MOX–Report No. 28/2014

Optimal techniques to simulate flow in fractured reservoirs

Antonietti, P; Panfili, P.; Scotti, A.; Turconi, L. ;
Verani, M.; Cominelli,A.; Formaggia,L.

MOX, Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox@mate.polimi.it http://mox.polimi.it

Optimal techniques to simulate flow in fractured reservoirs

P. Antonietti[♯] P. Panfili[†] A. Scotti[♯] L. Turconi[◦]
M. Verani[♯] A. Cominelli[†] L. Formaggia[♯]

July 2014

[♯] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica
Politecnico di Milano, Italy

[†] eni spa- exploration & production division
Via Emilia, 1, 20097 San Donato Milanese (MI), Italy

[◦] MOXOFF S.r.l.
Via d'Ovidio Francesco, 3, 20131 Milan, Italy

Abstract

Simulation of multiphase flow in fractured reservoir is a computational challenge. A key issue is the effective coupling between flow in the porous matrix and in the fracture network. It requires computational grids honouring as much as possible the fracture geometry without degenerated/distorted elements. Standard techniques may degrade efficiency and are not a foolproof solution. Moreover, two point flux approximation (TPFA) demands a good quality of the mesh to mitigate discretization error. In this work compare two different approaches. The first one has been proposed by B.T Mallison et al. in 2010. The second method we consider is the one originally proposed by H. Mustapha, in 2009. We evaluate the two techniques by means of 2D synthetic problems based on realistic discrete fracture networks. Steady state and unsteady state simulations are performed using TPFA. We also present results obtained with computational methods based on coupling the fracture network with mimetic finite differences or extended mixed finite elements. The latter two approaches, even though more complex, are more robust with respect to mesh geometry and can be beneficial for the problem at hand.

1 Introduction

The simulation of fluid flow in fractured porous media has received an increasing attention in the last decade because of its relevance in several important applications: oil reservoir, exploitation of geothermal fields, CO2 sequestration, as well as the study of the safety of nuclear waste long term

disposal. Relevant monographs on the subject are [Brooks and Corey, 1964, Adler et al., 2012, Sahimi, 2012]. The numerical modeling of fluid flow in fractured porous media poses several challenging difficulties. Indeed, the presence of fractures (and faults) may affect greatly the effective permeability and storage capacity of the medium at the typical space resolution used by computational models. Since their distribution is normally not uniform, but it presents preferential directions, they also introduce anisotropy. Moreover, the topology of the network of fractures has an important influence on the upscaled properties. Fracture distribution in subsurface formations usually displays significant variations in connectivity and size over the formation. Large and strongly connected fractures are typically located near bedding planes and fault zones, while small and disconnected fractures are usually located away from those regions. Traditional approaches based on heuristic modification of the permeability or semi-empirical transfer functions in standard dual porosity techniques, see [Warren and Root, 1963, Barenblatt and Zheltov, 1960] and [Arbogast et al., 1990], may be insufficient since a complex fracture distribution makes it impossible to compute transfer function analytically. Numerical upscaling procedures based on realistic fracture characterization need to be applied to compute transfer functions accurately. Numerical upscaling techniques for fractured media are based on the solution of local problems at grid blocks level, see for instance [Durlofsky, 2003, Karimi-Fard et al., 2006, Gong and Durlofsky, 2008] and [Lim et al., 2009]. This approach is a very challenging task both from the geometrical and the computational point of view. Fracture networks are modeled as a set of intersecting one co-dimensional manifolds immersed in the porous matrix. Flow inside the fracture system may be computed by the solution of a reduced model for Darcy's flow (even if extension to non-Darcian flow is possible, [Choi et al., 1997]), where an additional source term is added to account for the coupling with the porous matrix. In the latter, a Darcy's model may be used, where the coupling with the fractures is modeled through interface conditions.

Several coupling approaches are possible. For instance, the fracture and matrix system may be coupled using mortar techniques, as in [Mustapha, 2014] or [Saas et al., 2005]. Alternatively, a computational grid conforming to the fracture can be generated, so that the discretization of the fracture network coincides with a set of faces/edges of the computational grid. However, the generation of a conforming grid is not an easy task and typically gives rise to unstructured triangular or tetrahedral grids.

Interesting approaches to generate conforming grids are illustrated in the works by [Mustapha et al., 2011] [Mustapha and Dimitrakopoulos, 2011] [Mustapha, 2012] and in [Mallison et al., 2010]). Starting from the fact that the network of fractures is usually known only approximately, these techniques introduce small geometrical perturbations in the fracture geometry, aimed at obtaining conforming meshes with a restrained number of cells and good quality geometrical features. Indeed, good quality meshes are of paramount importance to guarantee, for example, the convergence to the correct physical solution of the two-points finite volume (FV) schemes widely used in oil industry. While, computational

efficiency requires to avoid the use of extremely fine meshes.

Alternatively, conforming grids may be generated by intersecting the fracture network with a corner point grid. This gives rise to a mesh of irregular polygons. Again, the fracture network is approximated by a subset of the polygon faces. A methodology well suited for polygonal grids with good stability and convergence properties is the mimetic finite difference (MFD) method (see, e.g., [Kuznetsov et al., 2004, Brezzi et al., 2005a, Brezzi et al., 2005b] for more details). It can be naturally employed on very general decomposition made of (possibly non convex) polygons which do not have to fulfill matching conditions. Thanks to such a flexibility, the MFD method has been rapidly applied to a wide range of problems including porous media flows and oil reservoir simulations (see, e.g., [Lipnikov et al., 2008, Guevara-Jordan and Arteaga-Arispe, 2007, Alpak, 2010]). A recent work where it has been coupled with a simple fracture model is [Nielsen et al., 2012]. A further alternative is to let the fracture discretization be independent from the underlying grid used for the porous matrix. A strategy of this type is the extended finite element (XFEM) method, whose application to Darcy problem using mixed finite elements has been pioneered in [D'Angelo and Scotti, 2012].

In this work we will focus on two schemes for the generation of meshes conforming with the fractures, which, for the sake of simplicity, we will denote in the following as Mustapha and Mallison algorithm, respectively. Then, after introducing the equations governing the fluid flow in fractured porous media, we will give some detail the construction of two discretization schemes based on the mimetic finite difference (MFD) the extended finite element (XFEM) method, respectively, as well as recalling the basic elements of a two-point finite volume (FV) method. The main part of the work will be devoted to numerical experiments. In particular, we will first give an extensive comparison of the performance of Mustapha and Mallison algorithms on a set of significant test cases. Then, we will compare the efficacy of MFD and XFEM discretizations schemes to approximate a simple benchmark problem.

2 Algorithms for conforming mesh generation

A possible approach to perform numerical simulations in fractured reservoirs is to use computational grids that are *conforming*, i.e., *aligned*, to the fracture network. In this section we review the two algorithms we have considered in this work. These techniques aim at building good quality meshes with a restrained number of cells. However, they need to apply some modifications to the fracture network, to ensure a global mesh quality.

2.1 Mustapha algorithm

Mustapha algorithm has been introduced in [Mustapha et al., 2011] [Mustapha and Dimitrakopoulos, 2011] [Mustapha, 2012], and it is based on constrained Delaunay triangulation algorithms. The key idea is to introduce a suitable decision strategy (called "Gabriel

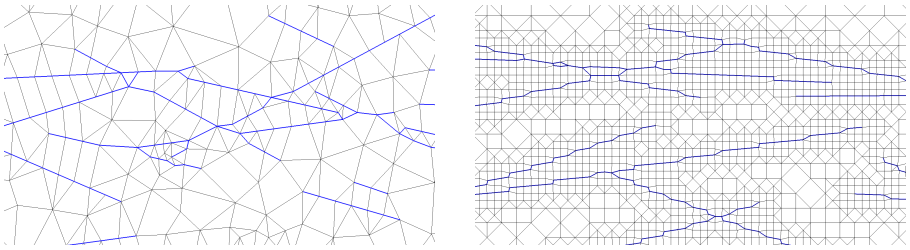

Figure 1: Examples of meshes built with two different gridding algorithms. On the left the one proposed by Mustapha that starts from an unstructured triangulation. On the right the Mallison method, starting from a structured Cartesian grid.

criterion") to select a part of the fracture network to which triangulation can be constrained without leading to an excessive degradation in cells quality, or excessively fine grids. At the end of the meshing procedure, the remaining parts of the fracture network will be approximated employing the existing edges/faces of the triangulation. As a result slight modifications of the fracture network are allowed to guarantee the overall quality of the mesh cells.

Delaunay triangulations are widely used in numerical simulations to discretize complex domains, since simplicial elements offer a great flexibility in the description of complex geometries, a reference is the monograph by [Cheng et al., 2012]. The constrained Delaunay algorithm allows to force a triangulation to be conforming with specified constraints, such as particular nodes or segments, for example. However, due to the typical complexity of fracture networks, it is not easy to employ these techniques to build a mesh which is aligned to the fractures. In two-dimensions, a very refined grid is usually needed to obtain shape regular elements in the regions close the fractures. In three-dimensions, the problem becomes even harder: the overall number of constraints could lead to a failure of the algorithm. Moreover, the standard algorithm may generate a large number of elements, with an unacceptable increase of the computational cost.

Before providing some more details of the Mustapha algorithm, we introduce some notation, focusing, for simplicity, on a two-dimensional setting. A mesh is called a Delaunay triangulation if the circumscribing circle of each triangular cell does not contain any other mesh node. A mesh edge is a Delaunay edge if it belongs to a triangle of a Delaunay triangulation. A conforming Delaunay triangulation is a constrained Delaunay triangulation in which every constrained edge is a Delaunay edge, cf. Figure 2 (left). A mesh edge is a Gabriel edge if it satisfies the empty-circle properties, i.e. the circle having the edge as diameter does not contain any mesh node in its interior, see Figure 2 (right). If every constrained edge is a Gabriel edge, the resulting mesh is called conforming Gabriel triangulation. We observe that the Gabriel property implies the Delaunay property, from this it follows that each conforming Gabriel triangulation is

a conforming Delaunay triangulation. After these definitions we are ready to

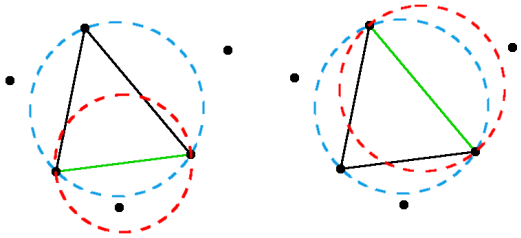

Figure 2: Example of Delaunay triangles together with a Delaunay (left) edge and Gabriel (right) edge. The black dots represent the mesh points.

introduce the main steps of the algorithm.

Assuming that the fracture can be represented with a segment (or a bounded number of segments), as a first step the fracture network is discretized. Then, each fracture segment is subdivided into a set of fracture edges according to a given (user dependent) mesh size $h_{\text{frac}} > 0$. As a second step, the Gabriel criterion is used to select a set of fracture edges that is then used as set of constraints for the constrained Delaunay algorithm. The reason of this approach is that, in a 2D setting, employing as a constraint only the set of fracture edges satisfying the Gabriel criterion guarantees that the resulting Delaunay grid satisfies a good quality test. In the final step of the algorithm, the edges of the Delaunay triangulation that has been generated are then used to suitably approximate the fracture edges that did not satisfied the Gabriel criterion and were not included as constraints. Clearly, this approach can lead to slight modifications of the fracture network, and such an issue should be considered as a possible drawback of the method.

2.2 Mallison algorithm

The second strategy we consider has been proposed by [Mallison et al., 2010]. The first step consists of generating a uniform structured triangular mesh of the computational domain, which is assumed to be a rectangular domain where the fractures are represented by line segments. The initial uniform structured triangular mesh is assumed to consist of right triangles formed by subdividing each square cell using its (left or right) diagonal. The limitation of rectangular domains may be lifted as long as a good initial regular mesh can be generated. The second step of the algorithm consists of a grid refinement, performed by consecutive splitting the hypotenuse of each triangles, as shown in Figure 3. Then, denoting by F the set of all fractures f in the fracture network and by $d(\mathbf{x}, f)$ the distance between the point $\mathbf{x}$ of and the fracture f, we define the function

$$g(\mathbf{x}) = d(\mathbf{x}, F) = \min_{f \in F} d(\mathbf{x}, f). \tag{1}$$

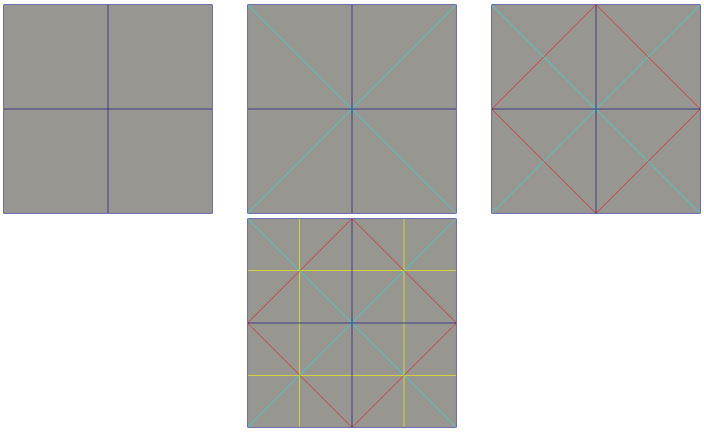

Figure 3: Example of the subsequent splitting procedure applied at the second step of the Mallison algorithm.

During the refinement process, the function g is used to determine the level of refinement of each cell. Cells that have the centroid closer to the fracture network are much more refined. The refinement process stops only when near the fractures a given level of refinement, which we still denote by h_{frac}, is achieved. During the second step of the algorithm, each fracture is approximated by using the edges of the mesh, and to better describe the fracture geometry, a moving mesh algorithm as the one proposed in [Persson and Strang, 2004] is usually employed. According to [Persson and Strang, 2004], the movement of each mesh node is performed according to the equilibrium configuration of a suitable spring system. To lower the computational costs, the moving mesh algorithm can be applied only to the nodes which belong to the triangles in the vicinity of the fractures.

2.3 Post-processing algorithm

To reduce the overall number of degrees of freedom, once a triangular grid has been generated using the above approaches, we employ a post-processing procedure similar to the one proposed by [Mallison et al., 2010]. The main idea behind the post-processing algorithm is to generate quadrilateral grid by suitably merging adjacent triangles. To select which triangles have to be merged, the following criterion is employed. A quadrilateral cell is generated if the ratio of its two diagonals d_1 and d_2 satisfies $\alpha_0 < d_1/d_2 < \alpha_1$, being α_0 and α_1 two positive parameters that can be chosen (in practice we chose $\alpha_0 = 0.8$ and $\alpha_1 = 1.25$). The above strategy ensures that the resulting quadrilateral grid satisfies a shape regularity criterion.

3 Governing equations and numerical discretization

In this section we present the mathematical model and provide its discretization through different numerical strategies, namely, the Mimetic Finite Difference method, the Extended Finite Element method and the two-points Finite Volume scheme.

Our model to describe an incompressible fluid flow problem in a N-dimensional fractured porous media $N = 2, 3$ is based on the following ingredients:

i) Governing equations for the porous medium (or bulk) flow;

ii) Governing equations for the fracture flow;

iii) Physically consistent coupling conditions along the bulk/fractures interfaces.

To ease the presentation, we will present our model in the case where there is only one fracture and it exactly cuts the domain into two disjoint sub-regions. The extension to the case of a finite number of (possibly intersecting) fractures or "partially immersed" fractures can be handled analogously. More precisely, let Ω be a sufficiently regular, open, bounded domain in $\mathbb{R}^N$ representing the porous matrix, and let $\Gamma \subset \mathbb{R}^{N-1}$ be an $N-1$ dimensional manifold representing the fracture. According to our assumption, Γ cuts Ω into two disjoint sub-domains, say Ω_1 and Ω_2.

3.1 Governing equations for the porous medium and fracture flows

We first introduce some notation and regularity assumptions and then provide the governing equations for the porous medium and the fracture flows.

We start from the porous medium model. The porous matrix is characterized by a permeability tensor field $\mathbf{K}$, which is assumed to satisfy the following (classical) regularity assumptions: $\mathbf{K}$ is a symmetric tensor whose entries $\mathbf{K}_{i,j}$, $i, j = 1, \dots, N$, are bounded, piecewise continuous real-valued functions defined on $\overline{\Omega}$, with

$$0 < \kappa_\star \leq \zeta^{\mathrm{T}} \mathbf{K}(x) \zeta \leq \kappa^\star \quad \forall \zeta \in \mathbb{R}^n \setminus \{0\} \quad \text{a.e. } x \in \overline{\Omega},$$

for constants $\kappa_\star, \kappa^\star$. Moreover, we will also assume that $\mathbf{K}$ has a block-diagonal structure in the sense explained in the following. Denoting by $\boldsymbol{n}_\Gamma$ the unit normal vector to Γ with a fixed orientation (say from Ω_1 to Ω_2), and by $\boldsymbol{\tau}_\Gamma$ the $\mathbb{R}^{N \times N-1}$ matrix whose columns forms an orthonormal basis for the tangent space at each $x \in \Gamma$, we also assume that $\mathbf{K}$ is block diagonal in local coordinates, i.e.,

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_n & 0 \\ 0 & \mathbf{K}_\tau \end{bmatrix},$$

where $\mathbf{K}_n$ and $\mathbf{K}_\tau$ are usually referred to as the normal and the tangential permeability tensors, respectively.

Assuming that $\partial\Omega = \Gamma_N \cup \Gamma_D$, $\Gamma_N \cap \Gamma_D = \emptyset$, and given a given smooth datum f, which represents a mass source or a sink, the motion of a incompressible fluid with pressure p and velocity $\boldsymbol{u}$ is governed by the Darcy's law:

$$\begin{cases} \mathbf{K}\nabla p + \boldsymbol{u} = 0 & \text{in } \Omega \\ \nabla \cdot \boldsymbol{u} = f & \text{in } \Omega \\ p = g_D & \text{on } \Gamma^D \\ \boldsymbol{u} \cdot \boldsymbol{n} = g_N & \text{on } \Gamma^N \ , \end{cases} \tag{2}$$

where the normal component of velocity and the pressure are prescribed on Γ^D and Γ^N, respectively, through the (sufficiently smooth) data g_D and g_N.

Next, we provide a model for the fracture flow. We first observe that, denoting by d_Γ the fracture's thickness, in practical applications d_Γ is very small, typically some orders of magnitude smaller than the size of the domain Ω. Therefore, the fracture can be modeled as an $(N-1)$–dimensional manifold immersed in an N-dimensional object without any loss of generality as explained in the following, cf. Figure 4 for a two-dimensional example. The main idea to

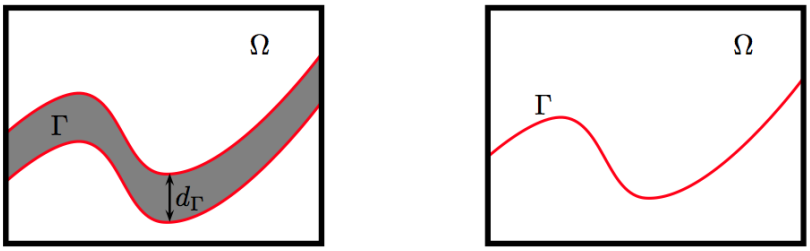

Figure 4: Physical configuration (right) and modeling of the fracture as an $N-1$ dimensional manifold immersed in an N-dimensional object (right).

obtain a reduced model for the fracture is to decompose the Darcy equations in the normal and tangential components and integrate the tangential components along the thickness d_Γ of the fracture domain. In such a way we obtain a reduced (i.e., $(N-1)$-dimensional) Darcy problem on Γ, cf. [Martin et al., 2005] for more details. The fracture flow is again characterized by the fracture permeability tensor $\mathbf{K}_\Gamma$, which is assumed to be a (positive) constant along the fracture thickness, and, writing it in its normal and tangential components, it is assumed to have a block-diagonal structure, i.e.,

$$\mathbf{K}_\Gamma = \begin{bmatrix} \mathbf{K}_n^\Gamma & 0 \\ 0 & \mathbf{K}_\tau^\Gamma \end{bmatrix}.$$

Finally, denoting by p_Γ and $\boldsymbol{q}_\Gamma$ the fracture pressure and flux, respectively, the

reduced Darcy problem on Γ reads as

$$\begin{cases} \widetilde{\mathbf{K}}_\Gamma \boldsymbol{q}_\Gamma + \nabla_\tau p_\Gamma = 0 & \text{in } \Gamma, \\ \nabla_\tau \cdot \boldsymbol{q}_\Gamma = f_\Gamma & \text{in } \Gamma, \\ p_\Gamma = g_D & \text{on } \Gamma \cap \Gamma_D, \\ \boldsymbol{q}_\Gamma \cdot \boldsymbol{n} = g_N & \text{on } \Gamma \cap \Gamma_N, \end{cases} \tag{3}$$

where $\widetilde{\mathbf{K}}_\Gamma = (d_\Gamma K_\tau^\Gamma)^{-1}$, and the source term takes into account for the fluxes entering at the interface and is defined as $f_\Gamma = d_\Gamma f + [\![\boldsymbol{u} \cdot \boldsymbol{n}_\Gamma]\!]$, being $[\![\cdot]\!]$ the standard jump operator. We finally observe that the boundary conditions impose no flux through the borders of fracture, which is an acceptable approximation.

Finally, we provide the interface conditions to couple problems (2) and (3). Following [Martin et al., 2005], for a parameter $\xi \in [0, 1]$, the coupling conditions can be written in the following general form

$$\begin{aligned} \left(\xi - \frac{1}{2}\right) \eta_\Gamma [\![\boldsymbol{u} \cdot \boldsymbol{n}_\Gamma]\!] = \{p\} - p_\Gamma \quad & \text{on } \Gamma, \\ \eta_\Gamma \{\boldsymbol{u} \cdot \boldsymbol{n}_\Gamma\} = [\![p]\!] \quad & \text{on } \Gamma, \end{aligned} \tag{4}$$

where $\eta_\Gamma = d_\Gamma (K_{\boldsymbol{n}}^\Gamma)^{-1}$ and $\{\cdot\}$ is the average operator.

3.2 Mimetic finite difference discretization

In this section we present a (low-order) MFD discretization of the couple problem (2)-(4).

Let $\mathcal{T}_h$ be a partition of Ω into non-overlapping (possibly non-convex) polygons $\mathtt{P}$, which is aligned with the fracture Γ. In practice, $\mathcal{T}_h$ can be simply built as follows: first Ω is meshed with a Cartesian grid, then the elements across Γ are simply cut in such a way that the resulting polygonal elements are aligned with, cf. Figure 5 for a representative example. Let the characteristic mesh size

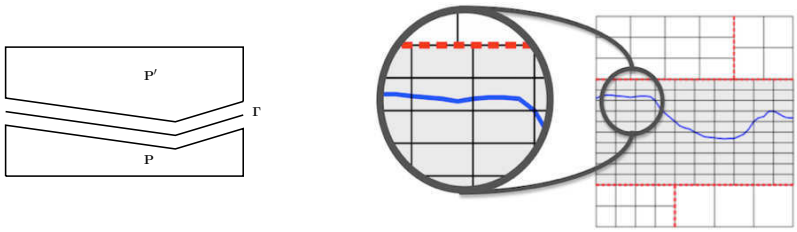

Figure 5: Example of a possible meshing strategy for Mimetic Finite Difference schemes (right) and zoom of polygonal decomposition of a rectangular element induced by the fracture (left).

$h = \max_{\mathtt{P} \in \Omega_h} h_\mathtt{P}$, being $h_\mathtt{P}$ the diameter of any element $\mathtt{P} \in \Omega_h$. The set of all internal edges of the decomposition Ω_h is denoted by $\mathcal{E}_h$. A fixed orientation is

also set for the mesh $\mathcal{T}_h$, which is reflected by a unit normal vector $\boldsymbol{n}_{\mathtt{e}}$, $\mathtt{e} \in \mathcal{E}_h$, fixed once and for all. The decomposition $\mathcal{T}_h$ is also assumed to satisfy certain (mild) shape-regularity properties, i.e., every mesh element $\mathtt{P} \in \mathcal{T}_h$

i) is simple, i.e. the boundary of $\mathtt{P}$ does not cross itself;

ii) has a uniformly bounded number of sides;

iii) $h_{\mathtt{P}}$ is uniformly comparable to the radius of its inscribed ball and to the diameters of all its sides;

cf. [Brezzi et al., 2005b]. In the spirit of MFD methods, to approximate problem (2)-(4), we denote by $\boldsymbol{\mathcal{V}}_h$ and $\mathcal{P}_h$ the discrete spaces where we will look for approximate velocities and pressures, respectively, and characterized these spaces uniquely through their degrees of freedom. More precisely, we associate the degrees of freedom for the scalar variables to mesh cells so that for $q_h \in \mathcal{P}_h$ we have $q_h = \{q_h^{\mathtt{P}}\}_{\mathtt{P}\in\mathcal{T}_h}$, being $p_{\mathtt{P}}^h \in \mathbb{R}$ the value of the discrete pressure associated to the element $\mathtt{P} \in \mathcal{T}_h$. Flux degrees of freedom are associated to mesh edges so that for $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h$, we have $\boldsymbol{v}_h = \{v_h^{\mathtt{e}}\}_{\mathtt{e}\in\mathcal{E}_h}$, with $v_h^{\mathtt{e}} \in \mathbb{R}$. With the above definitions, the dimension of $\mathcal{P}_h$ and $\boldsymbol{\mathcal{V}}_h$ is equal to the number of elements and to the number of edges of $\mathcal{T}h$, respectively.

Following [Brezzi et al., 2006], we define an inner product onto the discrete space $\mathcal{P}_h$ which corresponds to the $L^2(\Omega)$ scalar product for piecewise constant functions, i.e.,

$$[q_h, s_h]_{\mathcal{P}_h} = \sum_{\mathtt{P}\in\Omega_h} |\mathtt{P}| q_h^{\mathtt{P}} s_h^{\mathtt{P}} \quad \forall\, q_h, s_h \in \mathcal{P}_h.$$

Analogously, the inner product on $\boldsymbol{\mathcal{V}}_h$ is defined by assembling element-wise manner the contributions from each element, *i.e.*,

$$[\boldsymbol{v}_h, \boldsymbol{w}_h]_{\boldsymbol{\mathcal{V}}_h} = \sum_{\mathtt{P}\in\Omega_h} [\boldsymbol{v}_h, \boldsymbol{w}_h]_{\mathtt{P}} \quad \forall\, \boldsymbol{v}_h, \boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h,$$

where $[\cdot,\cdot]_{\mathtt{P}}$ can be defined in such a way that the following two conditions are satisfied:

(S1) *Stability*. For any $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h$ it holds

$$|\mathtt{P}| \sum_{\substack{\mathtt{e}\in\mathcal{E}_h \\ \mathtt{e}\subseteq\partial\mathtt{P}}} (v_h^{\mathtt{e}})^2 \lesssim [\boldsymbol{v}_h, \boldsymbol{v}_h]_{\mathtt{P}} \lesssim |\mathtt{P}| \sum_{\substack{\mathtt{e}\in\mathcal{E}_h \\ \mathtt{e}\subseteq\partial\mathtt{P}}} (v_h^{\mathtt{e}})^2 \quad \forall \mathtt{P} \in \mathcal{T}_h;$$

(S2) *Local consistency*. For every $\boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h$ and for every function q piecewise linear on $\mathcal{T}_h$ it holds

$$\left[(\nabla q)^{\mathrm{I}}, \boldsymbol{v}_h\right]_{\mathtt{P}} + \int_{\mathtt{P}} q\, \nabla_h \cdot \boldsymbol{v}_h \, \mathrm{dV} = \sum_{\substack{\mathtt{e}\in\mathcal{P}_h \\ \mathtt{e}\subseteq\partial\mathtt{P}}} v_h^{\mathtt{e}} \left(\boldsymbol{n}_{\mathtt{e}} \cdot \boldsymbol{n}_{\mathtt{e}}^{\mathtt{P}}\right) \int_{\mathtt{e}} q \, \mathrm{dS} \quad \forall \mathtt{P} \in \mathcal{T}_h,$$

where $\boldsymbol{n}_{\mathsf{e}}^{\mathsf{P}}$ is the unit vector normal to $\mathsf{e} \in \mathcal{E}_h$ pointing out of P, and

$$\boldsymbol{w}^{\mathrm{I}}|_{\mathsf{P}} := \frac{1}{|\mathsf{P}|}\left(\int_{\mathsf{P}} w_1 \ \mathrm{dV}, \int_{\mathsf{P}} w_2 \ \mathrm{dV},\right) \quad \forall \mathsf{P} \in \Omega_h \quad \forall\, \boldsymbol{w} := (w_1, w_2) \in [L^1(\Omega)]^2,$$
$$(\nabla_h \cdot \boldsymbol{w}_h)|_{\mathsf{P}} := \frac{1}{|\mathsf{P}|} \sum_{\substack{\mathsf{e} \in \mathcal{P}_h \\ \mathsf{e} \subseteq \partial \mathsf{P}}} |\mathsf{e}| w_h^{\mathsf{e}} \, (\boldsymbol{n}_{\mathsf{e}} \cdot \boldsymbol{n}_{\mathsf{e}}^{\mathsf{P}}) \qquad \forall \mathsf{P} \in \Omega_h \quad \forall\, \boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h;$$

cf. [Brezzi et al., 2006] for more details. Rewriting (2) in weak form and discretizing the pressure and velocity as described above, we then get the following linear system of equations for the porous medium model

$$\begin{bmatrix} \mathsf{M} & \mathsf{B}^{\mathsf{T}} \\ \mathsf{B} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h \\ p_h \end{bmatrix} = \begin{bmatrix} G \\ F \end{bmatrix}, \tag{5}$$

where M and B are the matrix representations of the inner products

$$\begin{aligned} \boldsymbol{v}_h^{\mathsf{T}} M \boldsymbol{w}_h &:= [\boldsymbol{v}_h, \boldsymbol{w}_h]_{\boldsymbol{\mathcal{V}}_h} && \boldsymbol{v}_h, \boldsymbol{w}_h \in \boldsymbol{\mathcal{V}}_h, \\ q_h^{\mathsf{T}} B \boldsymbol{v}_h &:= [q_h, \nabla_h \cdot \boldsymbol{v}_h]_{\mathcal{P}_h} && q_h \in \mathcal{P}_h, \boldsymbol{v}_h \in \boldsymbol{\mathcal{V}}_h \end{aligned}$$

respectively, and where G and F are suitable discretizations of the right hand side in (2) (and also take into account the boundary conditions). To discretize the reduced fracture model (3), we first observe that it is a one-dimensional equation, so its discretization is even simpler. Proceeding as before, we can get a linear system of equations in the discrete unknowns $\boldsymbol{q}_h^{\Gamma}$ and p_h^{Γ}. Then, we can condensate $\boldsymbol{q}_h^{\Gamma}$ and obtain a system in the variable p_h^{Γ} only. Taking also into account the coupling conditions (4), we finally get the following linear system of equations:

$$\begin{bmatrix} \widetilde{\mathsf{M}} & \mathsf{B}^{\mathsf{T}} & \mathsf{C}^{\mathsf{T}} \\ \mathsf{B} & 0 & 0 \\ \mathsf{C} & 0 & \mathsf{T} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h \\ p_h \\ p_h^{\Gamma} \end{bmatrix} = \begin{bmatrix} G \\ F \\ 0 \end{bmatrix}, \tag{6}$$

where matrices C and C^{T} allow to couple the two models, the matrix T is the one arising form MFD discretization (3) after having eliminated the $\boldsymbol{q}_h^{\Gamma}$, and $\widetilde{\mathsf{M}}$ is obtained from M after having imposed the coupling conditions.

3.3 Extended finite elements

We present here an alternative formulation to treat fractures, the eXtended Finite Element Method (XFEM). The technique is well known in the field of linear elasticity and the formulation we have used is inspired from the works by Hansbo *et al.* [Hansbo and Hansbo, 2002], [Becker et al., 2003], [Becker et al., 2009]. The technique is based on enriching with discontinuous functions the finite element space of choice in the elements cut by the fracture. The basic idea has been formulated in [Moës et al., 1999], originally for the computational analysis of the evolution of cracks in solid mechanics. Recently, this idea has been applied in combination with Nitsche's method based on penalization, see

[Becker et al., 2009], [Hansbo and Hansbo, 2002]. It has been extended to flow in fractured media by [Fumagalli and Scotti, 2013, D'Angelo and Scotti, 2012]. We have followed that approach by considering Raviart-Thomas finite elements and the coupled porous domain fracture domain problems illustrated in (2)-(4). The main characteristics of the XFEM is the ability of describing the fracture independently from the underlying computational grid. The high pressure and velocity gradients that may appear in the vicinity of the fracture are here approximated by discontinuities in the numerical solution. This allows to resolve them without resorting to mesh enrichment.

However, because of the complexity of a general implementation, particularly for 3D problems, the method is currently suited only for the study of relatively simple fracture networks. That's the reason why we here present the method on a simple configuration. Yet, the possibility of varying the fracture geometry independently from the underlying computational grid makes it rather interesting for sensitivity studies.

We consider triangular meshes and introduce discrete velocities $\mathbf{v}_h$ and pressures p_h in the following spaces,

$$\overline{\mathbf{V}}_h = \mathbf{V}_{1,h} \times \mathbf{V}_{2,h} \times \mathbf{V}_{\Gamma,h}, \quad \overline{P}_h = P_{1,h} \times P_{2,h} \times \mathbf{V}_{\Gamma,h}.$$

From now the discrete pressure is denoted by p_h and the associated discrete space by P_h. Each discrete velocity $\overline{\mathbf{v}}_h = (\mathbf{v}_h, \mathbf{v}_{\Gamma,h}) = (\mathbf{v}_{1,h}, \mathbf{v}_{2,h}, \mathbf{v}_{\Gamma,h})$ and pressure $\overline{p}_h = (p_h, p_{\Gamma,h}) = (p_{1,h}, p_{2,h}, p_{\Gamma,h})$ is thus made of three components, associated to the domains Ω_i, $i = 1, 2$ and the fracture Γ. The discrete variables in Ω are discontinuous on Γ, being defined on each part K_i of a cut triangular element $K \in \mathcal{G}_h$ by independent $(\mathbb{RT}_0, \mathbb{P}_0)$ local functions. Here $\mathbb{RT}_0$ and $\mathbb{P}_0$ indicate the standard lowest order Raviart-Thomas and piecewise constant functions, respectively. In other words, on cut elements each local function is actually a pair of independent restrictions of the traditional finite element spaces to the two sub-regions. On the fracture, we have employed again $\mathbb{RT}_0$ and P_0 elements. We wish to note that the discretization of the fracture and that of the solid matrix may be independent.

The finite element basis for the spaces $\mathbf{V}_{1,h}, \mathbf{V}_{2,h}$ and $P_{h,1}, P_{h,2}$ are obtained from the standard $\mathbb{RT}_0$ and $\mathbb{P}_0$ basis by following the approach proposed in [Hansbo and Hansbo, 2002] and replacing each standard basis function living on an element that intersects the interface by its restrictions to Ω_1 and Ω_2, respectively.

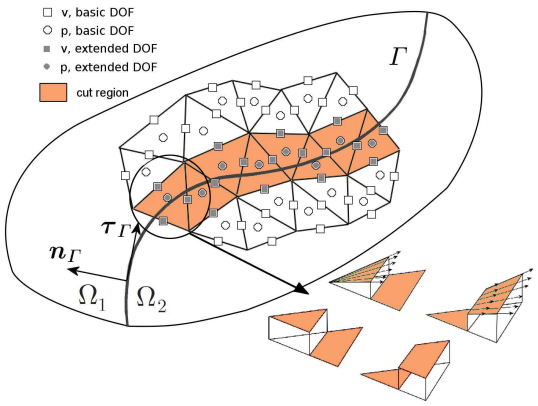

Figure 6: The porous domain Ω cut into Ω_1 and Ω_2 by the fracture Γ. Standard $\mathbb{P}_0$ degrees of freedom of the pressure associated to internal nodes are duplicated on elements K crossed by Γ (shaded), to provide constant pressure on each sub-element K_1, K_2. Analogously, the $\mathbb{RT}_0$ degrees of freedom of the velocity, associated to the edges midpoints are duplicated on such elements, leading to independent $\mathbb{RT}_0$ functions on $K_1 = K \cap \Omega_1$ and $K_2 = K \cap \Omega_2$.

Let us introduce the following bilinear and linear forms

$$a(\overline{\mathbf{u}}_h, \overline{\mathbf{v}}_h) = \int_\Omega \mathbf{K}^{-1}\mathbf{u}_h \cdot \mathbf{v}_h + \int_\Gamma \widetilde{\mathbf{K}}_\Gamma^{-1}\mathbf{p}_{\Gamma,h} \cdot \mathbf{v}_{\Gamma,h} + \eta_\Gamma \int_\Gamma \{\mathbf{u}_h \cdot \mathbf{n}_\Gamma\}\{\mathbf{v}_h \cdot \mathbf{n}_\Gamma\} + \xi \int_\Gamma \eta_\Gamma [\![\mathbf{u}_h \cdot \mathbf{n}_\Gamma]\!][\![\mathbf{v}_h \cdot \mathbf{n}_\Gamma]\!], \tag{7}$$

$$b(\overline{p}_h, \overline{\mathbf{v}}_h) = -\int_\Omega p_h(\nabla \cdot \mathbf{v}_h) - \int_\Gamma p_{\Gamma,h}(\nabla \cdot \mathbf{v}_{\Gamma,h}) + \int_\Gamma p_{\Gamma,h}[\![\mathbf{v}_h \cdot \mathbf{n}]\!], \tag{8}$$

$$\mathcal{F}(\overline{\mathbf{v}}_h) = \int_{\Gamma^D} g_D(\mathbf{v}_h \cdot \mathbf{n}) + \int_{\Gamma \cap \Gamma^D} g_D(\mathbf{v}_{\Gamma,h} \cdot \mathbf{n}) \tag{9}$$

$$\mathcal{F}(\overline{p}_h) = -\int_\Omega f p_h - \int_\Gamma f_\Gamma p_{\Gamma,h} \tag{10}$$

The standard saddle point problem: find $\overline{\mathbf{u}}_h \in \overline{\mathbf{V}}_h$ and $\overline{p}_h \in \overline{P}_h$ such that

$$\begin{aligned} a(\overline{\mathbf{u}}_h, \overline{\mathbf{v}}_h) + b(\overline{p}_h, \overline{\mathbf{v}}_h) &= \mathcal{F}(\overline{\mathbf{v}}_h) \\ b(\overline{q}_h, \overline{\mathbf{u}}_h) &= \mathcal{F}(\overline{q}_h), \end{aligned}$$

for all $\overline{\mathbf{v}}_h \in \overline{\mathbf{V}}_h$ and $\overline{q}_h \in \overline{P}_h$ eventually produces an algebraic system of the

type

$$\begin{bmatrix} \widehat{\mathsf{M}} & 0 & \widehat{\mathsf{B}}^{\mathsf{T}} & \widehat{\mathsf{B}}_{\Gamma}^{\mathsf{T}} \\ 0 & \widehat{\mathsf{M}}_{\Gamma} & 0 & \widehat{\mathsf{C}}_{\Gamma}^{\mathsf{T}} \\ \widehat{\mathsf{B}} & 0 & 0 & 0 \\ \widehat{\mathsf{B}}_{\Gamma} & \widehat{\mathsf{C}}_{\Gamma} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_h \\ \mathbf{u}_{\Gamma} \\ p_h \\ p_h^{\Gamma} \end{bmatrix} = \begin{bmatrix} \widehat{G} \\ \widehat{G}_{\Gamma} \\ \widehat{F} \\ \widehat{F}_{\Gamma} \end{bmatrix},$$

where $\mathbf{u}_h$, p_h, $\mathbf{u}_\Gamma$ and p_h^Γ are the degrees of freedom for the velocities and pressure in $\Omega_1 \cup \Omega_2$, and those in the fracture Γ, respectively. The terms in the right hand side account for the boundary conditions on Γ_D and the source terms. By algebraic manipulations it is still possible to eliminate $\mathbf{U}_\Gamma$ and produce a linear system of the form (6), but in this work we have chosen to keep the full matrix.

To account for fracture intersection in both mimetic and XFEM scheme we have implemented coupling conditions based on continuity of fluxes and transmissibility relations of the type described in [Karimi-Fard et al., 2004].

3.4 Two-point finite volume schemes

The two point finite volume scheme is a very well known technique, so we limit ourselves to recall that it is based on the primal formulation of Darcy's equations, namely

$$\begin{cases} -\nabla \cdot (\mathbf{K} \nabla p) = f & \text{in } \Omega \\ p = g_D & \text{on } \Gamma^D \\ -\mathbf{K} \dfrac{\partial p}{\partial n} = g_N & \text{on } \Gamma^N \end{cases}.$$

We have followed [Karimi-Fard et al., 2004], and, for the construction of the discrete problem, we have considered the fractures as particular control volumes, using at the intersections the conditions advocated in the cited reference. The degrees of freedom are the average pressures in each control volume in which the domain has been partitioned and in the fracture element (which, as already said, are considered as control volumes for discretization purposes). Control volumes can have, in general, a polygonal shape.

A characteristic of the two-point scheme is that for every control volume $\mathtt{P}$ the resulting algebraic system may be written in the form

$$\sum_{\mathtt{Q} \in \mathcal{E}_{\mathtt{P}}} T_{\mathtt{PQ}} (p_h^{\mathtt{Q}} - p_h^{\mathtt{P}}) = F_{\mathtt{P}},$$

where $\mathcal{E}_{\mathtt{P}}$ is the set of cells that share an edge (face) with $\mathtt{P}$, $T_{\mathtt{PQ}}$ are positive quantities called trasmissibilities, and $p_h^{\mathtt{Q}}$ and $p_h^{\mathtt{P}}$ are the discrete pressures associated to the cells $\mathtt{Q}$ and $\mathtt{P}$, respectively. Of course, an appropriate modification of the previous formula is needed for boundary cells.

4 Numerical results: performance of the two gridding schemes

We first compare Mallison and Mustapha algorithms in terms of computational complexity and quality of the generated grids. Then, we consider their performance on a single phase simulation using a two-points finite volume scheme.

In the first example we consider three realistic fracture distributions (Figure 7) on the domain $\Omega = (0, 1000)^2$: Γ_1 (201 fractures), Γ_2 (478 fractures) and Γ_3 (1028 fractures), and compare the performance of Mallison and Mustapha algorithms with that of a classical constrained Delaunay mesh generator based on the CGAL library (*http://www.cgal.org*). In this set of experiments, for Mallison and Mustapha algorithms the control parameter h_{frac} has been chosen as $h_{\text{frac}} = 10$. The comparison is carried out in terms of the number of mesh

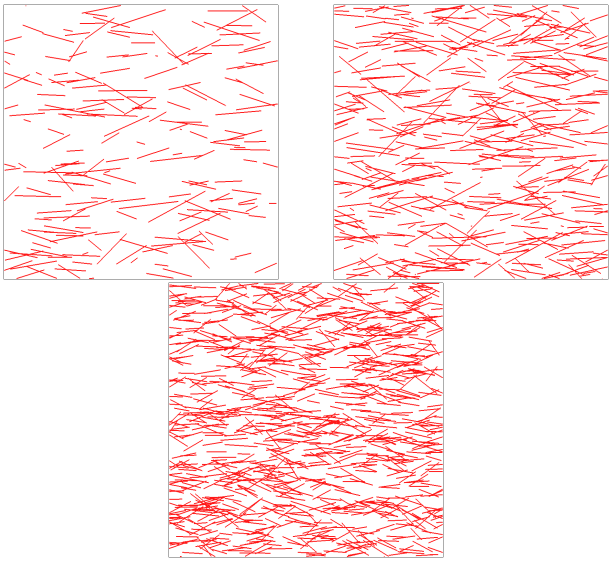

Figure 7: Different fracture distributions used to test the Mallison and Mustapha algorithms. From left to right: Γ_1 (201 fractures), Γ_2 (478 fractures) and Γ_3 (1028 fractures).

cells and two quality indexes. More precisely, for each grid $\mathcal{T}_h$, we define

$$\delta_{\max} := \max_{\mathtt{P} \in \mathcal{T}_h} \delta_{\mathtt{P}}, \qquad \delta_{\text{avr}} := \frac{\sum_{\mathtt{P} \in \mathcal{T}_h} \delta_{\mathtt{P}}}{\#\mathcal{T}_h},$$

where

$$\delta_{\mathtt{P}} = \frac{D_{\mathtt{P}}}{2d_{\mathtt{P}}}, \tag{11}$$

being $D_{\mathtt{P}}$ and $d_{\mathtt{P}}$ the diameters of the circumscribed and inscribed circle to triangle $\mathtt{P} \in \mathcal{T}_h$, respectively (note that the quality index $\delta_{\mathtt{P}} = 1$ for an equilateral

triangle).

The first comparison is carried out in terms of computational complexity of the resulting grids, i.e., number of cells of the generated meshes. Table 1 (left) shows the number of triangular cells generated by the classical constrained Delaunay, Mustapha and Mallison algorithms, for the fracture distributions Γ_1, Γ_2 and Γ_3. The corresponding number of cells resulting after the application of the post-processing merging procedure described in the previous section is also reported, cf. Table 1 (right). We can observe that both Mustapha and Mallison algorithms lead to meshes with a fewer number of elements than those obtained by a classical Delaunay mesh generator when the fracture network consists of a high number of fractures. Moreover, the post-processing merging algorithm seems to be more effective on the meshes generated by employing the Mallison algorithm than on the ones computed with the Mustapha algorithm. Indeed, the number of cells of the corresponding grids is reduced of about 40% and 25% for the meshes generated by the Mallison and Mustapha algorithms, respectively. This is not surprising, since Mallison algorithm is based on bisecting a regular mesh and it is then easier to partially recover good quadrilateral elements by post-processing.

Before post-processing

	Delaunay	Mustapha	Mallison
Γ_1	6238	7743	11077
Γ_2	25616	13594	12651
Γ_3	66517	17861	12781

After post-processing

	Delaunay	Mustapha	Mallison
Γ_1	4758	5683	6362
Γ_2	19446	10115	6951
Γ_3	50277	13653	7330

Table 1: Cell counts of the meshes generated by the classical constrained Delaunay, Mustapha, and Mallison algorithms before (left) and after (right) the post-processing merging algorithm on the fracture distributions Γ_1, Γ_2 and Γ_3.

We now compare the different strategies in terms of the overall quality of the resulting grids. In Table 2 we report the computed quality indexes δ_{avr} and $\delta_{\max}$ for the grids obtained on the different fracture distributions Γ_1, Γ_2 and Γ_3. From these results, we observe that *i)* Mustapha and Mallison algorithms outperform classical constrained Delaunay algorithm, in terms of average quality index; *ii)* Mustapha algorithm seems to better control cells degeneration leading to a smaller and uniform value of $\delta_{\max}$, whereas Mallison algorithm may lead to the creation of highly distorted cells.

Next, we consider the fracture distribution Γ_1 and address the robustness of the Mustapha and Mallison algorithms when varying the control parameter

δ_{avr}

	Delaunay	Mustapha	Mallison
Γ_1	1.8147	1.3324	1.2261
Γ_2	1.4116	1.3249	1.2496
Γ_3	1.3936	1.3141	1.3042

δ_{max}

	Delaunay	Mustapha	Mallison
Γ_1	1828.8900	3.8677	2.9878
Γ_2	180.7000	3.8737	15.8141
Γ_3	370.1960	3.9978	167.2120

Table 2: Computed quality indexes of the meshes generated by the constrained Delaunay, Mustapha and Mallison algorithms on the different fracture distributions Γ_1, Γ_2 and Γ_3.

h_{frac}, which represents the mesh granularity imposed on the fracture network in the Mustapha algorithm and the maximum refinement level allowed near the fractures in the Mallison scheme. In Table 3 the total number of mesh elements obtained based on employing Mustapha and Mallison algorithms for different choices of $h_{\text{frac}} = 40, 30, 20, 10, 5$ are reported. The corresponding computed quality indexes are listed in Table 4. From the results reported in Table 3, it is clear that the Mallison scheme tends to create more elements than Mustapha algorithm when h_{frac} is small. This is probably due to the fact that in order to maintain mesh conformity Mallison algorithm is forced to create transition regions between the more refined parts of the computational domain (adjacent to the fractures) and less refined ones. We also observe that for both algorithms, the average quality index δ_{avr} is almost constant as h_{frac} decreases, suggesting that both algorithms seems to be robust with respect to h_{frac}. Regarding the capability to avoid the creation of highly distorted cells, Mustapha algorithm seems to be slightly more robust than Mallison algorithm, as it gives consistent results for a wide range of h_{frac} .

h_{frac}	Mustapha	Mallison
40	1480	459
30	1925	990
20	2626	1696
10	5683	6362
5	12967	19772

Table 3: Cell counts of the meshes generated by the Mustapha and Mallison algorithms for different values of h_{frac} (fracture distribution Γ_1).

Next, we aim at investigating how the quality of the computational grids influences the accuracy of the numerical solutions resulting after discretization of our model problem. More precisely, we study the effects induced on

	h_{frac}	40	30	20	10	5
Mustapha	δ_{avr}	1.3276	1.3288	1.3247	1.3324	1.3344
	δ_{max}	3.8214	3.5217	3.4877	3.8677	4.1238
Mallison	δ_{avr}	1.2778	1.2996	1.2442	1.2261	1.2201
	δ_{max}	4.3024	31.0602	7.4317	2.9878	3.4915

Table 4: Computed quality indexes of the meshes generated by the Mustapha and Mallison algorithms for different values of h_{frac} (fracture distribution Γ_1).

the resulting accuracy of the numerical solution by the fracture network approximation. We focus on the fracture distribution Γ_1, and couple the two-meshing algorithms with a two-points Finite Volume scheme for the discretization of problem (2)-(3)-(4), following the approach proposed by Karimi-Fard in [Karimi-Fard et al., 2004] for fracture network discretization. Throughout the rest of this section, we choose $\Omega = (0, 1000)^2$ and set the source term $f = 0$, the permeability tensor field $\mathbf{K} = \mathbf{I}$, being $\mathbf{I}$ the identity tensor, and the fracture permeability $\widetilde{\mathbf{K}}_\Gamma = 10^5\mathbf{I}$. Aperture distributions in the fractures has an order of magnitude of 10^{-5}, and the fluid viscosity is set equal to 1. Finally, the boundary conditions imposed in this test case are shown in Figure 8 together with a computed reference solution obtained on a extremely fine constrained Delaunay triangulation (1252903 mesh elements, 1269208 pressure dofs, including fracture elements).

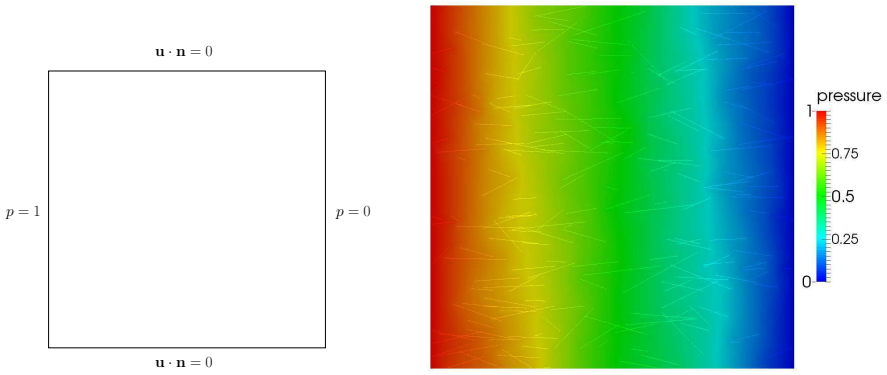

Figure 8: Boundary conditions imposed (left) and reference solution (right) computed on a grid of approximately 1250000 elements.

To compare the numerical solutions obtained on the different meshes, we compute the outer flux across the right boundary of the domain, i.e.,

$$F_{\text{out}}(\mathbf{u}_h) = \int_{\Gamma_{\text{out}}} \mathbf{u}_h \cdot \mathbf{n}. \tag{12}$$

In Table 5 we report the corresponding total number of pressure degrees of freedom and the computed relative error between the approximated outer flux

$F_{\text{out}}(\mathbf{u}_h)$ and the corresponding quantity $F_{\text{out}}(\mathbf{u}_h^{\text{ref}})$ computed with the reference solution $\mathbf{u}_h^{\text{ref}}$, i.e.,

$$e_h = \frac{|F_{\text{out}}(\mathbf{u}_h) - F_{\text{out}}(\mathbf{u}_h^{\text{ref}})|}{|F_{\text{out}}(\mathbf{u}_h^{\text{ref}})|}, \tag{13}$$

for different choices of h_{frac}.

	pressure dof		e_h	
h_{frac}	Mustapha	Mallison	Mustapha	Mallison
40	2051	703	0.0936	0.0739
30	2689	1437	0.0995	0.0539
20	3711	2355	0.0644	0.0299
10	7882	7879	0.0551	0.0081
5	17199	23052	0.0267	0.0031

Table 5: Total number of pressure degrees of freedom (dof) and relative error e_h as a function of the fracture discretization step h_{frac}.

In Figure 9, we report the discrete pressure fields computed on the grids generated by the Mallison (left) and Mustapha (right) algorithms for $h_{\text{frac}} = 40, 20, 5$.

The relative error in the outer flux approximation decreases when h_{frac} is refined, as expected. In terms of outer flux approximation, simulations performed on meshes generated by the Mallison algorithm seem to give better results, as shown in Figure 10, even when comparing meshes with similar number of degrees of freedom. The reason is probably due to the fact that Mallison scheme by construction provide a more uniformly fine grid in the vicinity of the fracture, so mass transfer between matrix and fracture is approximated better.

4.1 Transient Simulations

The second numerical experiment used to assess the performance of the two algorithms is the solution of the following time dependent problem:

$$\phi c_T \frac{\partial p}{\partial t} = \nabla \cdot \left(\frac{1}{\mu} K \nabla p\right) + q \tag{14}$$

where $\phi = 1$ is the matrix porosity, $c_T = 10^{-5}$ is the total compressibility, $\mu = 1$ is the fluid viscosity, K is the medium permeability ($K_m = 1$ for the matrix and $K_f \simeq 10^5$ for the fractures) and $q(t) = 0$ is the source term, per unit volume.

We perform two different tests. In the first test, equation (14) is solved by applying Neumann zero flux boundary condition on the whole boundary of the domain $\Omega = (0, 1000)^2$ and by setting to 1 the value of the pressure in the fracture. The initial value of the pressure is set equal to 0 on the whole domain. With the above settings, equation (14) models a situation where all the fracture network injects fluid into the matrix. It is clear that the total fluid flow from

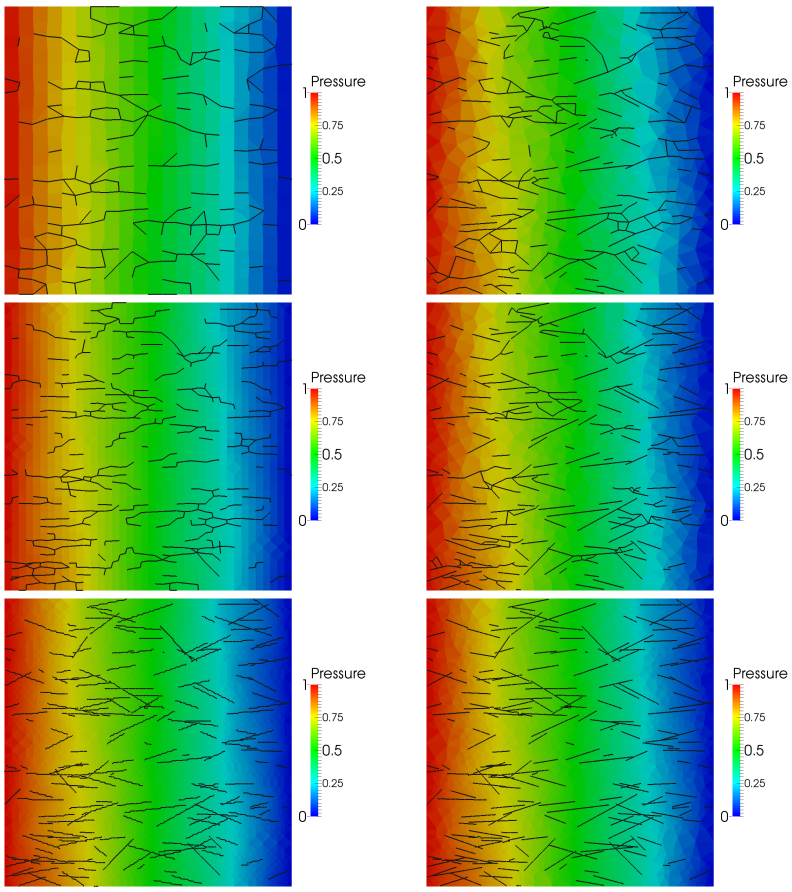

Figure 9: Discrete pressure fields computed on the grids generated by the Mallison (left) and Mustapha (right) algorithms for $h_{\text{frac}} = 40, 20, 5$.

the fracture network to the matrix converges to zero, whereas the final pressure in the whole domain converges to 1.

To compare the performance of Mustapha and Mallison algorithms, we monitor the discrete values of the flux $\tilde{F}_{in}$ injected from the fracture network into the matrix. In particular, we compare the values of $\tilde{F}_{in}$ obtained by the two algorithms at the initial time step (t=1) with the reference value $\tilde{F}_{in}^{ref} = 1.0860197$ obtained by employing the same fine grid of the steady state test case. Figure 11 shows the trend of the relative error e_r, computed similarly to (13), by varying h_{frac} and the number of degrees of freedom of the pressure.

From Figure 11 (left) we observe that for large values of h_{frac} the Mustapha algorithm better approximates the inflow flux reference value, while for smaller values of h_{frac} the Mallison algorithm exhibits a slightly better behavior. However, comparing the two algorithms in terms of the number of pressure dofs

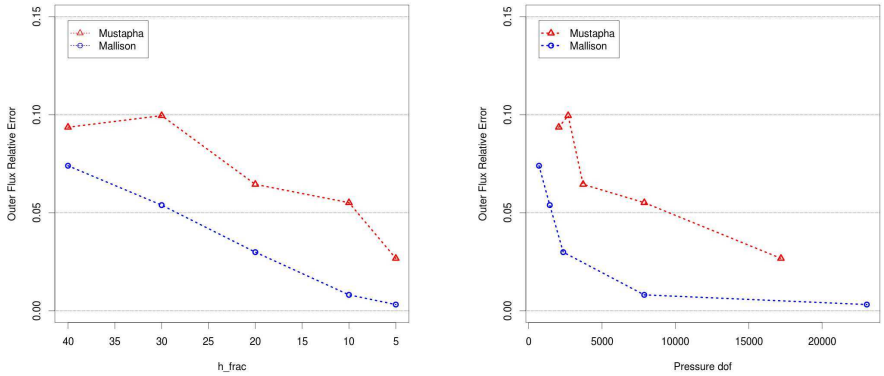

Figure 10: Relative error of the outer flux against fracture discretization step h_{frac} (left) and pressure degrees of freedom (right).

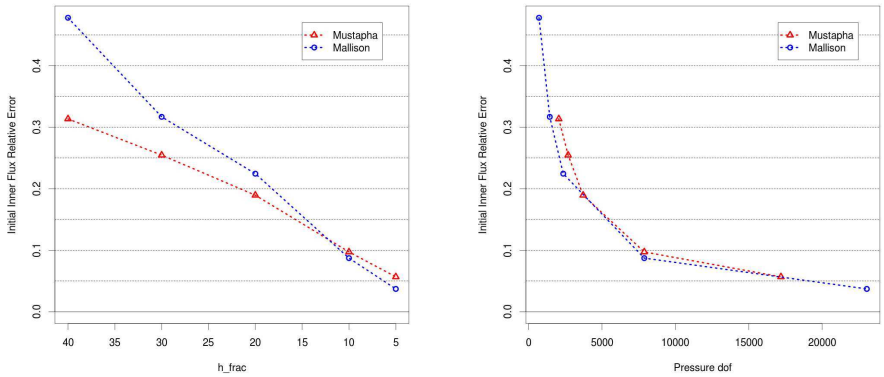

Figure 11: Transient simulation: relative error of the inflow flux at $t = 1$ against fracture discretization step h_{frac} (left) and pressure degrees of freedom (right).

(Figure 11, on the right) indicates that the two algorithms perform similarly.

A more important difference in the performance of the two algorithms is found if we compare the accuracy of the discrete inflow flux in terms of the total CPU Time (in seconds) needed to build the meshes. The computations have been carried out on a workstation with the following characteristics: I7 Intel processor, 3.50GHz, 8MByte cache, 32Gbytes of RAM.

On one hand, as shown in Figure 12, Mustapha algorithm seems to be able to return more accurate results in shorter time. The larger CPU time needed by the Mallison algorithm seems to be linked to the implementation of the moving mesh procedure (solution of a suitable spring system).

Mustapha

h_{frac}	Pressure dof	CPU Time	$\tilde{F}_{in}(t=1)$	e_r
40	2051	$\simeq 0$	0.7455	0.3135
30	2689	$\simeq 0$	0.8094	0.2546
20	3711	$\simeq 0$	0.8802	0.1894
10	7882	2	0.9803	0.0972
5	17199	10	1.0243	0.0567
3	31650	42	1.0385	0.0437

Mallison

h_{frac}	Pressure dof	CPU Time	$\tilde{F}_{in}(t=1)$	e_r
40	703	$\simeq 0$	0.5670	0.4779
30	1437	$\simeq 0$	0.7419	0.3168
20	2355	1	0.8422	0.2244
10	7879	7	0.9912	0.0872
5	23052	50	1.0456	0.0371

Table 6: Transient simulation: pressure degrees of freedom, total CPU Time needed to build the mesh, initial inflow flux and the flux relative error.

On the other hand, if we build a more refined grid using the Mustapha method (31650 pressure dof), such that the CPU time employed to build the mesh is similar to the one employed by the Mallison algorithm to build its mesh (see Table 6), the problem solution does not achieve the same accuracy.

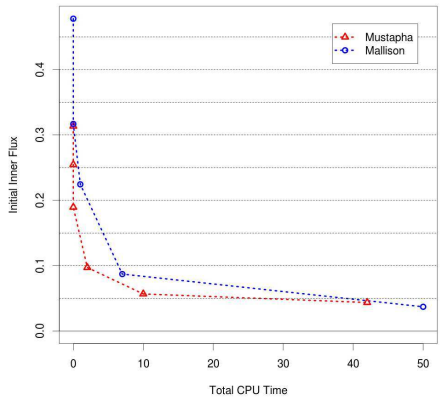

Figure 12: Relative error on the initial inflow flux vs the total CPU time (in seconds) needed to build the mesh.

The trend of the cumulative flux in the first time steps (Figure 13) highlights the difference between the methods. Indeed, Mustapha algorithm seems to be

more accurate on less refined grid. However, for smaller value of h_{frac} the flux computed using meshes obtained with the Mallison algorithm get closer to the reference value.

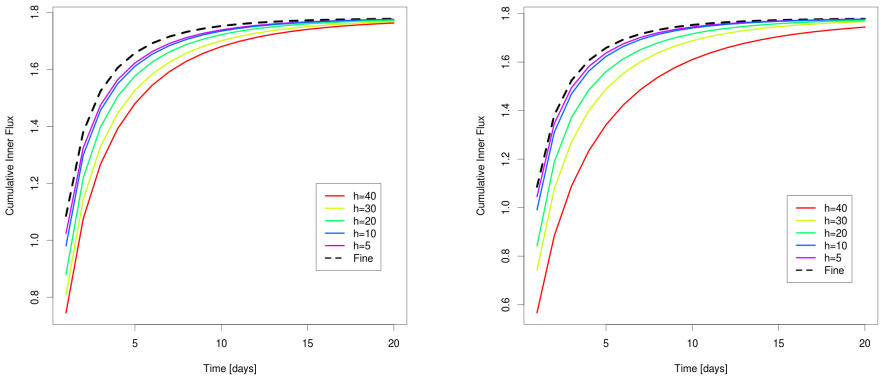

Figure 13: Transient simulation: trend of $\tilde{F}_{in}(t)$ in the first time steps for the solutions computed on the Mustapha (left) and Mallison (right) meshes by varying h_{frac}.

In the second test, equation (14) is solved by applying Neumann zero flux boundary condition on the whole boundary of the domain $\Omega = (0, 1000)^2$ and by setting to 1 and to 0 the value of the pressure in a fracture at the left-down corner of the domain ($ID = 1$ in figure 14) and in a fracture at the top-right corner of the domain ($ID = 193$ in figure 14), respectively. The initial value of the pressure is set equal to 0 on the rest of the domain. With this setting, equation (14) models a situation where an injection and a production wells are placed in the quoted fractures.

To compare the performance of the two algorithms, we monitor the discrete values of the flux $\tilde{F}_{in}$ injected from the fracture 1 into the matrix. In particular, we compare the values of $\tilde{F}_{in}$ obtained by the two algorithms at the initial time step (t=1) with the reference value $\tilde{F}_{in}^{ref} = 0.009880169$ obtained by employing the same fine grid used for the steady state test.

Figure 15 shows the trend of the relative error e_r, computed similarly to (13), by varying h_{frac} and the number of degrees of freedom of the pressure. Both methods converges, with Mallison algorithm seems to behave slightly better on the more refined grids.

In figure 16 we show the comparison in terms of CPU time needed to build the grid.

We now compare the error at the final time, which has been taken as the one when

$$\|p_h^{n+1} - p_h^n\|_{L^2(\Omega)} < \epsilon_{toll}. \tag{15}$$

where $\epsilon_{toll} = 1e-5$.

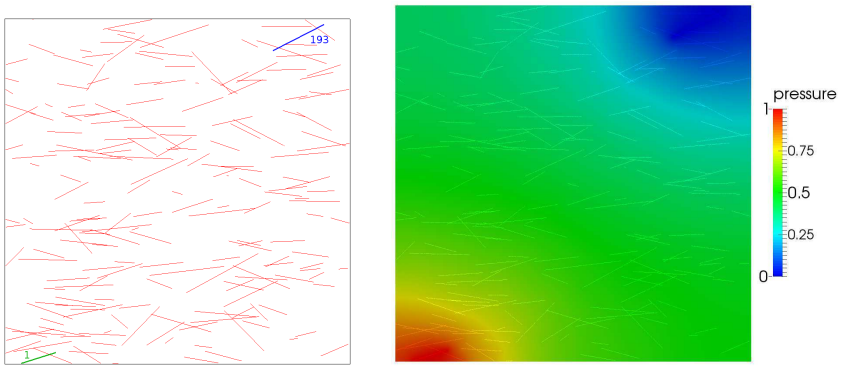

Figure 14: Pseudo-Steady State simulations: displacement of fracture 1 and 193 in the domain (left), and reference solution (right) computed on a grid of approximately 1250000 elements.

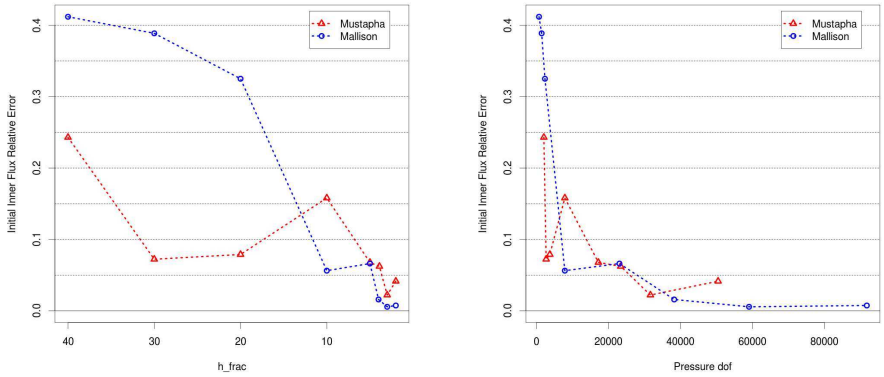

Figure 15: Transient simulation: relative error of the initial inflow flux from fracture 1 against fracture discretization step h_{frac} (left) and pressure degrees of freedom (right).

We note how for the same degrees of freedom Mallison algorithm provides a better approximation of the reference value, cf. Figure 17, particularly for finer grids.

Mustapha

h_{frac}	Pressure dof	CPU Time	$\tilde{F}_{in}(t=1)$	$e_r(t=1)$
40	2051	$\simeq 0$	0.0074	0.2430
30	2689	$\simeq 0$	0.0091	0.0722
20	3711	$\simeq 0$	0.0091	0.0789
10	7882	2	0.0083	0.1580
5	17199	10	0.0092	0.0677
$\simeq 4$	23398	21	0.0092	0.0623
3	31650	42	0.0096	0.0220
2	50459	141	0.0094	0.0414
1	109590	942	0.0095	0.0330

Mallison

h_{frac}	Pressure dof	CPU Time	$\tilde{F}_{in}(t=1)$	e_r
40	703	$\simeq 0$	0.0058	0.4120
30	1437	$\simeq 0$	0.0060	0.3888
20	2355	1	0.0066	0.3251
10	7879	7	0.0093	0.0562
5	23052	50	0.0092	0.0661
4	38251	154	0.0097	0.0158
3	59057	411	0.0098	0.0055
2	91769	1143	0.0098	0.0073

Table 7: Transient simulation: pressure degrees of freedom, total CPU Time needed to build the mesh, inflow flux and flux relative error at $t = 1$.

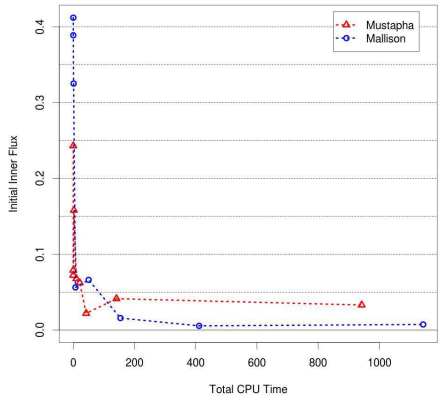

Figure 16: Relative error on the inflow flux vs the total CPU time (in seconds) needed to build the mesh.

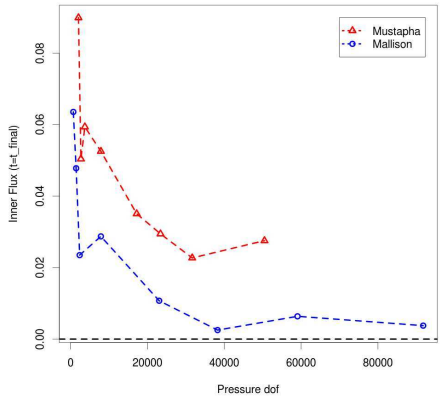

Figure 17: Relative error on the inflow flux at equilibrium against the pressure degrees of freedom.

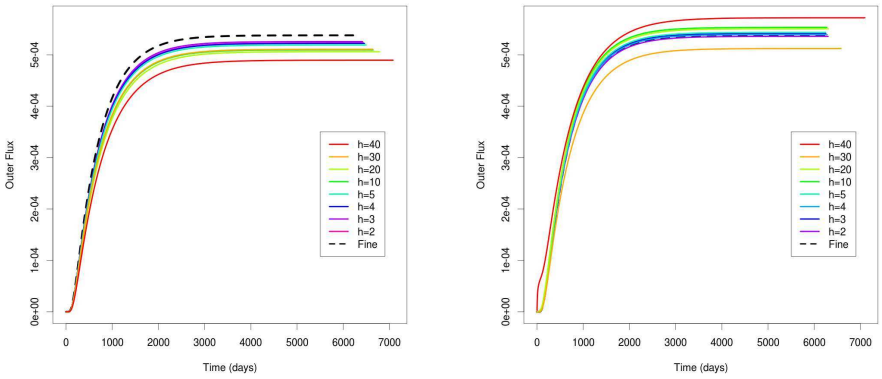

Figure 18: Transient simulation: trend of the outflow flux $\tilde{F}_{out}(t)$ for the solutions computed on the Mustapha (left) and Mallison (right) meshes by varying h_{frac}.

5 Numerical results: comparison of FV, MFD and XFEM discretization schemes

In this section we compare the performance of three different discretization techniques to simulate the fluid flow in fractured porous media. In particular, we will analyze algorithms based on two-points finite volumes (FV) coupled with gridding algorithms, mimetic finite differences (MFD) and extended finite elements (XFEM), respectively.

We consider the domain $\Omega = (0,1)^2$. The fracture network, shown in figure 19, is composed by 11 fractures with aperture $a = 10^{-5}$ and permeability $\mathbf{K}_\Gamma = 10^5\,\mathbf{I}$, being $\mathbf{I}$ the identity matrix. Finally, we set the permeability tensor $\mathbf{K} = \mathbf{I}$ in the surrounding porous medium. We solve the problem (2)-(4) with pressure p equal to 0 on $\partial\Omega$ and with external source/sink function $f(x)$ defined as

$$f(\mathbf{x}) = \begin{cases} 10 & \text{if } \mathbf{x} \in B_{0.2}(0.1, 0.1) \\ -10 & \text{if } \mathbf{x} \in B_{0.2}(0.9, 0.9) \end{cases} \tag{16}$$

where $B_r(x_0, y_0)$ denotes the ball with radius r and center in (x_0, y_0).

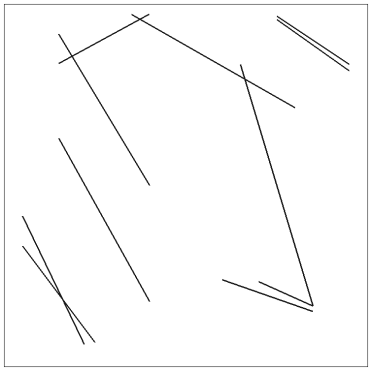

Figure 19: The fracture network geometry employed in the test case to compare FV, MFD and XFEM techniques.

For the sake of comparison, we employ the reference solution to (2)-(4) obtained by employing the two-points finite volume scheme on a very fine grid (1254831 cells) (see Figure 20).

In Figure 21 we show the isolines of the pressure solutions obtained employing two-points finite volumes coupled with (Mustapha and Mallison) gridding algorithm (top), the MFD applied on a Delaunay triangulation constrained on all the fracture network (bottom-left) and XFEM applied on a structured triangular mesh non conforming to the fractures (bottom-right), respectively. A close inspection of the numerical results reported in Figure 21 reveal a good qualitative agreement with the reference solution depicted in Figure 20. Specific gridding techniques thus allow us to obtain good results with a low computational

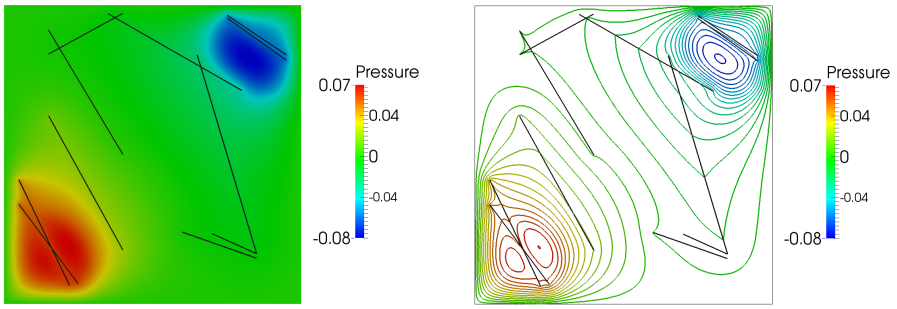

Figure 20: Reference solution computed employing two-points FV discretization on a very fine grid. Pressure field (left) and pressure isolines (right).

cost using a very simple numerical scheme such as the two point flux approximation. However, it is well known that this scheme provides an inaccurate estimate of the flux if the grids are generic. Therefore, it is worth investigating innovative techniques that, even if not yet competitive on demanding realistic cases, have good convergence properties and were able to give promising results on this simple test case.

6 Conclusions

In this paper we wanted to carry out two objectives. The first was to compare two different gridding strategies recently proposed for the generation of meshes conforming to a network of fractures. To that purpose, we have carried out simulations of single phase flow using a rather standard finite volume discretization, since it is most used by practitioners of the field. The second objective was to investigate the suitability of two alternative discretization schemes, namely MFD and XFEM, to describe flow in fractured media.

For what concerns the first objective, we have shown that both the selected gridding algorithms allow to obtain meshes of good quality with respect to the direct application of a classic constrained Delaunay approach, with a slightly better performance of the Mustapha procedure than Mallison's. The most evident advantage of the former is a better control of the distortion of the worst elements. As for the performance with respect to simulation results, we have compared the solution obtained using grids produced by the two algorithms with a reference solution on an extremely fine grid. And for three benchmark problems: a steady state solution with imposed pressure drop and two transient computation with imposed pressure value in the fractures.

The results do not allow to give a definite opinion on the two algorithms. Both perform satisfactorily, but a better grading of the mesh in the vicinity of the fractures allows the Mallison algorithm to give, in general, more accurate results for the same number of degrees of freedom, particularly when using more

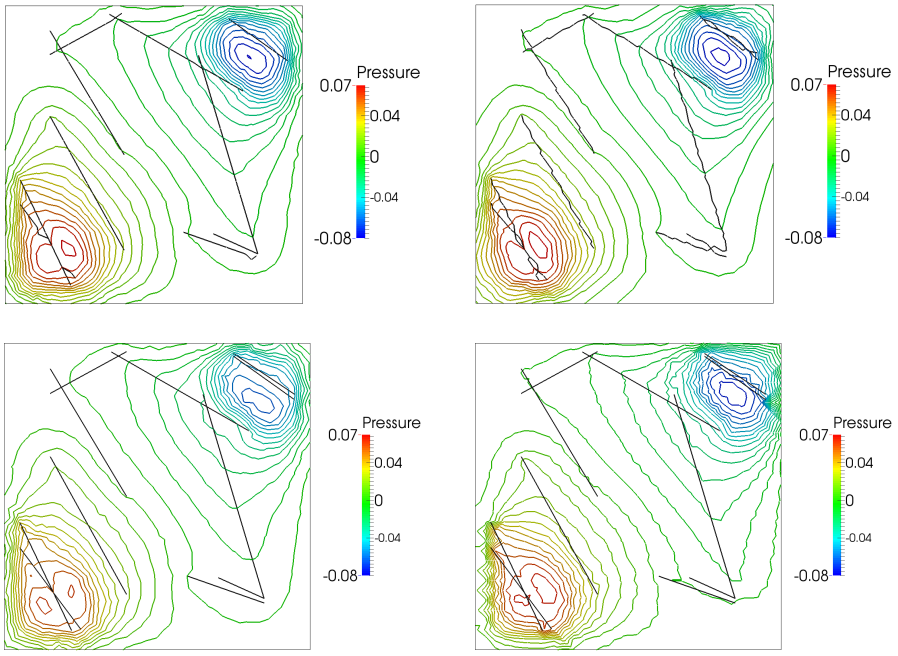

Figure 21: Comparison of the discretization methods: isolines of the pressure solution. FV coupled with Mustapha gridding algorithm (top-left), FV coupled with Mallison gridding algorithm (top-right), MFD applied on a Delaunay triangulation constrained on the fracture network (bottom-left) and XFEM applied on a Cartesian mesh non conforming to the fractures (bottom-right).

refined grids and in the steady state problem. For the first transient problem, the difference is small. Probably because, having imposed the pressure on the whole network, we are not in fact solving the problem along the fractures. Then, the effect of the stronger refinement near the fractures induced by Mallison algorithm is less important in this case. Indeed, in the other test, where we have instead imposed a pressure drop between two disconnected parts of the fracture network, we have obtained results more similar to those of the steady state problem in terms of solution accuracy. This points out the importance of well resolving the fracture network, rather than an intrinsic limitation of Mustapha algorithm.

As for the computational cost, the mesh movement algorithm in the Mallison scheme is rather demanding. Of course, this can be a matter of implementation, still we can conclude that this part can be a computational bottleneck and has to be addressed with care.

Coming to the comparison between XFEM and MFD, we have considered the solution obtained by the two methods on a relatively simple fracture network, against that obtained using the FV scheme on an extremely fine grid.

The results show that both techniques are able to capture the solution satisfactorily, even on a rather coarse mesh. MFD seems very promising, since it has optimal convergence property with respect to both pressure and velocity field, and can handle arbitrary polygonal grids. The XFEM technique has still to be developed further to be able to treat complex networks. Its capability of having the discretization of the fractures independent from that of the matrix, however, makes it interesting for sensitivity study with respect to fracture position and orientation. For both the MFD and XFEM methods we have used a mixed formulation of the equations, which is the one that guarantees a good representation of the fluxes. Yet, for both techniques it is possible to start from the primal formulation, or to apply algebraic manipulations to obtain a formulation with only pressure unknowns, but with a larger stencil than two-points FV.

In this work we have considered only 2D cases, for the sake of simplicity. The gridding algorithms as well as MFD and XFEM can clearly be extended in 3D. However, it is our feeling that Mallison algorithm allows a more straightforward extension, being based on recursive subdivision of an underlying grid.

7 Acknowledgments

We wish to thank Nicola Verzotti for his contribution to the coding of the MFD scheme.

References

[Adler et al., 2012] Adler, P., Thovert, J.-F., and Mourzenko, V. (2012). *Fractured porous media*. Oxford University Press.

[Alpak, 2010] Alpak, F. (2010). A mimetic finite volume discretization method for reservoir simulation. *SPE Journal*, 15(2):436–453.

[Arbogast et al., 1990] Arbogast, T., Douglas, Jr., J., and Hornung, U. (1990). Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM J. Math. Anal.*, 21(4):823–836.

[Barenblatt and Zheltov, 1960] Barenblatt, G. and Zheltov, Y. (1960). Fundamental equations of filtration of homogeneous liquids in fissured rocks. *Dokl. Akad. Nauk SSSR*, 13:545–548.

[Becker et al., 2009] Becker, R., Burman, E., and Hansbo, P. (2009). A Nitsche extended finite element method for incompressible elasticity with discontinuous modulus of elasticity. *Comput. Methods Appl. Mech. Engrg.*, 198(41-44):3352–3360.

[Becker et al., 2003] Becker, R., Hansbo, P., and Stenberg, R. (2003). A finite element method for domain decomposition with non-matching grids. *M2AN Math. Model. Numer. Anal.*, 37(2):209–225.

[Brezzi et al., 2005a] Brezzi, F., Lipnikov, K., and Shashkov, M. (2005a). Convergence of the mimetic finite difference method for diffusion problems on polyhedral meshes. *SIAM J. Numer. Anal.*, 43(5):1872–1896 (electronic).

[Brezzi et al., 2006] Brezzi, F., Lipnikov, K., and Shashkov, M. (2006). Convergence of mimetic finite difference method for diffusion problems on polyhedral meshes with curved faces. *Math. Models Methods Appl. Sci.*, 16(2):275–297.

[Brezzi et al., 2005b] Brezzi, F., Lipnikov, K., and Simoncini, V. (2005b). A family of mimetic finite difference methods on polygonal and polyhedral meshes. *Math. Models Methods Appl. Sci.*, 15(10):1533–1551.

[Brooks and Corey, 1964] Brooks, R. H. and Corey, A. T. (1964). *Hydraulic properties of porous media*. Civil Engineering Dept., Colorado State Univ., Fort Collins, CO.

[Cheng et al., 2012] Cheng, S.-W., Dey, T., and Shewchuk, J. (2012). *Delaunay mesh generation*. CRC Press.

[Choi et al., 1997] Choi, E., Cheema, T., and Islam, M. (1997). A new dual-porosity/dual-permeability model with non-Darcian flow through fractures. *Journal of Petroleum Science and Engineering*, 17(3):331–344.

[D'Angelo and Scotti, 2012] D'Angelo, C. and Scotti, A. (2012). A mixed finite element method for Darcy flow in fractured porous media with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(02):465–489.

[Durlofsky, 2003] Durlofsky, L. (2003). Upscaling of geocellular models for reservoir flow simulation: A review of recent progress. In *7th International Forum on Reservoir Simulation Bühl/Baden-Baden, Germany*, pages 23–27.

[Fumagalli and Scotti, 2013] Fumagalli, A. and Scotti, A. (2013). A numerical method for two-phase flow in fractured porous media with non-matching grids. *Advances in Water Resources*, 62:454–464.

[Gong and Durlofsky, 2008] Gong, B. and Durlofsky, L. (2008). Upscaling discrete fracture characterizations to Dual-Porosity, Dual-Permeability models for efficient simulation of flow with strong gravitational effects. *SPE Journal*, 13(1):58–67.

[Guevara-Jordan and Arteaga-Arispe, 2007] Guevara-Jordan, J. M. and Arteaga-Arispe, J. (2007). A second-order mimetic approach for tracer flow in oil reservoirs. *Latin American & Caribbean Petroleum Engineering Conference*.

[Hansbo and Hansbo, 2002] Hansbo, A. and Hansbo, P. (2002). An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Comput. Methods Appl. Mech. Engrg.*, 191(47-48):5537–5552.

[Karimi-Fard et al., 2004] Karimi-Fard, M., Durlofsky, L., and Aziz, K. (2004). An efficient discrete-fracture model applicable for general-purpose reservoir simulators. *SPE Journal*, 9(2):227–236.

[Karimi-Fard et al., 2006] Karimi-Fard, M., Gong, B., and Durlofsky, L. (2006). Generation of coarse-scale continuum flow models from detailed fracture characterizations. *Water Resources Research*, 42(10):1–13.

[Kuznetsov et al., 2004] Kuznetsov, Y., Lipnikov, K., and Shashkov, M. (2004). The mimetic finite difference method on polygonal meshes for diffusion-type problems. *Comput. Geosci.*, 8(4):301–324 (2005).

[Lim et al., 2009] Lim, K., Hui, M., and Mallison, B. (2009). A next-generation reservoir simulator as an enabling technology for a complex discrete fracture modeling workflow. *SPE Paper 124980*.

[Lipnikov et al., 2008] Lipnikov, K., Moulton, J. D., and Svyatskiy, D. (2008). A multilevel multiscale mimetic (M^3) method for two-phase flows in porous media. *J. Comput. Phys.*, 227(14):6727–6753.

[Mallison et al., 2010] Mallison, B., Hui, M., and Narr, W. (2010). Practical Gridding Algorithms for Discrete Fracture Modeling Workflows. In *ECMOR XII, 12th European Conference on the Mathematics of Oil Recovery*, number September 2010, pages 1–11.

[Martin et al., 2005] Martin, V., Jaffré, J., and Roberts, J. E. (2005). Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691 (electronic).

[Moës et al., 1999] Moës, N., Dolbow, J., and Belytschko, T. (1999). A finite element method for crack growth without remeshing. *Int. J. for Numerical Methods in Eng.*, 46(1):131–150.

[Mustapha, 2012] Mustapha, H. (2012). A Gabriel-Delaunay triangulation of complex fractured media for multiphase flow simulations. In *ECMOR XIII-13th European Conference on the Mathematics of Oil Recovery*, pages 1–18.

[Mustapha, 2014] Mustapha, H. (2014). An efficient hybrid local nonmatching method for multiphase flow simulations in heterogeneous fractured media. *Engineering with Computers*, pages 1–14. (on line).

[Mustapha and Dimitrakopoulos, 2011] Mustapha, H. and Dimitrakopoulos, R. (2011). Discretizing two-dimensional complex fractured fields for incompressible two-phase flow. *International Journal for Numerical Methods in Fluids*, 65(7):764–780.

[Mustapha et al., 2011] Mustapha, H., Dimitrakopoulos, R., Graf, T., and Firoozabadi, A. (2011). An efficient method for discretizing 3D fractured media for subsurface flow and transport simulations. *International Journal for Numerical Methods in Fluids*, 67(5):651–670.

[Nielsen et al., 2012] Nielsen, H. M., Natvig, J. R., and Lie, K.-T. (2012). Accurate modeling of faults by multipoint, mimetic and mixed methods. *SPE Journal*, 17(2):568–579.

[Persson and Strang, 2004] Persson, P.-O. and Strang, G. (2004). A simple mesh generator in Matlab. *SIAM Rev.*, 46(2):329–345 (electronic).

[Saas et al., 2005] Saas, L., Faille, I., Nataf, F., and Willien, F. (2005). Finite volume methods for domain decomposition on nonmatching grids with arbitrary interface conditions. *SIAM journal on numerical analysis*, 43(2):860–890.

[Sahimi, 2012] Sahimi, M. (2012). *Flow and Transport in Porous Media and Fractured Rock: From Classical Methods to Modern Approaches*. J. Wiley and Sons.

[Warren and Root, 1963] Warren, J. and Root, P. (1963). The behaviour of naturally fractured reservoirs. *SPE J.*, 3(3):245–255.

MOX Technical Reports, last issues

Dipartimento di Matematica "F. Brioschi",
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

27/2014 Vergara, C; Domanin, M; Guerciotti, B; Lancellotti, R.M.; Azzimonti, L; Forzenigo, L; Pozzoli, M
Computational comparison of fluid-dynamics in carotids before and after endarterectomy

28/2014 Antonietti, P; Panfili, P.; Scotti, A.; Turconi, L. ; Verani, M.; Cominelli,A.; Formaggia,L.
Optimal techniques to simulate flow in fractured reservoirs

26/2014 Discacciati, M.; Gervasio, P.; Quarteroni, A.
Interface Control Domain Decomposition (ICDD) Methods for Heterogeneous Problems

25/2014 Hron, K.; Menafoglio, A.; Templ, M.; Hruzova K.; Filzmoser, P.
Simplicial principal component analysis for density functions in Bayes spaces

24/2014 Ieva, F., Jackson, C.H., Sharples, L.D.
Multi-State modelling of repeated hospitalisation and death in patients with Heart Failure: the use of large administrative databases in clinical epidemiology

23/2014 Ieva, F., Paganoni, A.M., Tarabelloni, N.
Covariance Based Unsupervised Classification in Functional Data Analysis

22/2014 Arioli, G.
Insegnare Matematica con Mathematica

21/2014 Artina, M.; Fornasier, M.; Micheletti, S.; Perotto, S.
The benefits of anisotropic mesh adaptation for brittle fractures under plane-strain conditions

20/2014 Artina, M.; Fornasier, M.; Micheletti, S.; Perotto, S.
Anisotropic mesh adaptation for crack detection in brittle materials

19/2014 L.Bonaventura; R. Ferretti
Semi-Lagrangian methods for parabolic problems in divergence form