A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks

Géraldine Pichot, Jocelyne Erhel, Jean-Raynald De Dreuzy

A GENERALIZED MIXED HYBRID MORTAR METHOD FOR SOLVING FLOW IN STOCHASTIC DISCRETE FRACTURE NETWORKS*

G. PICHOT†, J. ERHEL‡, AND J.-R. DE DREUZY§

Abstract. The simulation of flow in fractured media requires handling both a large number of fractures and a complex interconnecting network of these fractures. Networks considered in this paper are three-dimensional domains made up of two-dimensional fractures intersecting each other and randomly generated. Due to the stochastic generation of fractures, intersections can be highly intricate. The numerical method must generate a mesh and define a discrete problem for any discrete fracture network (DFN). A first approach [Erhel, de Dreuzy, and Poirriez, *SIAM J. Sci. Comput.*, 31 (2009), pp. 2688–2705] is to generate a conforming mesh and to apply a mixed hybrid finite element method. However, the resulting linear system becomes very large when the network contains many fractures. Hence a second approach [Pichot, Erhel, and de Dreuzy, *Appl. Anal.*, 89 (2010), pp. 1629–1643] is to generate a nonconforming mesh, using an independent mesh generation for each fracture. Then a Mortar technique applied to the mixed hybrid finite element method deals with the nonmatching grids. When intersections do not cross or overlap, pairwise Mortar relations for each intersection are efficient [Pichot, Erhel, and de Dreuzy, 2010]. But for most random networks, discretized intersections involve more than two fractures. In this paper, we design a new method generalizing the previous one and that is applicable for stochastic networks. The main idea is to combine pairwise Mortar relations with additional relations for the overlapping part. This method still ensures the continuity of fluxes and heads and still yields a symmetric positive definite linear system. Numerical experiments show the efficiency of the method applied to complex stochastic fracture networks. We also study numerical convergence when reducing the mesh step. This method makes it easy to perform mesh optimization and appears to be a very promising tool to simulate flow in multiscale fracture networks.

Key words. mixed hybrid finite element method, Mortar method, fractured media

AMS subject classifications. 65N30, 65N55, 65N22

DOI. 10.1137/100804383

1. Introduction. Studying flow in fractured media is a challenging task due to the complex geometrical structures induced by the large variety of fractures and by their intricate interconnections. While most initial approaches relied on continuum approaches, discrete fracture network approaches (DFN) have been developed for sparsely fractured media [3], [4] that could not be homogenized. For the DFN approach, fractures are modeled individually in an otherwise impervious embedding medium generally called matrix. Flows are modeled by Poiseuille's law within the fractures and flow continuity is imposed at fracture intersections. Because of the lack of observations of fracture networks, fracture characteristics are mostly known as statistical distributions and modeling becomes stochastic. The fracture locations, shapes, and sizes follow statistical distributions derived from field data [23]. Note that a full generation of fracture networks by applying mechanical constraints is so

far not possible because of the difficulties in accounting for the coupled effects between fractures and in recovering the geological history of the externally applied constraints [22]. The DFN approach is now widely used for studying the influence of fractures on bulk hydraulic properties, both for phenomenological studies [5], [6], [7], [8], [9] and for natural field simulations [10], [11], [12], [13].

Several methods have been proposed to address the complexity issued by the fracture network structures. The first generation of methods consisted of simplifying drastically the model by replacing flows in the two-dimensional (2D) fractures by flows in well-chosen one-dimensional pipes [12], [7], [14]. Despite their simplicity, these initial methods are now discarded because it is not possible to quantify properly their approximation. This approach has been superseded by a second generation in which flows are solved within the 2D fracture planes. The geometrical configurations detrimental to the numerical methods are removed either by modifying locally the fracture network structure [15], [16], [17] or by performing a conformal discretization of all fracture intersections and borders at the network scale [1].

In this paper, we use a stochastic discrete fracture network generator [1] consistent with field observations similar in some statistical sense to the natural fracture networks. Because of the stochastic generation, the density of intersections sharply increases. Networks contain many X-shape intersections (Figures 5.1 and 5.6), that precludes the use of classical mesh generators. Out of 1640 generated networks, only 222 could be successfully meshed. This problem has been addressed in [1] by designing a new mesh generator. This new meshing procedure is based on staircase-like discretizations of fracture borders and intersections. It modifies slightly the initial position of the borders and intersections to produce a good-quality mesh and enable the simulation of flow within any stochastic network. However it requires building a conforming mesh at the network scale, constraining the mesh step on a global basis. On the contrary, it would be highly desirable to develop nonconforming methods coping with different mesh steps for the different fractures. The mesh step could be chosen according to the fractures properties or to their importance within the flow structure in order to increase the simulation accuracy and efficiency.

In a previous study [2], [18], we took a first step by adapting the mesh generator designed in [1] in order to mesh the fractures independently, which yields nonmatching grids at the fracture intersection. We have designed a Mortar method based on slave and master definitions under the assumption that no fracture intersections cross or overlap. In other words, intersections are common to exactly two fractures. It constraints the application of the method to simple networks. Simulation of more complex structures is required to enable stochastic modeling of natural fracture networks. But, for most complex networks, staircase-like discretizations introduce discretized intersection segments common to more than two fractures and the Mortar method developed in [2] cannot be applied. It motivates the generalization of the Mortar method to the junction of more than two subdomains.

Thus we present a generalization of the pairwise Mortar method suited for a nonlimited number of intersecting fractures per intersection segment. For intersection segments between more than two fractures, we show that the generalized Mortar method can build up on a combination of pairwise Mortar conditions with well-suited relations between flow and pressure variables at the intersections. After recalling the key characteristics of the fracture network geometry and governing flow equations in section 2 and the pairwise Mortar method in section 3, we define the generalized Mortar method in section 4 and show its performances on stochastic fracture network configurations in section 5.

2. Geometrical and hydraulic flow model.

2.1. Fractured network geometry. Multiscale implies large fractures that intersect numerous other fractures, yielding in turn complex local intersection configurations. Fracture generation follows the procedure described in [1].

Fractures are generated stochastically. They are ellipses or disks. Orientation and position distributions are taken as uniform. The broad natural fracture length distribution can be correctly modeled by a power-law distribution such as

$$(2.1) \qquad p(l)dl = \frac{1}{a-1}\frac{l^{-a}}{l_{\min}^{-a+1}}dl,$$

where $p(l)dl$ is the probability of observing a fracture with a length in the interval $[l, l+dl]$, $l_{\min}$ is the smallest fracture length, and a is a characteristic exponent [19], [23].

The cubic domain where the fractures lie is of characteristic size L. Consider we have N_f intersecting fractures that form a computational domain Ω. The boundary of Ω is composed of the border intersections between the cube and the fractures. Each fracture is denoted Ω_f, $f = 1, \ldots, N_f$.

We use the following notation within each fracture f:
- Let Γ_f be the border of the fracture f (which may be truncated by the edges of the cube).
- Let I_f be the set of all intersections within the fracture f. (For the sake of simplicity in the writing of the mathematical part, we assume that $\Gamma_f \cap I_f$ is of null measure.)
- We denote by Ω_f the fracture domain less the intersections, so that $\partial\Omega_f = \Gamma_f \cup I_f$.

2.2. Governing equations. DFNs are three-dimensional (3D) domains made of 2D fractures intersecting each other. We are interested in flow within the fractures, assuming that the rock matrix is impervious.

Classical laws governing the flux, mass conservation, and Poiseuille's law are assumed in each fracture. Continuity conditions are also imposed at the intersections of fractures. Longitudinal flux is neglected. Thus two conditions apply: continuity of the hydraulic head and continuity of the transversal flux at each intersection.

Boundary conditions on the cube edges are of Dirichlet or Neumann type. We denote by Γ_N (respectively, Γ_D, $\Gamma_D \neq 0$) the boundaries of the cube with Neumann (respectively, Dirichlet) boundary conditions. Since the rock matrix is supposed impervious, a homogeneous Neumann boundary condition is applied on the border of the fracture $\Gamma_{f,mat} = \Gamma_f \backslash \{(\Gamma_f \cap \Gamma_D) \cup (\Gamma_f \cap \Gamma_N)\}$.

Let N_i be the number of intersections of fractures, Σ_k be the kth intersection, $k = 1, \ldots, N_i$, and F_k be the set of fractures with Σ_k on the boundary.

Consider the coordinates $\mathbf{x} = (x, y)$, local to the plane Ω_f. For the unknown hydraulic head scalar function $p(\mathbf{x})$ and the flux per unit length function $\mathbf{u} = \mathbf{u}(\mathbf{x})$, we have

$$(2.2a) \qquad \nabla \cdot \mathbf{u}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega_f,$$

$$(2.2b) \qquad \mathbf{u}(\mathbf{x}) = -\mathcal{T}(\mathbf{x})\nabla p(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega_f,$$

$$(2.2c) \qquad p(\mathbf{x}) = p^D(\mathbf{x}) \text{ on } \Gamma_D \cap \Gamma_f,$$

$$(2.2d) \qquad \mathbf{u}(\mathbf{x}).\boldsymbol{\nu} = q^N(\mathbf{x}) \text{ on } \Gamma_N \cap \Gamma_f,$$

$$(2.2e) \qquad \mathbf{u}(\mathbf{x}).\boldsymbol{\mu} = \mathbf{0} \text{ on } \Gamma_{f,mat},$$

where $\boldsymbol{\nu}$ (respectively, $\boldsymbol{\mu}$) denotes the outward normal unit vector of the border $\Gamma_N \cap \Gamma_f$ (respectively, $\Gamma_{f,mat}$) with respect to the fracture f. The parameter $\mathcal{T}(\mathbf{x})$ is a given transmissivity field (unit $[\mathrm{m}^2.\mathrm{s}^{-1}]$). The function $\mathbf{f}(\mathbf{x}) \in L^2(\Omega_f)$ represents the sources/sinks.

Continuity conditions in each intersection are written [1], [17]

$$p_{k,f} = p_k \text{ on } \Sigma_k \ \forall f \in F_k, \tag{2.3a}$$

$$\sum_{f \in F_k} \mathbf{u}_{k,f}.\mathbf{n}_{k,f} = 0 \text{ on } \Sigma_k, \tag{2.3b}$$

where $p_{k,f}$ is the trace of hydraulic head on Σ_k in the fracture f, p_k is the unknown hydraulic head on the intersection Σ_k, and $\mathbf{u}_{k,f}.\mathbf{n}_{k,f}$ is the normal flux through Σ_k coming from the fracture Ω_f, with $\mathbf{n}_{k,f}$ the outward normal unit vector of the intersection Σ_k with respect to the fracture Ω_f. Equations (2.3a)–(2.3b) express the continuity of p and the mass balance of $\mathbf{u}$ across the intersections between fractures.

2.3. Mesh generation and notation. The flow model is solved using a mixed hybrid finite element method. Thus, it requires a mesh of the fractured network. For DFNs, mesh generation generally fails due to small angles between intersections of fractures. Specific meshing procedure have been proposed to mesh these networks. In [15], [17], [16], the geometry of the network is changed locally to get a mesh of good quality. As we are doing stochastic simulations, we rather need an automatic mesh generation suitable for any DFN. With the method proposed in [1], any DFN can be meshed but it requires matching grids at the intersection between fractures. The mesh generation we propose is more flexible in the sense that the grids may not match at the intersections, which allows us to choose a different mesh step from one fracture to another. For highly channelled flow, this meshing procedure will be very relevant, since only the fractures that govern the flow will be meshed finely, the others being meshed coarsely.

The mesh generation proposed in [1] leads to a conforming mesh at the fracture intersections.

The total mesh is called $\mathcal{T}_h = \cup_{f=1}^{N_f} \mathcal{T}_{h,f}$ and $N_{\mathcal{T}_h} = card(\mathcal{T}_h)$.

The procedure is the following:

1. (a) A first discretization of boundaries and intersections is done in three dimensions at the network scale by using elementary cubes,
   (b) The discretization of the boundaries and intersections within the fracture f is obtained by a projection of the previous voxel discretization within the fracture plane.
2. Some local corrections are applied to ensure some topological properties. For example, each boundary or intersection edge must have only one or two neighbors and each boundary or intersection node must have only two adjacent edges.
3. Once the borders and intersections are discretized, a 2D mesh of each fracture is built, using triangular elements.

The new meshing procedure we propose is the following. For each fracture f, we define a 2D grid whose center coincides with the ellipse (fracture) center and its discretization follows the direction of the ellipse's axes. The grid step is chosen as input data. The main advantage of the new procedure is that it can be chosen independently from one fracture to another. The grid size is chosen so that it totally encompasses the ellipse.

- For each fracture f,
  1. A first discretization of boundaries and intersections is done in two dimensions within each fracture by using elementary squares; it leads to staircase-like discretized borders.
  2. Some local corrections are applied to ensure some topological properties.
  3. Once the borders and intersections are discretized, a 2D mesh of each fracture, called $\mathcal{T}_{h,f}$, is built, using triangular elements.

An example is given in section 5.1. Figure 5.1 shows the geometry of the network and of one fracture (the black one) with its boundary and intersections. Figure 5.2 (left) shows the result of the first step using the meshing procedure proposed in [1] and Figure 5.3 (left) the corresponding 2D mesh (step 3). Figure 5.2 (right) displays the result of the first step of the new discretization procedure: a grid step has been chosen and the mesh centers are used to discretize the boundaries and intersections. The associated 2D mesh (step 3) is shown in Figure 5.3 (right).

Both methods are based on staircase-like representations of the borders and intersections. On one hand, this stage 1 modifies slightly the position of the borders and intersections. However, this approximation is motivated by the necessity of having a good-quality mesh for any generated stochastic network as, in most case, a classical mesh generator would fail to mesh the initial geometry [1]. Moreover this approximation can be lowered by a refinement of the 2D/3D grids, since the intersections and borders discretizations will stand closer to the original geometry. Section 5.4 shows the numerical convergence of the method with mesh refinement.

On the other hand, the staircase-like discretizations leads more likely to edges shared by more than two fractures. In such cases, the framework developed in [2] and recalled in section 3 needs to be generalized as the assumption that each intersection involves only two fractures is no longer valid. The generalized version is described in section 4.

The main difference between the two meshing methods comes from the first step. In [1], all borders and intersections are discretized at the network scale, leading to a conforming mesh at the intersections between fractures. With our method, fractures can be meshed independently, with a mesh step that can be different from one fracture to another. On one hand, the mesh generation is then easier. On the other hand, it leads to nonmatching grids at the intersections between fractures.

This paper explains how to handle, with the numerical method, such nonmatching grids.

We define

- $\mathcal{E}_{h,f,in}$: "inner" edges in $\mathcal{T}_{h,f}$, including boundaries, without intersection edges, $N_{f,in} = card(\mathcal{E}_{h,f,in})$;
- $\mathcal{E}_{h,f,\Sigma}$: intersection edges in I_f, $N_{f,\Sigma} = card(\mathcal{E}_{h,f,\Sigma})$;
- $\mathcal{E}_{h,f,k}$: intersection edges that discretize Σ_k within the fracture f;
- $\mathcal{E}_{h,\Sigma} = \cup_{f=1}^{N_f} \mathcal{E}_{h,f,\Sigma}$: all intersection edges, $N_\Sigma = card(\mathcal{E}_{h,\Sigma})$;
- $\mathcal{E}_{h,in} : \cup_{f=1}^{N_f} \mathcal{E}_{h,f,in}$: all inner edges, $N_{in} = card(\mathcal{E}_{h,in})$;
- $\mathcal{E}_{h,f} = \mathcal{E}_{h,f,in} \cup \mathcal{E}_{h,f,\Sigma}$: all edges within the fracture f;
- $\mathcal{E}_h = \cup_{f=1}^{N_f} \mathcal{E}_{h,f}$: all edges;
- $\mathcal{E}_{h,D} = \{E \in \mathcal{E}_h, E \in \Gamma_D\}$ and $\mathcal{E}_{h,N} = \{E \in \mathcal{E}_h, E \in \Gamma_N \cup \Gamma_{f,mat}\}$.

The next section recalls briefly the method for a mesh $\mathcal{T}_h$ satisfying the following property: within any fracture f, we assume that there is no common edge between two discretized intersections, that is,

$$(2.4) \qquad \forall f, \forall (k_1, k_2), \mathcal{E}_{h,f,k_1} \cap \mathcal{E}_{h,f,k_2} = \emptyset.$$

Section 4 presents the method for any DFN, where some edges may belong to several discretized intersections. It points out the differences with section 3.

3. Description of the numerical method for specific networks. In [2], [18], the networks satisfy the following assumption:

$$(3.1) \qquad \qquad \forall (k_1, k_2), \Sigma_{k_1} \cap \Sigma_{k_2} = \emptyset,$$

that is, intersections do not cross or overlap. Then each intersection Σ_k involves only two fractures.

In this section, we slightly enlarge the scope of the method since we make an assumption on the mesh rather than on the geometry:

$$(3.2) \qquad \qquad \forall f, \forall (k_1, k_2), \mathcal{E}_{h,f,k_1} \cap \mathcal{E}_{h,f,k_2} = \emptyset.$$

The only difference with the method proposed in [2], [18] is that now intersections may cross, but in a way they do not share any edge. We will point out that the only change in the method lies in additional contributions within the matrix of the linear system.

3.1. Basic principle of the method. Flow equations are solved with a Mortar mixed-hybrid finite element method described in [2], which allows us to deal with a nonconforming mesh at the intersections between fractures.

It consists of choosing for each intersection $\Sigma_k = \bar{\Omega}_f \cap \bar{\Omega}_{f'}$ a master fracture f and a slave fracture f'. Then, we denote by
- $\mathcal{E}_{h,k,m}$ the master discretization of Σ_k within the master fracture f and
- $\mathcal{E}_{h,k,s}$ the slave discretization of Σ_k within the slave fracture f'.

In the following, we will use the subscript s to refer to the slave side and m to refer to the master side. We denote by $N_{k,m}$ (respectively, $N_{k,s}$) the number of edges in $\mathcal{E}_{h,k,m}$ (respectively, $\mathcal{E}_{h,k,s}$) and $N_m = \sum_{k=1}^{N_i} N_{k,m}$, $N_s = \sum_{k=1}^{N_i} N_{k,s}$.

3.2. Algorithm for labeling with the master/slave property. The labeling algorithm consists of an outer loop on all intersections Σ_k, $(k = 1, N_i)$. One chooses arbitrarily the master fracture and the slave fracture. Then all edges in $\mathcal{E}_{h,k,m}$ are labeled with a master property and all edges in $\mathcal{E}_{h,k,s}$ are labeled with a slave property.

Then for each fracture f, we denote by
- $\mathcal{E}_{h,f,m}$ all edges labeled with a master property within the fracture f and
- $\mathcal{E}_{h,f,s}$ all edges labeled with a slave property within the fracture f.

With hypothesis (3.2), $\mathcal{E}_{h,f,m} \cap \mathcal{E}_{h,f,s}$ is empty.

3.3. Local unknowns in the fracture f. We denote by
- $\lambda_{E,f}$ the trace of hydraulic head unknown on the edge $E \in \mathcal{E}_{h,f}$ in the fracture f,
- $\mathbf{\Lambda}_f = (\lambda_{E,f})_{E \in \mathcal{E}_{h,f}}$ the vector of trace of hydraulic head unknowns on edges in $\mathcal{E}_{h,f}$,
- $\mathbf{\Lambda}_{f,in} = (\lambda_{E,f})_{E \in \mathcal{E}_{h,f,in}}$ the vector of trace of hydraulic head unknowns on edges in $\mathcal{E}_{h,f,in}$,
- $\mathbf{\Lambda}_{f,m} = (\lambda_{E,f})_{E \in \mathcal{E}_{h,f,m}}$ the vector of trace of hydraulic head unknowns on master edges in the fracture f,
- $\mathbf{\Lambda}_{f,s} = (\lambda_{E,f})_{E \in \mathcal{E}_{h,f,s}}$ the vector of trace of hydraulic head unknowns on slave edges in the fracture f.

On a fracture f, we distinguish inner edges, slave edges, and master edges so that the trace of hydraulic head unknowns form a vector

$$(3.3) \qquad \mathbf{\Lambda_f} = \begin{pmatrix} \mathbf{\Lambda_{f,in}} \\ \mathbf{\Lambda_{f,m}} \\ \mathbf{\Lambda_{f,s}} \end{pmatrix}.$$

We denote by $\mathbf{P_f}$ the vector of cell hydraulic head for all K in $\mathcal{T}_{h,f}$.

We define $Q_{E,f}$ the jump of flux through the edge E in the fracture f by

$$(3.4) \qquad Q_{E,f} = \sum_{K \in \mathcal{T}_{h,f}, \partial K \supset E} q_{K,E}.$$

We consider the vectors $\mathbf{Q_{f,m}} = (Q_{E,f})_{E \in \mathcal{E}_{h,f,m}}$ and $\mathbf{Q_{f',s}} = (Q_{E,f'})_{E \in \mathcal{E}_{h,f',s}}$.

3.4. Global unknowns in the fracture network. We introduce $\mathbf{\Lambda_{in}} = (\mathbf{\Lambda_{f,in}})_{\mathbf{f}}$ the vector containing all traces of hydraulic head unknowns on inner edges within the system and $\mathbf{P} = (\mathbf{P_f})_{\mathbf{f}}$ the vector containing all mean hydraulic head unknowns in the network.

We also define two global variables $\mathbf{\Lambda_s} = (\mathbf{\Lambda_{f,s}})_{\mathbf{f}}$ and $\mathbf{\Lambda_m} = (\mathbf{\Lambda_{f,m}})_{\mathbf{f}}$.

We define $\mathbf{Q_m} = (\mathbf{Q_{f,m}})_f$ (respectively, $\mathbf{Q_s} = (\mathbf{Q_{f,s}})_f$) the vector of size N_m (respectively, N_s) of jump of flux on master (respectively, slave) edges within the system.

3.5. Local equations. It is well known that in the context of the mixed hybrid finite element method, locally on each triangle K in $\mathcal{T}_{h,f}$, Poiseuille's law (2.2b) writes [20]

$$(3.5) \qquad \mathbf{B_K Q_K} = p_{K,f}\mathbf{e} - \mathbf{\Lambda_K}$$

with $\mathbf{Q_K}$ and $\mathbf{\Lambda}_K$ 3D vectors containing, respectively, the fluxes q_{K,E_i}, $i = 1, 2, 3$, and the traces of hydraulic head $\lambda_{E_i,f}$, $i = 1, 2, 3$, on each $E_i \subset \partial K$, $\mathbf{e} = (\begin{array}{ccc} 1 & 1 & 1 \end{array})^T$ and the 3×3 symmetric positive definite matrix $\mathbf{B_K}$ with elements

$$(3.6) \qquad (\mathbf{B_K})_{E_i,E_j} = \int_K \mathbf{w}_{K,E_i} \mathcal{T}_K^{-1} \mathbf{w}_{K,E_j} d\mathbf{x}$$

with the symmetric positive definite matrix $\mathcal{T}_K$ that represents the local transmissivity and $\mathbf{w}_{K,E_i}$ $i = 1, 2, 3$, $E_i \in \partial K$, $K \in \mathcal{T}_{h,f}$ the linearly independent vector basis functions of the lower order Raviart–Thomas space.

From (2.2a), local mass conservation applied on each triangle and continuity of the jump of fluxes have to be ensured through inner edges and at intersection edges:

$$(3.7) \qquad Q_{E,f} = \begin{cases} 0 \text{ if } E \in \mathcal{E}_{h,f,in} \backslash \Gamma_N, \\ q_E^N \text{ if } E \in \Gamma_N. \end{cases}$$

Equation 3.7 says that for border edges E in Γ_N, the so-called jump of flux $Q_{E,f}$ is simply equal to the Neumann imposed flux, q_E^N. Inner edges not belonging to Γ_N are shared by two triangles, $E = \partial K \cap \partial K'$, and the jump of flux must be zero: $q_{K,E} = -q_{K',E}$.

For intersection edges, the following Mortar conditions are required.

3.6. Mortar local conditions. Under assumption (3.2), on an intersection $\Sigma_k = \bar{\Omega}_f \cap \bar{\Omega}_{f'}$ (with f the master fracture and f' the slave fracture), we have [2]

$$(3.8a) \qquad (\lambda_{E,f'})_{E \in \mathcal{E}_{h,k,s}} = \mathbf{C_k}(\lambda_{E,f})_{E \in \mathcal{E}_{h,k,m}},$$

$$(3.8b) \qquad (Q_{E,f})_{E \in \mathcal{E}_{h,k,m}} + \mathbf{C_k^T}(Q_{E,f'})_{E \in \mathcal{E}_{h,k,s}} = \mathbf{0}$$

with $\mathbf{C_k}$ a matrix of size $N_{k,s} \mathrm{x} N_{k,m}$ that represents the L^2-projection from the master side to the slave side. Its coefficients C_{ln}, $l \in \{1,\dots,N_{k,s}\}$, $n \in \{1,\dots,N_{k,m}\}$, are the ratio of the intersection length of a slave edge E_l and a master edge E_n over the length of E_l:

$$(3.9) \qquad C_{ln} = \left(\frac{|E_n \cap E_l|}{|E_l|} \right),$$

where $|E|$ stands for the length of the edge E.

Equation (3.8b) expresses the continuity of the jump of flux at the intersection Σ_k and (3.8a) the continuity of heads at the intersection Σ_k. The matrix $\mathbf{C_k}$ takes into account the differences in the discretizations between the master and the slave sides of Σ_k. In the case of a matching grid discretization of the intersection, C_k is the identity matrix.

3.7. Mortar global conditions. We define the matrix $\mathbf{C}$ as a block matrix of dimension $N_s \times N_m$ with blocks $(\mathbf{C_k})$.

The previous relations (3.8a) and (3.8b) are easily rewritten in terms of the global variables as

$$(3.10a) \qquad \mathbf{\Lambda_s} = \mathbf{C}\mathbf{\Lambda_m},$$

$$(3.10b) \qquad \mathbf{Q_m} = -\mathbf{C}^T\mathbf{Q_s}.$$

3.8. Derivation of the linear system. Inverting the matrix $\mathbf{B_K}$ in (3.5) and using (3.10a), we assemble the mass conservation equation in each triangle into a system involving global variables:

$$(3.11) \qquad \mathbf{D}\,\mathbf{P} - \begin{pmatrix} \mathbf{R_{in}} & \mathbf{R_m} + \mathbf{R_s}\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} = \mathbf{F},$$

where $\mathbf{D}$ is a $N_{\mathcal{T}_h} \times N_{\mathcal{T}_h}$ diagonal matrix, $\mathbf{R_{in}}$ is a sparse matrix of dimension $N_{\mathcal{T}_h} \times N_{in}$, $\mathbf{R_m}$ is a sparse matrix of dimension $N_{\mathcal{T}_h} \times N_m$, $\mathbf{R_s}$ is a sparse matrix of dimension $N_{\mathcal{T}_h} \times N_s$, $\mathbf{C}$ is of dimension $N_s \times N_m$, and $\mathbf{F}$ is a vector of dimension $N_{\mathcal{T}_h}$, which corresponds to the source/sink function as well as to the imposed hydraulic head given by the Dirichlet boundary conditions.

Also, by inverting the matrix $\mathbf{B_K}$ in (3.5), we eliminate the flux unknowns in (3.7). Using global variables and the relation (3.10a), we eliminate $\mathbf{\Lambda_s}$, yielding

$$(3.12) \qquad \mathbf{R_{in}^T}\,\mathbf{P} - \begin{pmatrix} \mathbf{M_{in}} & \mathbf{M_m} + \mathbf{M_s}\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} + \mathbf{V_{in}} = \mathbf{0},$$

where $\mathbf{M_{in}}$ a sparse matrix of dimension $N_{in} \times N_{in}$, $\mathbf{M_m}$ of dimension $N_{in} \times N_m$, and $\mathbf{M_s}$ of dimension $N_{in} \times N_s$.

Following the same procedure in condition (3.10b), one has

$$(3.13)$$

$$(\mathbf{R_m^T} + \mathbf{C}^T\mathbf{R_s^T})\,\mathbf{P} - \begin{pmatrix} \mathbf{M_m^T} + \mathbf{C}^T\mathbf{M_s^T} & \mathbf{B_m} + \mathbf{C}^T\mathbf{B_s}\mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} + \mathbf{V_m} = \mathbf{0}$$

with $\mathbf{B_s}$ a square matrix of dimension $N_s \times N_s$ and $\mathbf{B_m}$ a square matrix of dimension $N_m \times N_m$ that has the form

$$(3.14) \qquad \mathbf{B_m} = \mathbf{B_{m,m}} + \mathbf{B_{m,s}} \mathbf{C} + \mathbf{C}^T \mathbf{B_{m,s}^T},$$

where $\mathbf{B_{m,m}}$ is the matrix of flux contributions from master edges and $\mathbf{B_{m,s}}$ is the matrix of flux contribution coming from slave edges. The terms $\mathbf{B_{m,s}}$ appear when triangles contain both master and slave edges. This is the only difference with the method presented in [2] because we use hypothesis (3.2) (on the mesh) instead of hypothesis (3.1) (on the network geometry).

Let us define

$$(3.15) \qquad \begin{aligned} \mathbf{M} &= \begin{pmatrix} \mathbf{M_{in}} & \mathbf{M_m} + \mathbf{M_s} \mathbf{C} \\ \mathbf{M_m}^T + \mathbf{C}^T \mathbf{M_s}^T & \mathbf{B_m} + \mathbf{C}^T \mathbf{B_s} \mathbf{C} \end{pmatrix}, \\ \mathbf{\Lambda} &= \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix}, \mathbf{R} = \begin{pmatrix} \mathbf{R_{in}} & \mathbf{R_m} + \mathbf{R_s} \mathbf{C} \end{pmatrix}, \\ \mathbf{S} &= \mathbf{M} - \mathbf{R}^T \mathbf{D}^{-1} \mathbf{R}. \end{aligned}$$

The matrix $\mathbf{M}$ and the Schur complement matrix $\mathbf{S}$ are proved to be symmetric positive definite in presence of Dirichlet boundary conditions [2]. The proof stays the same when using hypothesis (3.2). The linear system to solve is then

$$(3.16) \qquad \mathbf{S} \mathbf{\Lambda} = \mathbf{V} + \mathbf{R} \mathbf{D}^{-1} \mathbf{F}.$$

Once the system (3.16) is solved, (3.10a) allows us to recover the trace of hydraulic head values on the slave edges. Finally the 3D vectors $\mathbf{Q_K}$ containing the fluxes $q_{K,E_i}, i = 1, 2, 3$, on each triangle $K \in \mathcal{T}_h$ are derived locally for each $K \in \mathcal{T}_h$ using the relation (3.5).

4. Description of the numerical method for general networks.

4.1. Challenge. This section is the main contribution of this paper. It explains how to handle nonmatching mesh at the intersections between fractures for any DFN. The previous section was restricted to network satisfying hypothesis (3.2). The complexity in the geometry of fracture networks and their meshing cannot guarantee that this assumption is strictly satisfied. With the generation procedure we use (section 2.3), a fracture may contain many X-shaped intersections since fractures are allowed to intersect each other in a very intricate way. As pointed out in section 2.3, discretized intersections inside a fracture are likely to share an edge, which is then common to at least three fractures. If hypothesis (3.2) is not satisfied anymore, the challenge is that one may find edges shared by several intersections. Those edges will then be labeled with several properties (slaves and/or masters) at the same time. We propose general Mortar conditions to ensure continuity of fluxes and heads at the intersections. We assume that edges are not cut but are shared totally; this is satisfied with the meshing procedure we use (see section 2.3).

4.2. Labeling with the master/slave property. The algorithm remains the same as for specific networks (see subsection 3.2).

For each fracture f, we still denote by
- $\mathcal{E}_{h,f,m}$ all edges labeled with a master property within the fracture f, $N_{f,m} = card(\mathcal{E}_{h,f,m})$;
- $\mathcal{E}_{h,f,s}$ edges labeled with a slave property within the fracture f, $N_{f,s} = card(\mathcal{E}_{h,f,s})$.

The main difference is that now some intersection edges may be labeled with several master and/or slave properties. The consequences are that $\mathcal{E}_{h,f,m} \cap \mathcal{E}_{h,f,s}$ may be not empty and that an edge may be labeled several times with the same property.

4.3. Local unknowns in the fracture f. For each edge $E \in \mathcal{E}_{h,f}$, we define $\lambda_{E,f}$ its trace of hydraulic head value.

Now for an intersection edge $E \in \mathcal{E}_{h,f,\Sigma}$, there are $n_E = s_E + m_E$ new associated unknowns with m_E the number of times the edge E is labeled with a master property (it can be 0 if the edge is only a slave) and s_E the number of times it is labeled with a slave property (it can be 0 if the edge is only a master).

These n_E new unknowns are as follows:
- $\lambda_{E,f,m,k}$ its additional trace of hydraulic head value as master edge for the intersection Σ_k in the fracture f (if E is labeled with a master property for this intersection). We use also the notation $\lambda_{E,f,m,k} = \lambda_{E,f,m_k}$ with the index $m_k = (m,k)$, $m_k = 1, \ldots, m_E$.
- $\lambda_{E,f,s,k}$ its additional trace of hydraulic head value as slave edge for the intersection Σ_k in the fracture f (if E is labeled with a slave property for this intersection). We use also the notation $\lambda_{E,f,s,k} = \lambda_{E,f,s_k}$ with the index $s_k = (s,k)$, $s_k = 1, \ldots, s_E$.

New relations for each intersection edge $E \in \mathcal{E}_{h,f,\Sigma}$ are thus necessary between those new unknowns and $\lambda_{E,f}$.

We denote by $\mathbf{\Lambda_{f,\Sigma}} = (\lambda_{E,f})_{E \in \mathcal{E}_{h,f,\Sigma}}$ the vector of final trace of hydraulic head unknowns $\lambda_{E,f}$ on the intersection edges E in the fracture f of size $N_{f,\Sigma}$.

We define the vector $\mathbf{\Lambda_{f,m}} = (\lambda_{E,f,m})_{E \in \mathcal{E}_{h,f,m}}$ of size $N_{f,m}$ and the vector $\mathbf{\Lambda_{f,s}} = (\lambda_{E,f,s})_{E \in \mathcal{E}_{h,f,s}}$ of size $N_{f,s}$ of additional unknowns.

In a fracture f, we still distinguish inner edges and intersection edges so that the trace of hydraulic head unknowns form a vector

$$(4.1) \qquad \mathbf{\Lambda_f} = \begin{pmatrix} \mathbf{\Lambda_{f,in}} \\ \mathbf{\Lambda_{f,\Sigma}} \end{pmatrix}.$$

For each edge $E \in \mathcal{E}_{h,f}$ in a fracture f, we also consider the jump of flux $Q_{E,f}$ through the edge E.

For an intersection edge $E \in \mathcal{E}_{h,f,\Sigma}$, there are also new unknowns:
- $Q_{E,f,m,k}$ partial jump of flux through the edge E master for the intersection Σ_k.
- $Q_{E,f,s,k}$ partial jump of flux through the edge E slave for the intersection Σ_k.

For a given intersection Σ_k, we denote $\mathbf{Q_{f,m,k}} = (Q_{E,f})_{E \in \mathcal{E}_{h,f,m,k}}$ and $\mathbf{Q_{f',s,k}} = (Q_{E,f'})_{E \in \mathcal{E}_{h,f',s,k}}$.

We consider the vectors $\mathbf{Q_{f,m}} = (Q_{E,f})_{E \in \mathcal{E}_{h,f,m}}$ and $\mathbf{Q_{f',s}} = (Q_{E,f'})_{E \in \mathcal{E}_{h,f',s}}$.

We denote by $\mathbf{Q_{f,\Sigma}} = (\mathbf{Q_{E,f}})_{E \in \mathcal{E}_{h,f,\Sigma}}$ the vector of jumps of flux through the intersection edges within the fracture f.

4.4. Global variables in the fracture network. We denote by $\mathbf{\Lambda_\Sigma} = (\Lambda_{f,\Sigma})_f$ the global vector of final trace of hydraulic head unknowns in the system.

We denote by $N_m = \sum_{f=1}^{N_f} N_{f,m}$, $N_s = \sum_{f=1}^{N_f} N_{f,s}$, and $N_\Sigma = \sum_{f=1}^{N_f} N_{f,\Sigma}$. We define the global unknowns: $\mathbf{\Lambda_m} = (\mathbf{\Lambda_{f,m}})_f$ and $\mathbf{\Lambda_s} = (\mathbf{\Lambda_{f,s}})_f$.

We denote by $\mathbf{Q_\Sigma} = (\mathbf{Q_{f,\Sigma}})_f$ the vector of jumps of flux through the intersection edges within the system.

Similarly, we form a vector containing all the jumps of flux through the master edges in the system, $\mathbf{Q_m} = (\mathbf{Q_{f,m}})_f$, and a vector containing all the jumps of flux through the slave edges in the system, $\mathbf{Q_s} = (\mathbf{Q_{f,s}})_f$.

4.5. Mortar local conditions. Now we have to define new relations between the different unknowns at an intersection edge. We choose to allocate a mean value of hydraulic heads to $\lambda_{E,f}$,

$$(4.2) \qquad \lambda_{E,f} = \frac{1}{n_E} \left(\sum_{m_k=1}^{m_E} \lambda_{E,f,m_k} + \sum_{s_k=1}^{s_E} \lambda_{E,f,s_k} \right),$$

that is, the final trace of hydraulic head value on the edge E is equal to the mean of the additional temporary trace of hydraulic head values on the edge E.

This choice seems reasonable since the consideration of additional unknowns on a same edge is a numerical trick to deal with several labels (master/slave). There is no physical reason to introduce a different weighting rule.

From relations (4.2), we have

$$(4.3) \qquad \lambda_{E,f} = a_{m,E} \begin{pmatrix} \lambda_{E,f,m_1} \\ \dots \\ \lambda_{E,f,m_E} \end{pmatrix} + a_{s,E} \begin{pmatrix} \lambda_{E,f,s_1} \\ \dots \\ \lambda_{E,f,s_E} \end{pmatrix}$$

with $a_{m,E}$ the vector of size $1 \times m_E$ and $a_{s,E}$ the vector of size $1 \times s_E$ such that

$$(4.4) \qquad a_{m,E} = \left(\frac{1}{n_E} \ \dots \ \frac{1}{n_E} \right),$$
$$a_{s,E} = \left(\frac{1}{n_E} \ \dots \ \frac{1}{n_E} \right).$$

As previously in (3.8a), we apply Mortar relations between master and slave edges of the same intersection $\Sigma_k = \bar{\Omega}_f \cap \bar{\Omega}_{f'}$:

$$(4.5) \qquad (\lambda_{E,f',s,k})_{E \in \mathcal{E}_{h,f',s,k}} = \mathbf{C_k} (\lambda_{E,f,m,k})_{E \in \mathcal{E}_{h,f,m,k}}.$$

Now we define the equations for the jump of flux through intersection edges. We first define the dual relations of (4.3) for all the flux unknowns associated to one edge:

$$(4.6) \qquad Q_{E,f,m,k} = \frac{1}{n_E} Q_{E,f},$$
$$Q_{E,f,s,k} = \frac{1}{n_E} Q_{E,f}.$$

As previously in (3.8b), we apply Mortar relations between master and slave edges of the same intersection $\Sigma_k = \bar{\Omega}_f \cap \bar{\Omega}_{f'}$:

$$(4.7) \qquad (Q_{E,f,m,k})_{E \in \mathcal{E}_{h,f,m,k}} + \mathbf{C_k^T} (Q_{E,f',s,k})_{E \in \mathcal{E}_{h,f',s,k}} = \mathbf{0},$$

4.6. Mortar global relations. Relation (4.3) can be written for each edge E within the fracture f, so that we have

$$(4.8) \qquad \mathbf{\Lambda_{f,\Sigma}} = \mathbf{A_{f,m}} \mathbf{\Lambda_{f,m}} + \mathbf{A_{f,s}} \mathbf{\Lambda_{f,s}}$$

with $\mathbf{A_{m,f}}$ the matrix of size $N_{f,\Sigma} \times N_{m,f}$ obtained from the vector $a_{m,E}$ by completing by 0 value where needed and $\mathbf{A_{s,f}}$ the matrix of size $N_{f,\Sigma} \times N_{s,f}$ obtained from the vector $a_{s,E}$ by completing by 0 value where needed.

We denote by $\mathbf{A_m} = (\mathbf{A_{f,m}})_f$ the matrix of size $N_\Sigma \times N_m$ and by $\mathbf{A_s} = (\mathbf{A_{f,s}})_f$ the matrix of size $N_\Sigma \times N_s$.

In terms of global variables, we still have relation (3.10a):

$$(4.9) \qquad \mathbf{\Lambda_s} = \mathbf{C\Lambda_m}$$

with $\mathbf{C}$ an intersection block matrix of dimension $N_s \times N_m$.

Combining (4.8) and (4.9), we get

$$(4.10) \qquad \mathbf{\Lambda_\Sigma} = \mathbf{A_m\Lambda_m} + \mathbf{A_s\Lambda_s} = (\mathbf{A_m} + \mathbf{A_sC})\mathbf{\Lambda_m}.$$

We now set the global relations for the flux.

From relations (4.7), we have

$$(4.11) \qquad \mathbf{Q_m} + \mathbf{C}^T\mathbf{Q_s} = 0.$$

Conditions (4.6) are written

$$(4.12) \qquad \begin{aligned} \mathbf{A_m^T Q_\Sigma} &= \mathbf{Q_m}, \\ \mathbf{A_s^T Q_\Sigma} &= \mathbf{Q_s}. \end{aligned}$$

Combining (4.11) and (4.12), we get

$$(4.13) \qquad (\mathbf{A_m^T} + \mathbf{C}^T\mathbf{A_s^T})\mathbf{Q_\Sigma} = 0.$$

4.7. Derivation of the linear system. We use the same procedure as in the previous section.

Using relations (4.9) and (4.10), we eliminate $\mathbf{\Lambda_s}$ and $\mathbf{\Lambda_\Sigma}$ to keep only the unknowns $\mathbf{\Lambda_m}$.

Inverting the matrix $\mathbf{B_K}$ in (3.5), we assemble the mass conservation equation on each triangle into a system involving global variables:

$$(4.14) \qquad \mathbf{D\,P} - \begin{pmatrix} \mathbf{R_{in}} & \mathbf{R_\Sigma}(\mathbf{A_m} + \mathbf{A_sC}) \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} = \mathbf{F},$$

where $\mathbf{D}$ is a $N_{\mathcal{T}_h} \times N_{\mathcal{T}_h}$ diagonal matrix, $\mathbf{R_{in}}$ is a sparse matrix of dimension $N_{\mathcal{T}_h} \times N_{in}$, $\mathbf{R_\Sigma}$ a sparse matrix of dimension $N_{\mathcal{T}_h} \times N_\Sigma$, $\mathbf{C}$ is of dimension $N_s \times N_m$, and $\mathbf{F}$ is a vector of dimension $N_{\mathcal{T}_h}$, which corresponds to the source/sink function as well as to the imposed hydraulic head given by the Dirichlet boundary conditions.

Also, by inverting the matrix $\mathbf{B_K}$ in (3.5), we eliminate the flux unknowns in (3.7):

$$(4.15) \qquad \begin{pmatrix} \mathbf{M_{in}} & \mathbf{M_\Sigma}(\mathbf{A_m} + \mathbf{A_sC}) \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} - \mathbf{R_{in}^T P} - \mathbf{V_{in}} = 0,$$

where $\mathbf{M_{in}}$ a sparse matrix of dimension $N_{in} \times N_{in}$ with $\mathbf{M_\Sigma}$ of dimension $N_{in} \times N_\Sigma$.

Following the same procedure in condition (4.13), one has

$$(4.16) \quad \begin{pmatrix} (\mathbf{A_m^T} + \mathbf{C}^T\mathbf{A_s^T})\mathbf{M_\Sigma^T} & (\mathbf{A_m^T} + \mathbf{C}^T\mathbf{A_s^T})\mathbf{B_\Sigma}(\mathbf{A_sC} + \mathbf{A_m}) \end{pmatrix} \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix}$$
$$- \left((\mathbf{A_m^T} + \mathbf{C}^T\mathbf{A_s^T})\mathbf{R_\Sigma^T}\right)\mathbf{P} - \mathbf{V}_m = 0$$

with $\mathbf{B_\Sigma}$ a square matrix of size $N_\Sigma \times N_\Sigma$.

Remark. When hypothesis (3.2) is satisfied, $\mathbf{M_\Sigma A_s} = \mathbf{M_s}$, $\mathbf{M_\Sigma A_m} = \mathbf{M_m}$, $\mathbf{R_\Sigma A_s} = \mathbf{R_s}$, $\mathbf{R_\Sigma A_m} = \mathbf{R_m}$, $\mathbf{A_s^T B_\Sigma A_s} = \mathbf{B_s}$, $\mathbf{A_m^T B_\Sigma A_m} = \mathbf{B_{m,m}}$, $\mathbf{A_m^T B_\Sigma A_s} = \mathbf{B_{m,s}}$ with the notation used in section (3.8).

We denote by

$$
(4.17) \quad
\begin{aligned}
\mathbf{M} &= \begin{pmatrix} \mathbf{M_{in}} & \mathbf{M_\Sigma(A_m + A_sC)} \\ (\mathbf{A_m^T + C^T A_s^T})\mathbf{M_\Sigma^T} & (\mathbf{A_m^T + C^T A_s^T})\mathbf{B_\Sigma(A_sC + A_m)} \end{pmatrix}, \\
\mathbf{\Lambda} &= \begin{pmatrix} \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix}, \ \mathbf{R} = \begin{pmatrix} \mathbf{R_{in}} & \mathbf{R_\Sigma(A_m + A_sC)} \end{pmatrix}, \\
\mathbf{S} &= \mathbf{M} - \mathbf{R}^T \mathbf{D}^{-1} \mathbf{R}.
\end{aligned}
$$

PROPOSITION 4.1. *Assuming the transmissivity is locally symmetric positive definite, the matrix*

$$
(4.18) \quad \mathcal{J} = \begin{pmatrix} \mathbf{D} & -\mathbf{R} \\ -\mathbf{R}^T & \mathbf{M} \end{pmatrix}
$$

is symmetric and, with the presence of Dirichlet boundary conditions within at least one fracture, it is positive definite.

Proof. The proof is similar to the one given in [2]. We use relation (4.10) to introduce $\mathbf{\Lambda_\Sigma}$ and to identify $\mathbf{\Lambda_f}$ in the expression.

For any nonzero vector $\begin{pmatrix} \mathbf{P} \\ \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix}$ of size $card(\mathcal{T}_h) \times card(\mathcal{E}_{h,in}) \times N_m$, we have

$$
(4.19) \quad
\begin{aligned}
&\begin{pmatrix} \mathbf{P}^T & \mathbf{\Lambda_{in}^T} & \mathbf{\Lambda_m^T} \end{pmatrix} \mathcal{J} \begin{pmatrix} \mathbf{P} \\ \mathbf{\Lambda_{in}} \\ \mathbf{\Lambda_m} \end{pmatrix} \\
&= \mathbf{P}^T\mathbf{DP} - 2\mathbf{P^T R_{in}\Lambda_{in}} - 2\mathbf{P^T R_\Sigma(A_m + A_sC)\Lambda_m} \\
&\quad + \mathbf{\Lambda_{in}^T M_{in}\Lambda_{in}} + \mathbf{\Lambda_{in}^T M_\Sigma(A_m + A_sC)\Lambda_m} \\
&\quad + ((\mathbf{A_m + A_sC})\mathbf{\Lambda_m})^T\mathbf{M_\Sigma}^T\mathbf{\Lambda_{in}} \\
&\quad + ((\mathbf{A_m + A_sC})\mathbf{\Lambda_m})^T\mathbf{B_\Sigma(A_m + A_sC)\Lambda_m} \\
&= \mathbf{P}^T\mathbf{DP} - 2\mathbf{P^T R_{in}\Lambda_{in}} - 2\mathbf{P^T R_\Sigma\Lambda_\Sigma} + \mathbf{\Lambda_{in}^T M_{in}\Lambda_{in}} + \mathbf{\Lambda_{in}^T M_\Sigma\Lambda_\Sigma} \\
&\quad + \mathbf{\Lambda_\Sigma^T M_\Sigma}^T\mathbf{\Lambda_{in}} + \mathbf{\Lambda_\Sigma^T B_\Sigma\Lambda_\Sigma} \\
&= \sum_{f=1}^{N_f} \left(\mathbf{\Lambda_f^T M_f\Lambda_f} + \mathbf{P_f^T D_f P_f} - 2\mathbf{P_f^T R_f\Lambda_f} \right)
\end{aligned}
$$

with $\mathbf{M}_f$, $\mathbf{D}_f$, and $\mathbf{R}_f$ the matrices containing the contribution associated to the triangles K within the fracture f and

$$
(4.20) \quad \mathcal{J}_\mathbf{f} = \begin{pmatrix} \mathbf{D}_f & -\mathbf{R}_f \\ -\mathbf{R}_f^T & \mathbf{M}_f \end{pmatrix}.
$$

We get a system within each fracture involving the matrix $\mathcal{J}_\mathbf{f}$ which is proved to be positive definite in the presence of Dirichlet boundary conditions and provided the transmissivity is locally symmetric positive definite [20]. $\quad\square$

The Schur complement matrix $\mathbf{S}$ is then symmetric positive definite in the presence of Dirichlet boundary conditions. Once the system

$$(4.21) \qquad \mathbf{S\Lambda} = \mathbf{V} + \mathbf{R^T D^{-1} F}$$

is solved in $\mathbf{\Lambda}$, (4.10) allows us to recover the trace of hydraulic head values on the intersection edges $\mathbf{\Lambda_\Sigma}$. Finally, the 3D vectors $\mathbf{Q_K}$ containing the fluxes $q_{K,E_i}, i = 1, 2, 3$, on each triangle $K \in \mathcal{T}_h$ are derived locally for each $K \in \mathcal{T}_h$ using the relation (3.5).

5. Numerical simulations. This section presents the benefits of using a nonmatching grid approach. We also describe two examples of simulations in DFNs with permeameter boundary conditions and analyze the convergence of the numerical method (when the mesh is refined).

Many DFNs have been generated randomly. Several criteria have been checked for each DFN to test the consistency of the results:
- Null sum of the fluxes over all the system;
- Null sum of the fluxes over all intersections between fractures;
- Boundary conditions satisfied (notice, however, that edges that are both intersection and on a border are uppermost treated as intersection);
- Continuity of the flux on inner edges.

5.1. Nonmatching versus matching grids. In this section we compare the mesh obtained with our meshing procedure (presented in section (2.3)), referred to as the 2D grid, with the mesh generated according to [1], referred to as the 3D grid. Let us take the example of a fracture, say, fracture 1 (in black) embedded in the network. Its intersections and boundaries are shown in Figure 5.1.

The discretization procedure of the boundaries and intersections by the means of a 3D grid (and projection) and a 2D grid is given in Figure 5.2.

The corresponding meshes are presented in Figure 5.3.

The mesh generated by using our 2D grid method is of better quality since there is no projection step on the fracture plane as is the case with a 3D grid. Moreover, the benefit is nonconforming discrete intersections between fractures. It is thus easy to refine or coarse the mesh.

5.2. Example 1: Nonrealistic test case. We first describe a nonrealistic network where all fractures intersect at the same place. It is a typical configuration to get multiply defined edges. This is an extreme case but it is useful to validate the method. Figure 5.4 shows the geometry of 10 fractures in intersection and the
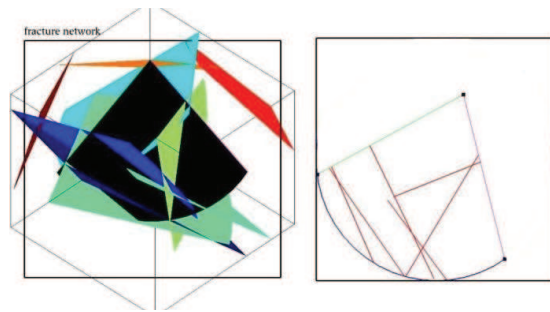

FIG. 5.1. *Left: the 3D network; right: the black fracture with its boundaries and intersections.*
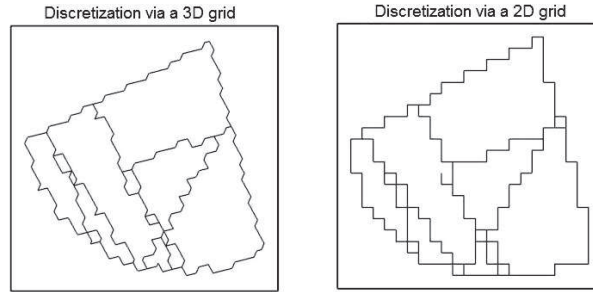
FIG. 5.2. *Discretization of the boundaries and intersections. Left: via 3D grid (and projection); right: via a 2D grid.*
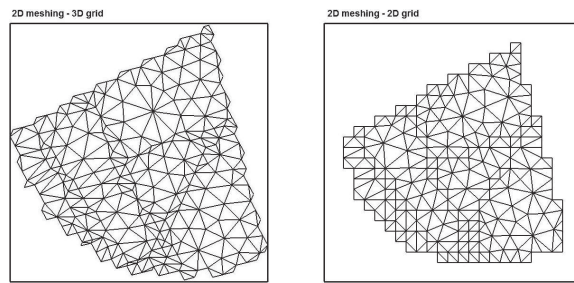

FIG. 5.3. *Associated 2D mesh. Left: 3D grid (and projection); right: 2D grid.*
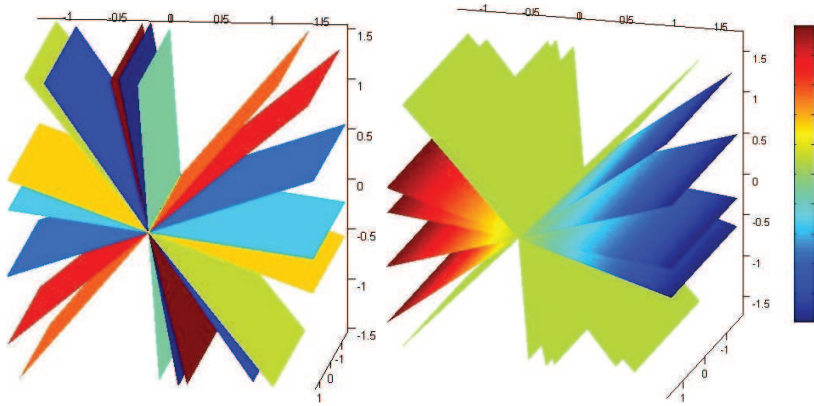

FIG. 5.4. *Nonrealistic example. Left, its geometry; right: mean hydraulic head.*

associated mean hydraulic head for classic permeameter boundary conditions (head value of 0m on the right side of the cube, 10m on the left side, and null flux on the other sides). The cube side L is 2 m and the mesh step is the same in each fracture and equal to 0.08 m. The transmissivity tensor is homogeneous and equal to $\mathcal{T} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ m^2.s^{-1}. In this example, we have $N_{in} = 81{,}589$ inner edges, $N_\Sigma = 370$ intersection edges, $N_m = 1665$ master hydraulic head unknowns, and $N_s = 1665$ slave hydraulic head unknowns. As the mesh step is the same within all fractures, the discretization is matching at the common intersection and is made of 37 edges. We have 45 pairs of
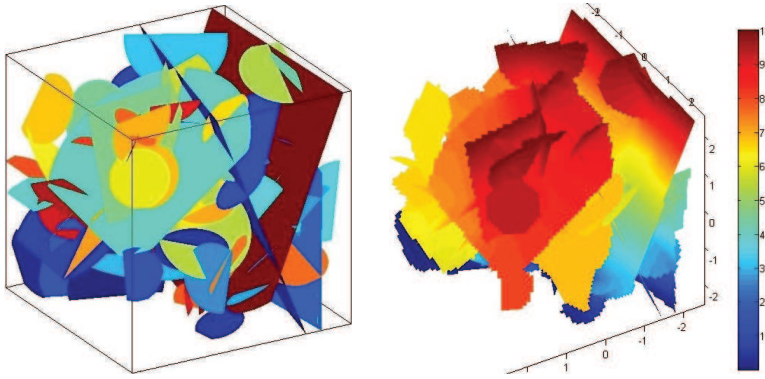
FIG. 5.5. *Example of DFN, test case* 1: *left, its geometry; right: mean hydraulic head.*

TABLE 5.1
Random distributions of parameters for the generated DFN, Example 2.

Parameter	Random distribution
length	power law
shape	disks
position	uniform
orientation	uniform

TABLE 5.2
Values of the input parameters in Example 2.

Parameter	Value
a	3.5
$L/l_{\min}$	5
Mesh step	test case 1 : 0.08, test case 2: 0.08 and 0.16
Number of fractures	generated:100 connected cluster: 91

two fractures. Then the numbers of master and slave hydraulic head unknowns are as expected since $N_m = 45 * 37$ and $N_m = N_s$.

Figure 5.4 shows that the mean head is as expected. Moreover, the method does not create any artificial flux: the sum of fluxes over the whole system is equal to 2.11×10^{-12} m³.s⁻¹ and the sum of all the fluxes on intersections gives 7.42×10^{-13} m³.s⁻¹.

5.3. Example 2: Random DFN. The second example is a general random network. It does not satisfy hypothesis (3.2).

Figure 5.5 shows the geometry. Tables 5.1 and 5.2 give the random distributions and parameters used for the simulations. We perform two test cases, one with the same mesh step within all fractures and the second with some fractures being coarsened. Among 100 generated fractures, the connected cluster, defined as the ensemble of fractures connected to the limits of the system, has 91 fractures in intersection. We impose classical permeameter boundary conditions: head value of 10 m on the top side of the cube, 0 m on the bottom side, and null flux on the other sides. The transmissivity tensor is equal to $\mathcal{T} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ m².s⁻¹. The parameter a of the power law (2.1) is equal to 3.5 and the ratio $L/l_{\min}$ is equal to 5 with $l_{\min}$ the smallest fracture length and L the cube side.
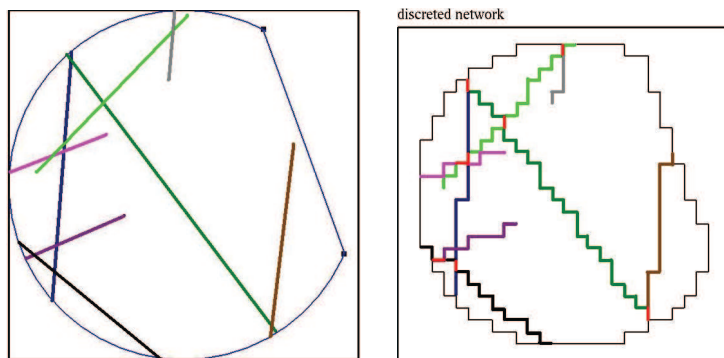
FIG. 5.6. *Left: one fracture and its intersections; right: its border and intersections discretizations: common intersection edges are in red.*

Due to the mesh generation, one may find some edges that belong to two or more discretized intersections within a fracture. One fracture of the network is shown in Figure 5.6: common edges are in red.

The first test case deals with a mesh step of the 2D grid used for the discretization of border and intersections within each fracture of 0.08 m. In this example, $N_{in} = 212{,}721$ inner edges, $N_\Sigma = 8380$ intersection edges, $N_m = 4484$ master hydraulic head unknowns, and $N_s = 4418$ slave hydraulic head unknowns. Here $N_m \neq N_s$ and $N_\Sigma < N_m + N_s$ since there are some edges common to more than one intersection (like edges in red in Figure 5.6). Moreover, due to the use of a 2D grid (see section 2.3), even using the same mesh step within all fractures does not guarantee that a given intersection between two fractures is discretized with the same number of edges in both fractures. This is not a problem since the method is built to handle such nonconforming meshes at intersections.

The sum of all the fluxes on intersections gives 7.2×10^{-13} m^3.s^{-1}. The sum of fluxes over the whole system is equal to -8.51×10^{-12} m^3.s^{-1}. The equivalent permeability defined as the ratio between the input flux over the product of the domain length by $\Delta h = 10$ is equal to 0.789 m.s^{-1}.

In the second test case, we use the same network but with 40 fractures having their border and intersections discretized with a 2D grid having a mesh step of 0.08 like in the previous simulation and others with a 2D grid having a mesh step of 0.16 m. Now $N_{in} = 171{,}937$ inner edges, $N_\Sigma = 7421$ intersection edges, $N_m = 3801$ master hydraulic head unknowns, and $N_s = 3837$ slave hydraulic head unknowns. The sum of all the fluxes on intersections gives -7.43×10^{-13} m^3.s^{-1}. The sum of fluxes over the system is equal to -2.88×10^{-12} m^3.s^{-1}. The mean hydraulic head looks similar in the two Figures 5.5 and 5.7 and the equivalent permeability is equal to 0.771 m.s^{-1}. Notice that the refinement procedure here is not optimized as it would have been with a posteriori estimators, where the refined fractures would govern the flow. Combining with a posteriori estimators, this method is very promising to get an accurate solution of good quality while reducing the number of unknowns and thus the computational cost.

5.4. Analysis of convergence. The numerical convergence is estimated via a discrete relative L^2 error [21]. A computation is performed on a fine mesh $\mathcal{T}_\eta$ that gives a reference mean pressure P_η in the sense that it is assumed to approximate well the solution. Simulations are performed in coarsened grids $\mathcal{T}_h$ of mesh step $h > \eta$. The
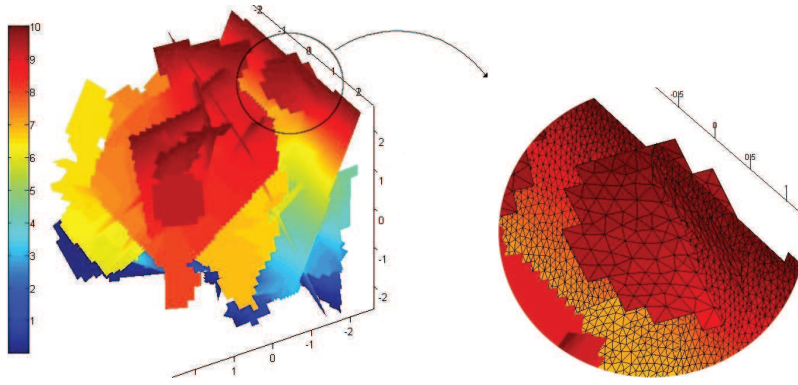
FIG. 5.7. *Example of DFN, test case 2: left: mean hydraulic head; right: zoom on the meshing.*

TABLE 5.3
Random distributions of parameters for the DFNs generated for the analysis of convergence.

Parameter	Random distribution
length	power law
shape	disks
position	uniform
orientation	uniform

TABLE 5.4
Values of the input parameters for the DFNs generated for the analysis of convergence.

| Parameter	Value
a	3.5
$L/l_{\min}$	2
N_{MC}	25
Mesh step	from 0.05 to 0.09
Density	100

mean pressure obtained on that mesh, P_h, is then compared with P_η in the following way:

$$(5.1) \qquad ||P_h - P_\eta||^2_{L^2(\Omega)} = \frac{\sum_{K \in \mathcal{T}_\eta} (\Pi_\eta P_h - P_\eta)^2 |K|}{\sum_{K \in \mathcal{T}_\eta} (P_\eta)^2 |K|},$$

where $|K|$ denotes the area of the triangle $K \in \mathcal{T}_\eta$ and $\Pi_\eta P_h$ denotes the projection of P_h onto the fine mesh $\mathcal{T}_\eta$.

We perform Monte Carlo simulations over 25 different randomly generated networks. Random distributions and parameters used for the simulations are given in Tables 5.3 and 5.4. For each Monte Carlo simulation, the mesh step is the same in each fracture. (Density of fractures is set by a percentage above the percolation threshold.)

Five mesh steps have been compared: 0.09, 0.08, 0.07, 0.06, and 0.05. The simulation at mesh step 0.05 is taken as a reference. Figure 5.8 shows that the method converges with mesh refinement. In this figure, each dot is the mean of the criterium value over the 25 networks.
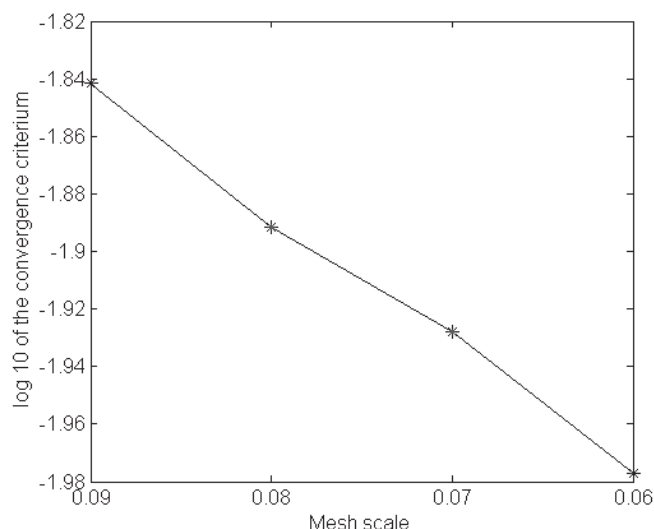
FIG. 5.8. *Log*10 *of the convergence criterium vs. mesh scale.*

6. Conclusion. Our paper presents a Mortar method within a mixed hybrid finite element method dedicated to solve flow in random DFNs. Intricate intersection configurations of fractures make the meshing procedure a complex and nonflexible task. Here we propose a flexible mesh generation while handling the difficulties by adapting the finite element method. More precisely, fractures are first meshed independently, ensuring a mesh of good quality by a three-step procedure. Second, a Mortar method is used to ensure the continuity of fluxes and heads at the fracture intersections. This method is designed in order to deal with discretized intersections involving more than two fractures. Numerical experiments show the efficiency of the method. It can handle most if not all randomly generated DFNs with complex 3D geometrical configurations. Thus our method leads to a promising tool for reducing the number of edges without loosing accuracy.

A forthcoming step is to define a posteriori error estimators in order to mesh finely only the fractures accounting for most of the flow. These estimators will then be used for generating an adaptative mesh with refining and coarsening. Another investigation line is to optimize the solving process by designing a specific linear solver using the structure of the matrix. Also, it will be interesting to work with a variable aperture within the fracture plane, yielding a heterogeneous transmissivity tensor. The objective is to run simulations and to study the channelling effects.

REFERENCES

[1] J. ERHEL, J.-R. DE DREUZY, AND B. POIRRIEZ, *Flow simulation in three-dimensional discrete fracture network*, SIAM J. Sci. Comput., 31 (2009), pp. 2688–2705.
[2] G. PICHOT, J. ERHEL, AND J.R. DE DREUZY, *A mixed-hybrid mortar method for solving flow in discrete fracture networks*, Appl. Anal., 89 (2010), pp. 1629–1643.
[3] P. A. HSIEH, *Scale effects in fluid flow through fractured geological media*, in Scale Dependence and Scale Invariance in Hydrology, Cambridge University Press, London, 1998, pp. 335–353,
[4] S. P. NEUMAN, *Trends, prospects and challenges in quantifying flow and transport through fractured rocks*, Hydrogeology J., 13 (2005), pp. 124–147.

[5] J. R. DE DREUZY, P. DAVY, AND O. BOUR, *Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 2-Permeability of networks based on log-normal distribution of apertures*, Water Resources Res., 37 (2001), pp. 2079–2095.

[6] J. R. DE DREUZY, P. DAVY, AND O. BOUR, *Permeability of 2D fracture networks with power-law distributions of length and aperture*, Water Resources Res., 38 (2002).

[7] A. W. NORDQVIST, Y. W. TSANG, C.-F. TSANG, B. DVERSTORP, AND J. ANDERSSON, *Effects of high variance of fracture transmissivity on transport and sorption at different scales in a discrete model for fractured rocks*, J. Contaminant Hydrology, 22 (1996), pp. 39–66.

[8] A. BAGHBANAN AND L. R. JING, *Hydraulic properties of fractured rock masses with correlated fracture length and aperture*, Internat. J. Rock Mech. Mining Sci., 44 (2007), pp. 704–719.

[9] J. C. S. LONG, J. S. REMER, C. R. WILSON, AND P. A. WITHERSPOON, *Porous media equivalents for networks of discontinuous fractures*, Water Resources Res., 18 (1982), pp. 645–658.

[10] J. C. S. LONG AND D. BILLAUX, *From field data to fracture network modeling: An example incorporating spatial structure*, Water Resources Res., 23 (1987), pp. 1201–1216.

[11] L. MARTINEZ-LANDA AND J. CARRERA, *A methodology to interpret cross-hole tests in a granite block*, J. Hydrology, 325 (2006), pp. 222–240.

[12] M. C. CACAS, E. LEDOUX, G. D. MARSILY, A. BARBEAU, P. CALMELS, B. GAILLARD, AND R. MAGRITTA, *Modeling fracture flow with a stochastic discrete fracture network: Calibration and validation. 1. The flow model*, Water Resources Res., 26 (1990).

[13] V. CVETKOVIC, S. PAINTER, N. OUTTERS, AND J. O. SELROOS, *Stochastic simulation of radionuclide migration in discretely fractured rock near the Aspo Hard Rock Laboratory*, Water Resources Res., 40 (2004).

[14] W. S. DERSHOWITZ AND C. FIDELIBUS, *Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method*, Water Resources Res., 35 (1999), pp. 2685–2691.

[15] M. VOHRALIK, *Méthodes numériques pour des équations elliptiques et paraboliques non linéaires, Application à des problèmes d'écoulement en milieux poreux et fracturés*, Ph.D. thesis, Paris IX Orsay University and Czech Technical University, Prague, 2004.

[16] T. KALBACHER, R. METTIER, C. MCDERMOTT, W. WANG, G. KOSAKOWSKI, T. TANIGUCHI, AND O. KOLDITZ, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the Grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[17] J. MARYSKA, O. SEVERYN, AND M. VOHRALIK, *Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model*, Comput. Geosci., 8 (2004), pp. 217–234.

[18] G. PICHOT, J.-R. DE DREUZY, J. ERHEL, AND P. DAVY, *Flow in multi-scale fracture networks: Numerical optimization by use of a Mortar-like method*, in Proceedings of the 2nd International Conference on Approximation Methods and Numerical Modelling in Environment and Natural Resources, 2009, pp. 761–766.

[19] E. BONNET, O. BOUR, N. ODLING, P. DAVY, I. MAIN, P. COWIE, AND B. BERKOWITZ, *Scaling of fracture systems in geological media*, Rev. Geophys., 39 (2001), pp. 347–383.

[20] H. HOTEIT, J. ERHEL, R. MOSÉ, B. PHILIPPE, AND P. ACKERER, *Numerical reliability for mixed methods applied to flow problems in porous media*, Comput. Geosci., 6 (2002), pp. 161–194.

[21] V. MARTIN, J. JAFFRE, AND J. E. ROBERTS, *Modeling fractures and barriers as interfaces for flow in porous media*, SIAM J. Sci. Comput., 26 (2005), pp. 1667–1691.

[22] P. DAVY, R. LE GOC, C. DARCEL, O. BOUR, J.-R. DE DREUZY, AND R. MUNIER, *A likely universal model of fracture scaling and its consequence for crustal hydromechanics*, J. Geophys. Res., 115 (2010).

[23] O. BOUR, P. DAVY, C. DARCEL, AND N. ODLING, *A statistical scaling model for fracture network geometry, with validation on a multiscale mapping of a joint network (Hornelen Basin, Norway)*, J. Geophys. Res., 107 (2002).