

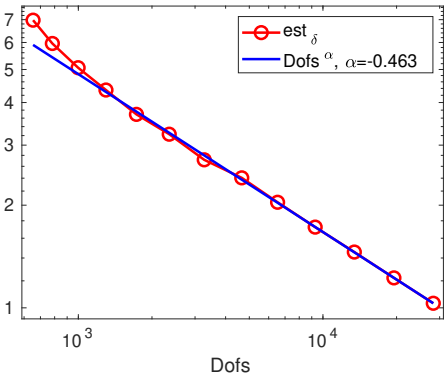


(c) Step 8

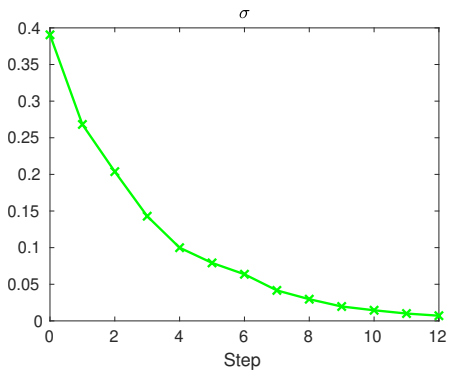


(d) Step 12

Figure 6: Frac6. Plots

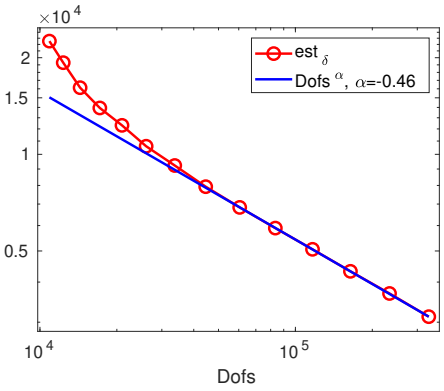


(a) Rates of convergence for error and error estimator

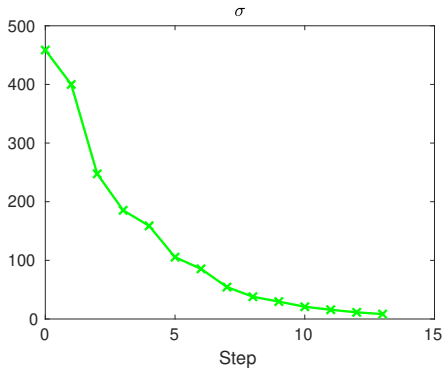


(b) Standard deviation of the estimator

Figure 7: Frac86. Plots



(a) Rates of convergence for error estimator



(b) Standard deviation of the estimator

Figure 8: Frac15K. Full DFN

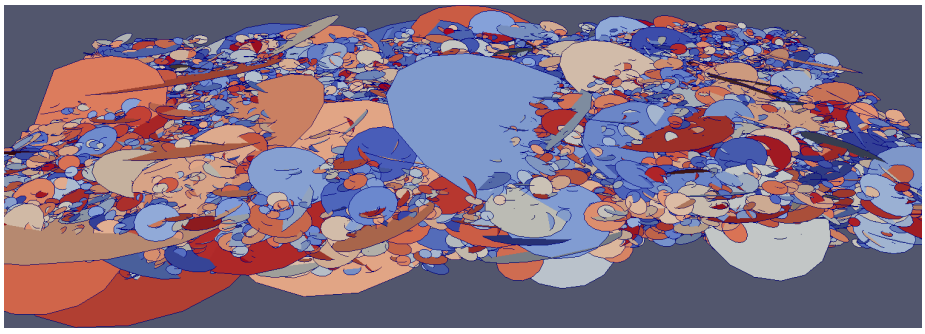


Table 6: Frac86. Data during the refinement process

Step	Ω-Cells	Γ-Cells	Dofs	σ	CG-It	J
0	13765	1354	10890	4.59E+02	9722	7.04E-01
1	16093	1476	12335	4.00E+02	10257	6.58E-01
2	19373	1636	14362	2.47E+02	5508	4.99E-01
3	24050	1826	17156	1.85E+02	2909	3.87E-01
4	30459	2073	20950	1.59E+02	2206	3.04E-01
5	39252	2369	26043	1.05E+02	2157	2.52E-01
6	52592	2788	33705	8.55E+01	1995	1.93E-01
7	71656	3361	44557	5.44E+01	1712	1.59E-01
8	99618	4240	60530	3.79E+01	1619	1.23E-01
9	140479	5345	83449	2.97E+01	1661	9.09E-02
10	199936	6919	116680	2.09E+01	1742	6.84E-02
11	285895	8979	164218	1.58E+01	1719	5.00E-02
12	411647	11972	233611	1.13E+01	1813	3.76E-02
13	593352	15760	332707	8.38E+00	2270	2.56E-02

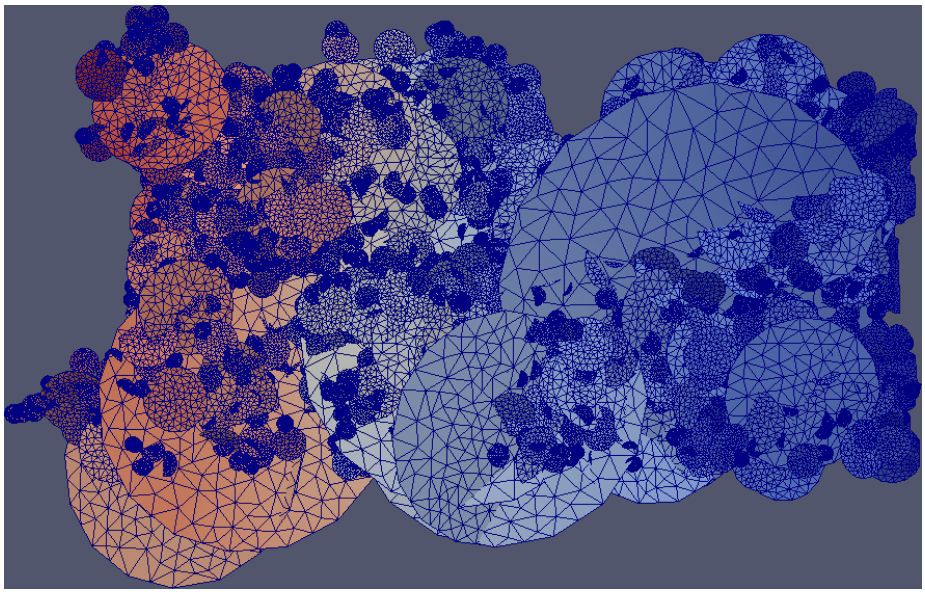
7.2.3. Frac15K

The last network that we analyse is depicted in Figure 8 and it is characterized by many geometrical complexities, such as small fractures intersecting very larger ones, almost parallel traces or traces intersecting with very small angles: for a further discussion of these issues we refer to [43]. In Figure 9a we report the initial mesh on a portion of the fractures of the DFN, whereas in Figure 9b the mesh obtained after 8 refinement steps on the same fractures is depicted. We note that in this mesh we can see the refinement around the traces, this is especially evident around the intersections of the fractures present in the picture with fractures not included in the picture. In Figure 10a a detail of this mesh is reported, and we can notice that the mesh is not refined around all traces in the same way but only around some of the traces that should be that with a more relevant flux. In Figure 10b a closer view of the mesh is reported and this non-uniform refinement around the traces is even more evident. In Table 7 we report the most significant quantities of the problem during the adaptive refinement. We notice that, although the number of the control variables is increasing during refinement, the number of conjugate gradient iterations decreases after step 7, this denotes a positive effect of the refinement on the conditioning of the discrete problem.

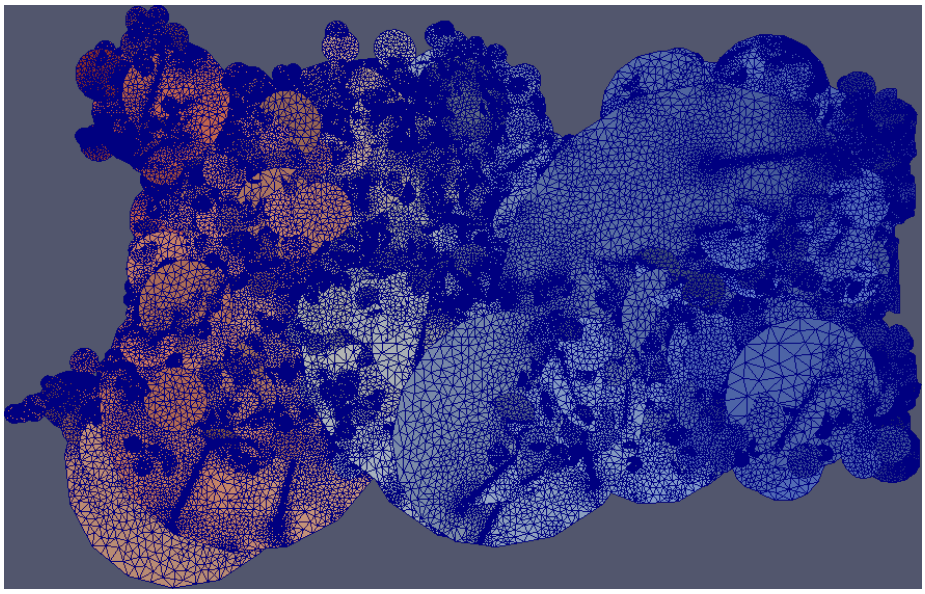
7.2.4. Frac15K, adaptive and uniform refinement

In this section we compare the convergence behaviour of the adaptive mesh refinement and of the uniform mesh refinement when applied to the same DFN of the previous subsection. The initial mesh on each fracture contains about 16 elements. During the solution process the maximal number of CG iteration is always reached. In Figure 12a we report the initial mesh generated on the DFN for adaptive and uniform refinement, and in Figure 12b a detail of the mesh is reported from where we can see the chaotic intersections of fractures. In Figure 13a we compare the behaviour of the error estimator during the adaptive and uniform refinement process. We also report the

Figure 9: Frac15K. DFN with meshes at some refining steps

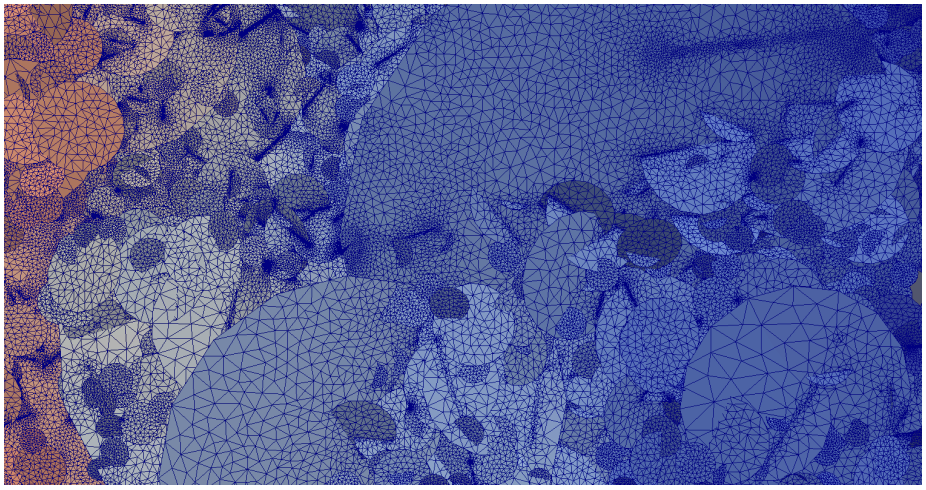


(a) Initial mesh

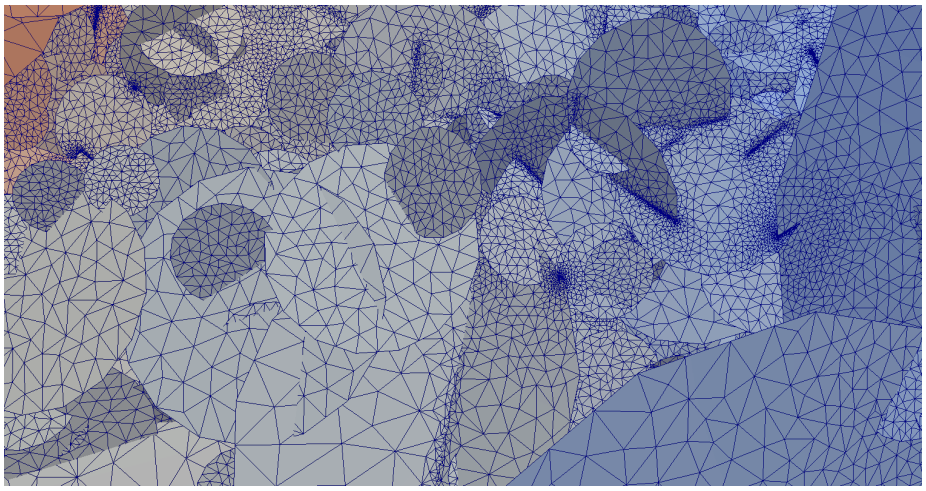


(b) Step 8

Figure 10: Frac15K. Details of the mesh, refining step 8

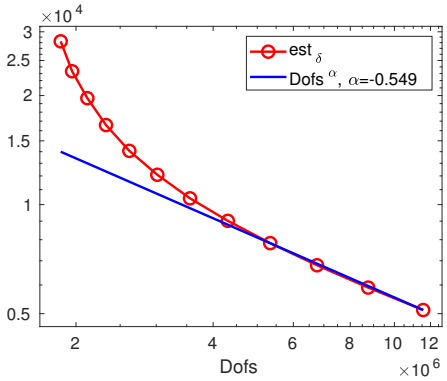


(a)

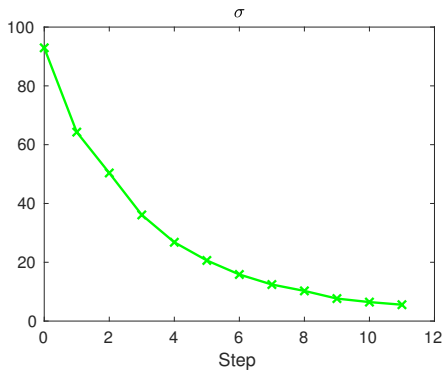


(b)

Figure 11: Frac15K. Plots



(a) Rates of convergence for error estimator



(b) Standard deviation of the estimator

Table 7: Frac15K, Data during the refinement process

Step	Ω-Cells	Γ-Cells	Dofs	σ	CG-It	J
0	2417952	211478	1851290	9.29E+01	80000	3.93E-01
1	2593115	221175	1961046	6.43E+01	80000	4.11E-01
2	2842022	234468	2118259	5.04E+01	80000	4.06E-01
3	3182041	251833	2327579	3.61E+01	80000	3.90E-01
4	3660033	274509	2619066	2.68E+01	80000	3.68E-01
5	4329339	302330	3016777	2.06E+01	80000	3.42E-01
6	5256393	336908	3558969	1.58E+01	80000	3.04E-01
7	6557532	380421	4307937	1.24E+01	77055	2.69E-01
8	8364479	436335	5337434	1.03E+01	74740	2.32E-01
9	10893729	507994	6762737	7.65E+00	72113	1.99E-01
10	14462152	602244	8757293	6.46E+00	71689	1.68E-01
11	19527267	726470	11565176	5.56E+00	70432	1.39E-01

Table 8: Frac15K, Data during adaptive refinement process in Section 7.2.4

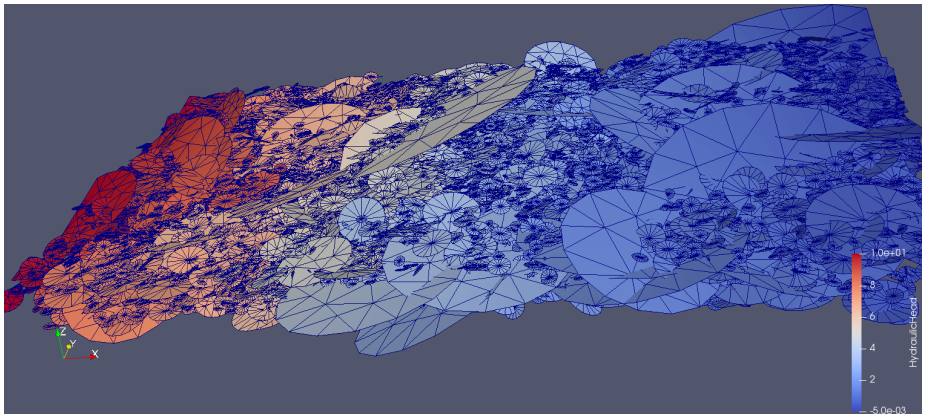
Step	Ω-Cells	Γ-Cells	Dofs	σ	est_δ	J
0	242855	145888	549087	6.64E+02	6.98E+04	2.79E-01
1	261204	145648	557797	3.68E+02	4.52E+04	2.99E-01
2	298859	146330	578051	2.18E+02	3.61E+04	3.50E-01
3	355061	148626	612975	1.43E+02	2.99E+04	3.71E-01
4	435454	152500	662769	1.01E+02	2.55E+04	3.95E-01
5	547776	158279	733433	7.92E+01	2.22E+04	4.00E-01
6	703220	166511	830831	5.33E+01	1.92E+04	4.05E-01
7	929697	178259	972138	3.93E+01	1.67E+04	3.92E-01
8	1255878	194066	1172353	2.83E+01	1.46E+04	3.76E-01
9	1730104	215214	1459305	2.20E+01	1.27E+04	3.51E-01
10	2416585	243498	1868478	1.57E+01	1.11E+04	3.23E-01
11	3417205	280966	2456115	1.26E+01	9.62E+03	2.91E-01
12	4864293	330821	3294938	9.92E+00	8.34E+03	2.56E-01
13	6971745	396246	4499103	7.56E+00	7.22E+03	2.20E-01
14	10032775	483200	6227824	5.85E+00	6.22E+03	1.86E-01
15	14500876	599711	8725258	5.98E+00	5.36E+03	1.54E-01
16	20935334	752245	12284501	4.43E+00	4.60E+03	1.26E-01
17	30236802	956686	17388891	4.06E+00	3.95E+03	1.01E-01

slopes of the fitting of the error estimators and we can see that the adaptive refinement is always very close to the optimal rate of convergence using linear finite elements, whereas the uniform refinement is displaying a reduced rate of convergence. From Figures 13a-13c we can appreciate the very good behaviour of the adaptive method with respect to uniform refinement in term of error reduction, as well as in term of equidistribution of the error and functional reduction.

We note that the reduced rate of convergence of the estimator in the case of uniform refinement can be ascribed to the global low regularity of the solution, that does not belong to $\mathrm{H}^2(\Omega)$. In particular, the asymptotic rates of convergence are compatible with the regularity of the solution, that belongs to $\mathrm{H}^{\frac{3}{2}-\epsilon}(\Omega)$, $\forall \epsilon > 0$, due to the lack of smoothness around trace tips. This behaviour cannot be circumvented also using meshes conforming to the intersections. Applying adaptive approaches, we are performing a non-linear approximation [44] that depends on the higher Besov regularity of the solution, and this is compatible with the maximal rate of convergence possible with linear finite elements ($\alpha \sim 0.5$), that is well approximated in Figure 13a. Furthermore, we note that in the pre-asymptotic regime the error decrease is much stronger in the adaptive approach with respect to the uniform refinement.

Figure 13d compares the computational time required on each uniform mesh with the cumulative time required to reach and perform each step of the adaptive process. We remark that in the current implementation we recompute all the trace-mesh intersections: this can be avoided in case of nested refined meshes. In Tables 8-9 we report the values of the most significant quantities to describe the refinement process.

Figure 12: Frac15K. Details of the mesh used for the test in Section 7.2.4



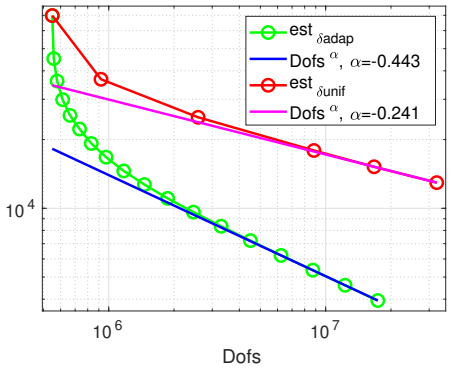
(a)

(b)

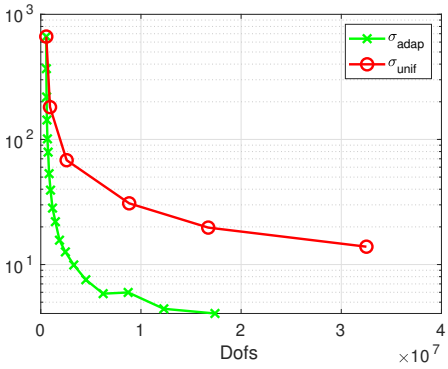
Table 9: Frac15K, Data during uniform refinement process in Section 7.2.4

Step	Ω-Cells	Γ-Cells	Dofs	σ	est_δ	J
0	242855	145888	549087	6.64E+02	6.98E+04	2.79E-01
1	894372	168924	921096	1.81E+02	3.67E+04	3.71E-01
2	3699937	243136	2581970	6.82E+01	2.50E+04	3.99E-01
3	15084448	404039	8826748	3.08E+01	1.79E+04	3.87E-01
4	29978097	542013	16718991	1.97E+01	1.52E+04	3.53E-01
5	60191364	740664	32510129	1.39E+01	1.30E+04	3.06E-01

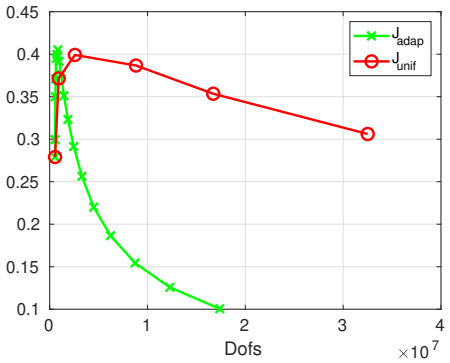
Figure 13: Frac15K. Comparison between adaptive and uniform refinements



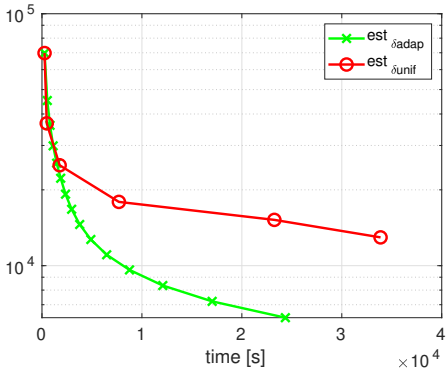
(a) Rates of convergence for error estimator



(b) Standard deviation of the estimator



(c) Comparison of the behaviour of the functional J



(d) Comparison of the behaviour of the estimator vs. total computing time

8. Conclusions

An adaptive algorithm based on a residual a posteriori error estimate is devised for the computation of the hydraulic head distribution in a Discrete Fracture Network. The approach reduces the error under a certain percentage of the error at the first step of the adaptive process and equidistributes it in space. Numerical tests verify the equivalence of the true error and the error estimate and show the capability to capture the relevant features of the flux automatically.

References

[1] 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.

[2] 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.

[3] S. Berrone, A. Borio, S. Scialò, A posteriori error estimate for a PDE-constrained optimization formulation for the flow in DFNs, SIAM J. Numer. Anal. 54 (1) (2016) 242–261. doi:10.1137/15M1014760.

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

[5] 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.

[6] 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.

[7] S. Berrone, C. Canuto, S. Pieraccini, S. Scial, 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

[8] 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:http://dx.doi.org/10.1029/WR026i003p00479.

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

[10] G. Pichot, B. Poirriez, J. Erhel, J.-R. de Dreuzy, A Mortar BDD method for solving flow in stochastic discrete fracture networks, in: Domain Decomposition Methods in Science and Engineering XXI, Springer, 2014, pp. 99–112, lecture Notes in Computational Science and Engineering.

[11] 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:http://dx.doi.org/10.1080/00036811.2010.495333.

[12] 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:http://dx.doi.org/10.1137/100804383.

[13] M. Karimi-Fard, L. J. Durlofsky, Unstructured adaptive mesh refinement for flow in heterogeneous porous media, in: ECMOR XIV-14th European conference on the mathematics of oil recovery, 2014. doi:10.3997/2214-4609.20141856.

[14] 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:http://dx.doi.org/10.1137/130942541.

[15] 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

[16] 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

[17] B. Noetinger, A quasi steady state method for solving transient darcy flow in complex 3d fractured networks accounting for matrix to fracture flow, Journal of Computational Physics 283 (2015) 205 – 223.

[18] 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.

[19] M. F. Benedetto, S. Berrone, A. Borio, The Virtual Element Method for underground flow simulations in fractured media, in: Advances in Discretization Methods, Vol. 12 of SEMA SIMAI Springer Series, Springer International Publishing, Switzerland, 2016, pp. 167–186. doi:10.1007/978-3-319-41246-7_8.

[20] 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.

[21] 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.

[22] 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:http://dx.doi.org/10.1016/j.advwatres.2013.04.001.

[23] L. Formaggia, P. Antonietti, P. Panfili, A. Scotti, L. Turconi, M. Verani, A. Cominelli, Optimal techniques to simulate flow in fractured reservoir, in: ECMOR XIV-14th European conference on the mathematics of oil recovery, 2014.

[24] 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:http://dx.doi.org/10.2516/ogst/2013192.

[25] 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.

[26] A. Fumagalli, E. Keilegavlen, S. Scialò, Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations, J. Comput. Phys.In press. doi:10.1016/j.jcp.2018.09.048.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] J. Jaffré, J. E. Roberts, Modeling flow in porous media with fractures; discrete fracture models with matrix-fracture exchange, Numerical Analysis and Applications 5 (2) (2012) 162–167.

[33] 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.

[34] 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

[35] 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.

[36] 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.

[37] S. Berrone, S. Pieraccini, S. Scialò, F. Vicini, A parallel solver for large scale DFN flow simulations, SIAM J. Sci. Comput. 37 (3) (2015) C285–C306. doi:10.1137/140984014.

[38] S. Berrone, Adaptive discretization of stationary and incompressible navier-stokes equations by stabilized finite element methods, Computer Methods in Applied Mechanics and Engineering 190 (34) (2001) 4435–4455, cited By 35. doi:10.1016/S0045-7825(00)00327-3.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-0035946979&doi=10.1016%2fS0045-7825%2800%2900327-3&partnerID=40&md5=d63c66c0913fea47a1bb8e95752fc9b7

[39] S. Berrone, M. Marro, Space-time adaptive simulations for unsteady navier-stokes problems, Computers and Fluids 38 (6) (2009) 1132–1144, cited By 24. doi:10.1016/j.compfluid.2008.11.004.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-62549120315&doi=10.1016%2fj.compfluid.2008.11.004&partnerID=40&md5=9e39bc23ae6e41e1976b03b5ab6ea4ac

[40] S. Berrone, M. Marro, Numerical investigation of effectivity indices of space-time error indicators for navier-stokes equations, Computer Methods in Applied Mechanics and Engineering 199 (25-28) (2010) 1764–1782, cited By 4. doi:10.1016/j.cma.2010.02.004.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-77952548156&doi=10.1016%2fj.cma.2010.02.004&partnerID=40&md5=2d6762a49d4d0ca25981c0b0a2726b0a

[41] S. Berrone, V. Garbero, M. Marro, Numerical simulation of low-reynolds number flows past rectangular cylinders based on adaptive finite element and finite volume methods, Computers and Fluids 40 (1) (2011) 92–112, cited By 11. doi:10.1016/j.compfluid.2010.08.014.
URL https://www.scopus.com/inward/record.uri?eid=2-s2.0-78549276803&doi=10.1016%2fj.compfluid.2010.08.014&partnerID=40&md5=e32e181ba6e3b7ff4615c796ca1606b9

[42] J. Medina, M. Picasso, J. Rappaz, Error estimates and adaptive finite elements for nonlinear diffusion-convection problems, Mathematical Models & Methods in Applied Sciences 6 (5) (1996) 689–712.

[43] S. Berrone, S. Pieraccini, S. Scialò, Towards effective flow simulations in realistic discrete fracture networks, J. Comput. Phys. 310 (2016) 181–201. doi:10.1016/j.jcp.2016.01.009.

[44] R. A. DeVore, Nonlinear approximation, Acta Numerica 7 (1998) 51150. doi:10.1017/S0962492900002816.