POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

Dottorato in Matematica per le Scienze dell'Ingegneria - XXVIII Ciclo

Thesis submitted for the degree of Doctor of Philosophy

Applications of the Virtual Element Method to Discrete Fracture Networks

Matías Fernando Benedetto

Supervisor
Prof. Stefano Berrone

Coordinator
Prof. Lamberto Rondoni

March 2016

POLITECNICO DI TORINO

Abstract

DISMA - Dipartimento di Scienze Matematiche

Doctor of Philosophy

Applications of the Virtual Element Method to Discrete Fracture Networks

by Matías Fernando Benedetto

We put forward in this work several novel applications of the Virtual Element Method in the context of Discrete Fracture Networks. A family of methods is presented here for solving Darcy flow, time dependent-problems and the complete transport equation in both diffusion-dominated and convection-dominated problems. We present as well an implementation of mixed Virtual Elements in the context of Discrete Fracture Networks.

Acknowledgements

First and foremost, I would like to express my gratitude to the authorities in charge of the Erasmus Mundus Action II programme, and in particular those in charge of the ARCOIRIS project, who with their complete financial support have made it possible for me to enroll and participate in this PhD programme.

I am also indebted to my advisor, Prof. Stefano Berrone, who has had the patience and the knowledge to help and guide me through my studies and has always been very supportive and available. I am also thankful to all the colleagues of the research group Sandra Pieraccini, Stefano Scialò, Andrea Borio and Fabio Vicini, for their input and the fruitful discussions from which I have profited a lot. From the first day, I always felt very comfortable and pleased about the hours that we shared working together. I am also grateful to the staff and people of the Politecnico di Torino and in particular the Dipartimento di Matematica, for being so welcoming and providing a wonderful place to study.

Finally, I would like to dedicate this work to my girlfriend, my family and my friends for their constant support and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Subsurface fluid flow has applications in a wide range of fields, including e.g. oil/gas recovery, gas storage, pollutant percolation and water resources monitoring among others [65] [34]. Underground fluid flow is a complex heterogeneous multi-scale phenomenon that involves complicated geological configurations. The medium is approximated by a Discrete Fracture Network (DFN): fractures are assumed to have a negligible thickness with respect to the other dimensions, and are represented as planar polygons intersecting each other in three dimensional space, with an equivalent bidimensional conductivity obtained by averaging the tridimensional one along the negligible dimension. This setting has been widely studied both from the modeling [1, 29, 35, 38, 62, 63] and from the computational point of view [20–22, 39, 45, 62, 63, 66, 67], and it finds its application in the field of evaluation of the properties of a fractured soil, where uncertainty quantification analysis [19, 59] is used to obtain information about the soil, thus requiring a large amount of simulations of the hydraulic head distribution. Typically, a DFN is obtained stochastically using probabilistic data to determine a distribution of orientation, density, size, aspect ratio, aperture and hydrological properties of the fractures [36], and it is a viable alternative to conventional continuum models in sparse fracture networks.

DFN simulations are very demanding from a computational point of view and due to the uncertainty of the statistical data, a great number of numerical simulations is required. Furthermore, the resolution of each configuration requires vast computational effort, increasing greatly with problem size.

In this work, we begin by focusing on the resolution of the steady-state flow in large fracture networks. The quantity of interest is the hydraulic head in the whole

network, which is the sum of pressure and elevation and is evaluated by means of Darcy's law. Afterwards we explore the possibilities of solving more complicated problems. We consider impervious rock matrix and fluid can only flow through fractures and traces (intersections of fractures), but no longitudinal flow along the traces is allowed. We introduce a transmissivity tensor for each fracture that depends on its aperture and its resistance to flow. The hydraulic head is a continuous function, but with discontinuous derivatives across the traces, which act as sources/sinks of flow. Matching conditions need to be added in order to preserve continuity along traces and flux balance at fracture intersections. More complex models for the flow in the fractures can be found in the literature [57]. Other important issues concern the coupling between the DFN and the surrounding rock matrix [39, 48].

The classical approach to DFN simulations consists in a finite element discretization of the network and in the resolution of the resulting algebraic linear system. With this approach, a great numerical obstacle to overcome is the need to provide on each fracture a good quality mesh conforming not only to the traces within the fracture, but also conforming to the other meshes on fractures sharing a trace.

Geometrical complexity is the greatest challenge when dealing with DFN-based simulations. Since the fracture generation has a random component, many complex situations arise that render the meshing process very complicated and sometimes impossible, e.g. very small angles, very close and almost parallel traces, high disparity of traces lengths, etc. In order to use traditional finite elements, fracture grids have to match in all the intersections between fractures, since these are discontinuity interfaces for the first order derivatives of the solution. All the aforementioned geometrical configurations complicate the meshing process and are the biggest obstacle in the discretization of the problem because it becomes very computationally demanding to obtain a good mesh from such a badly predisposed geometry. Furthermore, the meshing procedure depends on the whole DFN and is not independent for each fracture. In large realistic systems, which can count thousands, or even millions, of fractures, this mesh conformity constraints might lead to the introduction of a very large number of elements, independently of the accuracy required on the solution and possibly leading to over solving, if we consider the level of accuracy of the physical model.

Therefore, the main problem to be addressed is the geometrical treatment of the domain, in particular when the global or local conformity of the mesh is required [44]. Strategies are proposed in literature to ease the process of mesh generation and resolution for DFNs of large size. Some authors, see e.g. [47, 71], propose a simplification of DFN geometry to better handle the meshing procedure. In other cases, dimensional reduction is explored as in [29] and [35], where a system of 1D pipes that connect traces with

fractures has been used to simplify the problem. Mortar methods are used to relax the conformity condition with fracture meshes, that are only required to be aligned along the traces (see [66] and [67]). In [52], a BEM (Boundary Element Method) was applied that aims to minimize core memory usage by defining and storing only a relation between nodal fluxes and hydraulic head on traces for each fracture. The problem of obtaining a good globally conforming mesh is the subject of ongoing research. In [54], an adaptive mesh refinement method is described that aims for a high resolving mesh. In some works [45, 60] this difficulty is overcome by modifying the nature of the DFN, notably reducing the number of small angles between the intersections of fractures, which are responsible for the generation of degenerate polygons. This approach, though, changes the global statistical properties of the DFN; this is an issue, in the framework of stochastical analysis. In [58], a mixed formulation and a mesh modifying procedure was used to solve DFNs and reducing the number of elements for each fracture. Another mixed formulation was used in [37], where local corrections of traces are applied in order to obtain a globally conforming mesh. A novel approach was proposed in [20], [21], [22] and [23] in which the problem was reformulated as a PDE-constrained optimization. The minimization of a properly defined functional is adopted to enforce hydraulic head continuity and flux conservation at fracture intersections. Traditional finite elements (FEM) as well as extended finite elements (XFEM) were implemented to solve the problem.

In this work we study applications of the Virtual Element Method (VEM) to our problem. This is a very recent method whose origins are in the Mimetic Differences framework, but has been re introduced as a generalization of standard Finite Elements to meshes of arbitrary polygons. We explain the main characteristics of the method, its advantages, and how to exploit them for our purposes. The most important profit obtained form using Virtual Elements is without a doubt the possibility of fracture independent meshing.

The work is structured as follows: in chapter 2, we describe the first of our proposed techniques, which consist of a minimization of a certain quadratic functional that arises when we introduce a control variable representing the value of the flux along traces. Following that, in chapter 3 we suggest a globally conforming, as in many previous works, while still retaining locally independent meshing. It can be regarded as a case of domain decomposition. After that, we introduce in chapter 4 a Hybrid Mortar method that weakly imposes continuity of the hydraulic head on traces by introducing a base of Lagrange multipliers. In chapter 5 we set out to solve more complex problems on DFNs using the techniques established in the previous chapters and we focus mainly on solving the transport equation. Finally, chapter 6 presents an application of the mixed

formulation for the VEM and we study its use for solving problems on DFNs. A final conclusion and comparison of results is reserved for the final chapter of this thesis.

Chapter 2

An optimization approach

The contents of this chapter follow closely those found in the article "The virtual element method for discrete fracture network simulations" published in *Computer Methods in Applied Mechanics and Engineering* , Volume 280, 1 October 2014, Pages 135–156.

2.1 Introduction

In the recent paper [20] and follow up works [21] and [22], the problem of flow in a DFN is retooled as a PDE constrained optimization problem. The approach proposed in these works completely drops the need for any kind of mesh conformity, regardless of trace number and disposition; this goal is attained via the minimization of a given quadratic functional, allowing to obtain the solution for any given mesh. In this framework, any mesh independently generated on each fracture can be used. Since the solution may display a non-smooth behaviour along traces (namely, discontinuous normal derivatives), FEM on meshes not conforming to traces would result in poor solutions in a neighborhood of the traces.

The XFEM (eXtended Finite Element Method) is used in order to improve the solution near traces through the introduction of additional non-smooth basis functions, customized for the problem under consideration. The handling of these basis functions requires special care in numerical integration, and might be source of ill-conditioning due to the possible introduction of almost linearly dependent basis function (see, e.g., [41] and the references therein). In the present work the newly conceived Virtual Element Method is in charge for the space discretization on each fracture. Taking advantage from the great flexibility of VEM in allowing the use of rather general polygonal mesh elements, the aforementioned complexities related to XFEM enrichment functions can

be avoided. Indeed, a suitable mesh for representing the solution can be easily obtained starting from an arbitrary triangular mesh independently built on each fracture, and independent of the trace disposition. Then, whenever a trace crosses a mesh element, this can be split in two sub-elements obtaining a partial conformity.

All the steps needed for the use of the VEM in conjunction with the optimization approach for DFNs simulations are inherently fracture oriented, and can be executed in parallel. Numerical tests show that this approach leads to an efficient and reliable method.

We remark that the polygonal mesh obtained for VEM discretization naturally paves the way also for the use of a domain decomposition method or a Mortar approach. These possibilities are explored in chapters 3 and 4. Nevertheless, our main target here is to assess the viability of the optimization approach in conjunction with the VEM. Furthermore, within the optimization method, mixing of different discretization strategies (standard finite elements on meshes not necessarily conforming to traces, extended finite elements and virtual elements of different orders) remains possible, thus improving the flexibility to deal with any possible DFN configurations.

The present chapter is organized as follows: a description of the general problem is provided in section 2.2, followed by a brief introduction to the application of virtual element method to the problem at hand in section 2.3. Formulation and resolution of the discrete problem are sketched in section 2.4. Some technical issues concerning VEM implementation in this context as well as numerical results are given in section 2.5. We end with some conclusions in section 2.6.

2.2 Problem description

In this section we briefly sketch the main ideas of the PDE optimization method for discrete fracture network simulations introduced in [20–22].

Let us denote by Ω the DFN, composed by the union of planar open polygons F_i, with $i = 1, \dots, I$, resembling the fractures in the network. Let us denote by ∂F_i the boundary of F_i and by $\partial\Omega$ the set of all the fracture boundaries, $\partial\Omega = \cup_{i=1}^{I} \partial F_i$. We decompose $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$, $\Gamma_D \neq \emptyset$ being Γ_D the Dirichlet boundary and Γ_N the Neumann boundary. The boundary of each fracture is divided into a Dirichlet part $\Gamma_{iD} = \Gamma_D \cap \partial F_i$ and a Neumann part $\Gamma_{iN} = \Gamma_N \cap \partial F_i$, hence $\partial F_i = \Gamma_{iD} \cup \Gamma_{iN}$, with $\Gamma_{iD} \cap \Gamma_{iN} = \emptyset$. An empty Dirichlet boundary, $\Gamma_{iD} = \emptyset$ is allowed on fractures such that $\partial F_i \cap \Gamma_D = \emptyset$. Functions $H_i^D \in \mathrm{H}^{\frac{1}{2}}(\Gamma_{iD})$ and $G_i^N \in \mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN})$ are given and prescribe Dirichlet and Neumann boundary conditions, respectively, on the

boundary ∂F_i of each fracture. Intersections between fractures are called traces and are denoted by S_m, $m = 1, \dots, M$, while $\mathcal{S}$ denotes the set of all the traces of the system, and $\mathcal{S}_i$, for $i = 1, \dots, I$, denotes the subset of $\mathcal{S}$ corresponding to the M_i traces belonging to F_i. Each S_m uniquely identifies two indices $I_{S_m} = \{i, j\}$, such that $S_m \subseteq \bar{F}_i \cap \bar{F}_j$. Finally J_i collects all the indices $\{j\}$ relative to the fractures F_j intersected by F_i, i.e. $j \in J_i \iff \bar{F}_j \cap \bar{F}_i \neq \emptyset$.

The quantity of interest is the hydraulic head H that can be evaluated in Ω by means of the Darcy law. This originates a system of equations on the fractures defined as follows. Let us introduce for each fracture the following functional spaces:

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

and

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

and let us denote by H_i the restriction of H on F_i. Furthermore, let $\mathbf{K}_i$ denote a symmetric and uniformly positive definite tensor representing the fracture transmissivity. Without loss of generality and for the sake of simplicity, we assume that all traces are disjoint; this is not a restricting assumption as noted in [20]. Then H_i satisfies, for $i = 1, \dots, I$, the following problem: find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\begin{aligned} \int_{F_i} \mathbf{K}_i \nabla H_i \nabla v \mathrm{d}\Omega \quad &= \quad \int_{F_i} q_i v \mathrm{d}\Omega + \langle G_i^N, v_{|\Gamma_{iN}} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} \\ &\quad + \sum_{S \in \mathcal{S}_i} \langle \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(S), \mathrm{H}^{\frac{1}{2}}(S)}, \end{aligned} \tag{2.2.1}$$

where $q_i \in \mathrm{L}^2(F_i)$ denotes a source term on F_i and the symbol $\frac{\partial H_i}{\partial \hat{\nu}^i}$ represents the outward co-normal derivative of the hydraulic head:

$$\frac{\partial H_i}{\partial \hat{\nu}^i} = \hat{n}_i^T \mathbf{K}_i \nabla H_i,$$

with $\hat{n}_i$ outward normal to the boundary Γ_{iN}, and $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S$ denotes the jump of the co-normal derivative along the unique normal $\hat{n}_S^i$ fixed for the trace S on F_i, and represents the flux incoming into the fracture F_i through the trace S. The equations (2.2.1) for $i = 1, ..., I$ are coupled with the following matching conditions, ensuring hydraulic head continuity and flux balance across the traces:

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i, j \in I_{S_m}, \ \forall m = 1, \dots, M, \tag{2.2.2}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_{S_m}^i} \right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}_{S_m}^j} \right]\!\right]_{S_m} = 0, \qquad \text{for } i, j \in I_{S_m}. \tag{2.2.3}$$

The simultaneous resolution of equations (2.2.1)-(2.2.3) might result infeasible for practical applications, as previously discussed. In contrast, the approach developed in [20–22] only requires the resolution of local problems on each fracture independently, resorting to an optimization approach to enforce matching at the intersections. In order to describe this strategy, let us introduce for each trace in each fracture the control variables $U_i^S \in \mathcal{U}^S = \mathrm{H}^{-\frac{1}{2}}(S)$, defined as $U_i^S = \alpha H_{i|_S} + \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i}\right]\!\right]_S$, where α is a fixed positive parameter, and the quadratic functional

$$\begin{aligned} J(H,U) &= \sum_{m=1}^{M} \left(\left\| H_{i|_{S_m}} - H_{j|_{S_m}} \right\|^2_{\mathrm{H}^{\frac{1}{2}}(S)} \right. \\ &+ \left. \left\| U_i^{S_m} + U_j^{S_m} - \alpha \left(H_{i|_{S_m}} + H_{j|_{S_m}} \right) \right\|^2_{\mathrm{H}^{-\frac{1}{2}}(S)} \right). \end{aligned} \tag{2.2.4}$$

Equations (2.2.1), prescribed on the fractures, are equivalently restated as:

$$\begin{aligned} &\int_{F_i} \mathbf{K}_i \nabla H_i \nabla v \mathrm{d}\Omega + \alpha \sum_{S \in \mathcal{S}_i} \int_S H_{i|_S} v_{|_S} \mathrm{d}\Gamma = \\ &\int_{F_i} q_i v \mathrm{d}\Omega + \langle G_i^N, v_{|_{\Gamma_{iN}}} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} + \textstyle\sum_{S \in \mathcal{S}_i} \langle U_i^S, v_{|_S} \rangle_{\mathcal{U}^S, \mathcal{U}^{S\prime}}. \end{aligned} \tag{2.2.5}$$

Let us define $\mathcal{U}^{\mathcal{S}_i} = \mathrm{H}^{-\frac{1}{2}}(\mathcal{S}_i)$ and let $\mathcal{R}_i$ denote an operator providing lifting of the Dirichlet boundary conditions on Γ_{iD}, if not empty. We then introduce the following linear bounded operators:

$$\begin{aligned} A_i &\in \mathcal{L}(V_i, V_i'), \quad \langle A_i w, v \rangle_{V_i', V_i} = (\mathbf{K}_i \nabla w, \nabla v) + \alpha \left(w_{|_{\mathcal{S}_i}}, v_{|_{\mathcal{S}_i}} \right)_{\mathcal{S}_i}, \\ B_i^S &\in \mathcal{L}(\mathcal{U}^S, V_i'), \quad \langle B_i^S U_i^S, v \rangle_{V_i', V_i} = \langle U_i^S, v_{|_S} \rangle_{\mathcal{U}^S, \mathcal{U}^{S\prime}}, \\ B_i &= \prod_{S \in \mathcal{S}_i} B_i^S \in \mathcal{L}(\mathcal{U}^{\mathcal{S}_i}, V_i'), \quad \langle B_i U_i, v \rangle_{V_i', V_i} = \langle U_i, v_{|_{\mathcal{S}_i}} \rangle_{\mathcal{U}^{\mathcal{S}_i}, \mathcal{U}^{\mathcal{S}_i\prime}}, \end{aligned}$$

with $w, v \in V_i$, and $U_i \in \mathcal{U}^{\mathcal{S}_i}$ is the tuple of control variables U_i^S for $S \in \mathcal{S}_i$. Analogously, $U \in \mathcal{U}^{\mathcal{S}}$ denotes the tuple of control variables U_i for $i = 1, ..., I$. The dual operator of A_i is denoted by A_i^* and B_i^* denotes the dual of B_i. The operator $B_{iN} \in \mathcal{L}(\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), V_i')$ imposing Neumann boundary conditions is defined such that

$$\langle B_{iN} G_i^N, v \rangle_{V_i', V_i} = \langle G_i^N, v_{|_{\Gamma_{iN}}} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})}.$$

According to this functional setting and definitions, problems (2.2.5) are restated as: $\forall i = 1, ..., I$, find $H_i \in V_i^D$, with $H_i = H_i^0 + \mathcal{R}_i H_i^D$ and $H_i^0 \in V_i$, such that

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_i^N - A_i^D \mathcal{R}_i H_i^D, \quad \text{in } F_i, \tag{2.2.6}$$

where A_i^D is an operator defined similarly to A_i, but operating on elements in $\mathrm{H}^1(F_i)$. We remark that, if $\alpha > 0$, for a given U_i, the solution H_i to (2.2.6) exists and is unique for a non isolated fracture even if we set Neumann boundary conditions on the whole ∂F_i.

Following the arguments proposed in [22], it can be shown that the unique minimum of functional (2.2.4) is obtained for values of H and of the control functions U that correspond to the fulfilment of conditions (2.2.2) and (2.2.3) on the traces. In other words, the solution of the problem

$$\min J \quad \text{subject to (2.2.6)} \tag{2.2.7}$$

corresponds to the solution of the coupled system of equations (2.2.1)-(2.2.3).

As shown in previous works (see e.g. [22]) this optimization problem can be tackled with a gradient based method. Even if different approaches could also be employed, gradient-based methods are particularly appealing since they allow to independently solve problems on fractures and can be straightforwardly plugged in a parallel resolution process.

In the continuous setting, the gradient based method is formally devised on the following considerations: the optimal $U \in \mathcal{U}$, solution to (2.2.7), satisfies the following system of equations, corresponding to the Fréchet derivatives of J with respect to the control variables: $\forall i = 1, \dots, I$

$$B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \left(U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right) - \alpha \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) = 0, \tag{2.2.8}$$

where the operators $C_i^S = B_i{}^*$ are restriction operators on the traces, $\Lambda_{\mathcal{U}^{\mathcal{S}_i}} : \mathcal{U}^{\mathcal{S}_i} \to \mathcal{U}^{\mathcal{S}_i\prime}$ is the Riesz isomorphism, and functions $P_i \in V_i$ are the solution to

$$\begin{aligned} A_i^* P_i \;=\; & C_i{}^* \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} \left[\prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right) \right. \\ & \left. + \alpha^2 \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) \right] - \alpha C_i{}^* \left(U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right), \qquad \text{in } F_i \end{aligned} \tag{2.2.9}$$

with homogeneous Neumann and Dirichlet boundary conditions. Then, we can set $\forall i = 1, \dots, I$

$$\nabla J(U_i) = B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S \in \mathcal{S}_i} \left(U_i^S + U_j^S - \alpha \Lambda_{\mathcal{U}^S}^{-1} (C_i^S H_i(U_i) + C_j^S H_j(U_j)) \right), \tag{2.2.10}$$

and

$$\nabla J(U) = \prod_{i=1}^{I} \nabla J(U_i). \tag{2.2.11}$$

The gradient based algorithm for solving (2.2.7) is fully described in [22]. Here, we focus on a first-discretize-then-optimize approach, and we move on by introducing, in the next section, the space discretization.

2.3 The Virtual Element Method

The Virtual Element Method [2, 10, 11, 26] is a very recent technique for solving partial differential equations on meshes of fairly general polygonal elements with an arbitrary number of sides. This characteristic is very attractive for the application considered herein. Indeed, on each fracture we solve equation (2.2.6), whose solution can have a discontinuous gradient across the traces. In order to correctly reproduce this irregular behaviour, we can take advantage of the flexibility of virtual elements by transforming, on each fracture, a given triangulation (non conforming to traces) in a more general mesh, conforming to traces, simply obtained by splitting the triangles along traces into more general sub-polygons not crossed by traces. We remark that we do not require conformity between the meshes of the two fractures intersecting at a trace. As a consequence of the meshing process, a partial conformity (i.e. conformity to traces but no conformity between the meshes of intersecting fractures) will result, but the meshing process is still independent on each fracture and thus easy and reliable (see Figure 2.3.1).

Let us now describe the application of the VEM to the problem considered. For the sake of simplicity, we consider in this section homogeneous conditions on the Dirichlet boundary; furthermore, we consider in this work the case of virtual elements of order $k = 1$ and we assume that the fracture transmissivity $\mathbf{K}_i$ is constant on each fracture, but might vary from one fracture to another. We will focus on a generic fracture $F_i \subset \Omega$, since the process is independent on each fracture. Let $\{\mathcal{T}_{i,\delta}\}_\delta$ be a family of meshes on

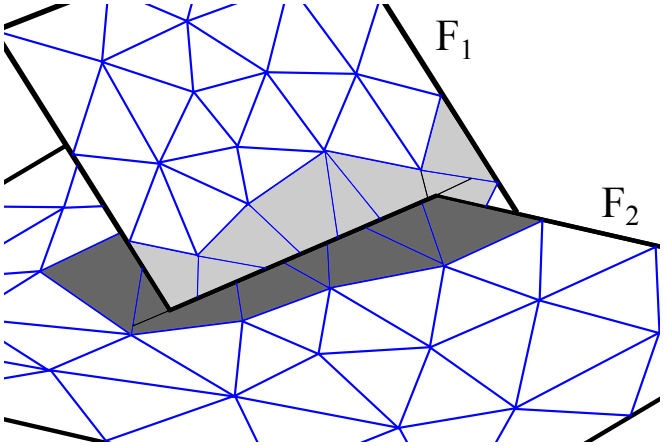


FIGURE 2.3.1: Example of the mesh for the VEM: elements shaded have been cut into polygons to match the trace on the two fractures independently

F_i, being δ the mesh parameter (corresponding to the square root of the largest element size). Each mesh is built as previously sketched: we start with a given triangulation, and whenever a trace crosses an element, the latter is split by the trace itself in two sub-polygons. If the trace ends inside an element, it is prolonged up to the boundary of the element. To note is that we obtain convex polygons, thus satisfying the assumptions in [10]. Each $\mathcal{T}_{i,\delta}$ is therefore made of open polygons $\{E\}$ with an arbitrary number n_E of edges e, and we call N_i the total number of vertices. We define for each δ a space $V_{i,\delta} \subset \mathrm{H}^1(F_i)$ as follows. Following the notation in [10], for a generic element E of the mesh, let us introduce the space

$$\mathbb{B}_1(\partial E) = \left\{ v \in \mathrm{C}^0(\partial E) : v_{|e} \in \mathbb{P}_1(e),\ \forall e \subset \partial E \right\}.$$

Let $V^{E,1}$ be the space of harmonic functions that are linear on the boundaries of the element,

$$V^{E,1} = \left\{ v \in \mathrm{H}^1(E) : v_{|\partial E} \in \mathbb{B}_1(\partial E), \Delta v_{|E} = 0 \right\}.$$

We finally set

$$V_{i,\delta} = \left\{ v \in \mathrm{H}^1_0(F_i) : v_{|E} \in V^{E,1},\ \forall E \in \mathcal{T}_{i,\delta} \right\}.$$

For each element, functions in $V^{E,1}$ are uniquely identified by prescribing the polynomial functions on ∂E, or, equivalently, specifying the values at the n_E vertices of the polygon. With this natural choice for the degrees of freedom, the C^0 continuity of functions in $V_{i,\delta}$ is easily enforced. The dimension of $V_{i,\delta}$ is N_i, and we introduce a Lagrange basis $\{\phi_1, \ldots, \phi_{N_i}\}$, defined by $\phi_j(x_k) = \delta_{jk}$, where x_k is the k-th vertex in the mesh. Functions $\{\phi_j\}$ are in general not explicitly known inside the elements, but only on the boundaries of the elements, and this is a key point of VEM. Further we observe that the space of polynomials $\mathbb{P}_1(E) \subset V_{i,\delta_{|E}}$ for each element E in $\mathcal{T}_{i,\delta}$.

On the space $V_{i,\delta}$ we define a symmetric bilinear form $a_{i,\delta} : V_{i,\delta} \times V_{i,\delta} \mapsto \mathbb{R}$ as the discrete counterpart of the bilinear form $a_i : V_i \times V_i \mapsto \mathbb{R}$ defined as

$$a_i(H_i, v) = \langle A_i H_i, v \rangle_{V_i', V_i}.$$

On each element E we introduce the bilinear form $a^E_{i,\delta}(\cdot,\cdot) : V_{i,\delta_{|E}} \times V_{i,\delta_{|E}} \mapsto \mathbb{R}$:

$$a^E_{i,\delta}(\phi, \varphi) = (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla \mathcal{P}^E \varphi)_E + \alpha \left(\phi_{|_{\mathcal{S}_i \cap \partial E}}, \varphi_{|_{\mathcal{S}_i \cap \partial E}} \right)_{\mathcal{S}_i \cap \partial E} + S^E(\phi, \varphi), \quad (2.3.1)$$

and for any two functions $\phi,\ \varphi \in V_{i,\delta}$ we have

$$a_{i,\delta}(\phi, \varphi) = \sum_{E \in \mathcal{T}_{i,\delta}} a^E_{i,\delta}(\phi, \varphi). \quad (2.3.2)$$

In (2.3.1), the projection operator $\mathcal{P}^E : V_{i,\delta|E} \mapsto \mathbb{P}_1(E)$ is defined for any function $\phi \in V_{i,\delta|E}$ by

$$\begin{cases} (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla p)_E & = (\mathbf{K}_i \nabla \phi, \nabla p)_E \quad \forall p \in \mathbb{P}_1(E) \\ \sum_{k=1}^{n_E} \mathcal{P}^E \phi(\mathbf{x}_k) & = \sum_{k=1}^{n_E} \phi(\mathbf{x}_k) \end{cases} \tag{2.3.3}$$

being $\{\mathbf{x}_k\}_k$ the coordinates of the vertices of element E, and $S^E : V_{i,\delta|E} \times V_{i,\delta|E} \mapsto \mathbb{R}$ is a properly designed functional that is non-zero on the kernel of $\mathcal{P}^E$.

Remark 2.1. Let us observe that the definition (2.3.1) for the bilinear form and (2.3.3) for the projection operator slightly differ from the definitions introduced in [10]. In our definition of the discrete bilinear form the projection operator does not affect the portion of the operator defined on the traces, and consequently this term does not appear in (2.3.3) or in the definition of the stability operator S^E. According to [10], we assume that there exist two positive constants c_0 and c_1 independent from the mesh element E and of element diameter, such that:

$$c_0(\mathbf{K}_i \nabla\varphi, \nabla\varphi)_E \leq S^E(\varphi, \varphi) \leq c_1(\mathbf{K}_i \nabla\varphi, \nabla\varphi)_E, \quad \forall \varphi \in V_{i,\delta|E}, \text{ with } \mathcal{P}^{\mathrm{E}} \varphi = 0. \tag{2.3.4}$$

On each element E of the triangulation we have:

$$\begin{aligned} a_i^E(\phi, \varphi) \quad = \quad & a_i^E(\mathcal{P}^E \phi, \mathcal{P}^E \varphi) + a_i^E(\phi - \mathcal{P}^E \phi, \varphi - \mathcal{P}^E \varphi) \\ & + \alpha \left(\phi - \mathcal{P}^E \phi, \mathcal{P}^E \varphi\right)_{\mathcal{S}_i \cap \partial E} + \alpha \left(\varphi - \mathcal{P}^E \varphi, \mathcal{P}^E \phi\right)_{\mathcal{S}_i \cap \partial E} \end{aligned} \tag{2.3.5}$$

that replaces equation (4.22) of [10].

It is possible to show that the given definition of the bilinear form is consistent and stable. Consistency easily follows from definition (2.3.1) and from (2.3.3): for all $E \in \mathcal{T}_{i,\delta}$, $\forall p \in \mathbb{P}_1(E)$, $\forall \phi \in V_{i,\delta|E}$ we have:

$$\begin{aligned} a_{i,\delta}^E(\phi, p) \quad &= \quad \left(\mathbf{K}_i \nabla(\phi - \mathcal{P}^E \phi), \nabla p\right)_E + \left(\mathbf{K}_i \nabla(\mathcal{P}^E \phi), \nabla p\right)_E + \alpha \left(\phi, p\right)_{\mathcal{S}_i \cap \partial E} \\ &= \quad \left(\mathbf{K}_i \nabla(\mathcal{P}^E \phi), \nabla p\right)_E + \alpha \left(\phi, p\right)_{\mathcal{S}_i \cap \partial E} = a_i^E(\phi, p), \end{aligned}$$

being $a_i^E(\cdot, \cdot)$ the restriction to a mesh element of the continuous bilinear form. Stability can be proved similarly to [10], using (2.3.4) and (2.3.5).

Assuming basic quality properties for the triangulation, functional S^E can be chosen as in [10], thus satisfying conditions (2.3.4): for all $\phi, \varphi \in V_{i,\delta|E}$ we set

$$S^E(\phi, \varphi) = \sum_{k=1}^{n_E} \mathbf{K}_i(\phi(\mathbf{x}_k) - (\mathcal{P}^E \phi)(\mathbf{x}_k))(\varphi(\mathbf{x}_k) - (\mathcal{P}^E \varphi)(\mathbf{x}_k)). \tag{2.3.6}$$

Concerning the treatment of the source term q_i at right hand side of equation (2.2.6), it is shown in [11] that convergence rates are preserved by approximating q_i with a piecewise constant function on each element of the mesh.

Given the previous results and definitions, it is possible to use the convergence theorem in [10] to prove that the discrete problems on the fractures are well posed and convergence rates are equal to those of standard finite elements of the same order.

Even if functions in $V_{i,\delta}$ are only known on the edges of mesh elements, the knowledge of the degrees of freedom allows us to compute the discrete bilinear forms. In fact, in order to compute $\mathcal{P}^E \phi$, for any $\phi \in V_{i,\delta|E}$ and $p \in \mathbb{P}_1(E)$ we evaluate:

$$\begin{aligned}(\mathbf{K}_i \nabla\phi, \nabla p)_E &= \int_E \mathbf{K}_i \nabla\phi \nabla p \, \mathrm{d}E = \int_E \mathbf{K}_i \Delta p \, \phi \, \mathrm{d}E + \int_{\partial E} \mathbf{K}_i \frac{\partial p}{\partial n_{\partial E}} \phi \, \mathrm{d}\gamma \\ &= \int_{\partial E} \mathbf{K}_i \frac{\partial p}{\partial n_{\partial E}} \phi \, \mathrm{d}\gamma\end{aligned}$$

where $n_{\partial E}$ is the outward unit normal vector to ∂E.

2.4 Formulation and resolution of the discrete problem

As shown in section 2.2, the problem has been reformulated as a PDE-constrained optimization problem (see equation (2.2.7)) in which the quadratic functional J is to be minimized subject to linear constraints. In this section, following a first-discretize-then-optimize approach, we give some details about the discrete formulation of the problem and the numerical approach for computing a solution to the problem. In the following, we will use lower case letters for the finite dimensional approximations of functions H and U.

2.4.1 Discrete formulation

As outlined in the previous section, we introduce a finite dimensional basis for each fracture F_i, with a total number $N^F = \sum_{i=1}^{I} N_i$ of DOFs on the fractures. Concerning the functional space on the traces, in order to simplify the discussion, we consider the following different numbering for the control functions u_i^S, induced by the trace numbering. Being $S = S_m$ a given trace, with $I_{S_m} = \{i, j\}$ and assuming $i < j$, we denote by u_m^- and by u_m^+ the control functions related to the m-th trace and corresponding to fractures F_i and F_j, respectively. By overloading the notation, we use the same symbol for the corresponding vector of DOFs. Let us introduce basis functions $\psi_{m,k}^-$, $k = 1, ..., N_m^-$ and $\psi_{m,k}^+$, $k = 1, ..., N_m^+$ for the space of the control function u_m^- and

u_m^+, respectively. Note that here we allow to use different spaces on the two "sides" of each trace. Then we have, for $m = 1, ..., M$, $\star = -, +$, $u_m^\star = \sum_{k=1}^{N_m^\star} u_{m,k}^\star \psi_{m,k}^\star$. Setting $N^T = \sum_{m=1}^{M} (N_m^- + N_m^+)$, we define $u \in \mathbb{R}^{N^T}$ concatenating $u_1^-, u_1^+, \ldots, u_M^-, u_M^+$.

Let us consider the functional J, whose expression is given in section 2.2 by equation (2.2.4), and let us write the discrete functional in terms of L^2 norms instead of $\mathrm{H}^{-\frac{1}{2}}$ and $\mathrm{H}^{\frac{1}{2}}$ norms on the traces: its discrete counterpart is

$$\begin{aligned} J \;=\; & \frac{1}{2} \sum_{i=1}^{I} \sum_{S \in \mathcal{S}_i} \left(\int_S \Big(\sum_{k=1}^{N_i} h_{i,k} \varphi_{i,k_{|S}} - \sum_{k=1}^{N_j} h_{j,k} \varphi_{j,k_{|S}} \Big)^2 \mathrm{d}\gamma + \right. \\ & \left. \int_S \Big(\sum_{k=1}^{N_m^-} u_{m,k}^- \psi_{m,k}^- + \sum_{k=1}^{N_m^+} u_{m,k}^+ \psi_{m,k}^+ - \alpha \sum_{k=1}^{N_i} h_{i,k} \varphi_{i,k_{|S}} - \alpha \sum_{k=1}^{N_j} h_{j,k} \varphi_{j,k_{|S}} \Big)^2 \mathrm{d}\gamma \right). \end{aligned} \tag{2.4.1}$$

Let us define for all $S_m \in \mathcal{S}$, for $p, q \in I_{S_m}$ (possibly $p = q$), the matrices

$$(C_{p,q}^{S_m})_{k,\ell} = \int_{S_m} \varphi_{p,k_{|S_m}} \varphi_{q,\ell_{|S_m}} \, \mathrm{d}\gamma, \qquad C_{p,q} = \sum_{S_m \in \mathcal{S}_p} C_{p,q}^{S_m}.$$

Furthermore, for $m = 1, ..., M$ and $\star = -, +$ define $\mathcal{C}_m^\star \in \mathbb{R}^{N_m^\star \times N_m^\star}$, $\mathcal{C}_m^\pm \in \mathbb{R}^{N_m^- \times N_m^+}$ and $\mathcal{C}_m$ as:

$$(\mathcal{C}_m^\star)_{k\ell} = \int_{S_m} \psi_{m,k}^\star \psi_{m,\ell}^\star \, \mathrm{d}\gamma, \quad (\mathcal{C}_m^\pm)_{k\ell} = \int_{S_m} \psi_{m,k}^- \psi_{m,\ell}^+ \, \mathrm{d}\gamma, \quad \mathcal{C}_m = \begin{pmatrix} \mathcal{C}_m^- & \mathcal{C}_m^\pm \\ (\mathcal{C}_m^\pm)^T & \mathcal{C}_m^+ \end{pmatrix},$$

and $B_{i,m}^\star \in \mathbb{R}^{N_i \times N_m^\star}$ and $B_{j,m}^\star \in \mathbb{R}^{N_j \times N_m^\star}$ as

$$(B_{i,m}^\star)_{k\ell} = \int_{S_m} \psi_{m,k}^\star \varphi_{i,\ell_{|S_m}} \, \mathrm{d}\gamma, \qquad (B_{j,m}^\star)_{k\ell} = \int_{S_m} \psi_{m,k}^\star \varphi_{j,\ell_{|S_m}} \, \mathrm{d}\gamma.$$

The functional J in (2.4.1) is therefore written, in algebraic form, as

$$\begin{aligned} J(h,u) \;=\; & \frac{1}{2} \sum_{i=1}^{I} \sum_{S \in \mathcal{S}_i} (1+\alpha^2) h_i^T C_{i,i}^S h_i + (1+\alpha^2) h_j^T C_{j,j}^S h_j - 2(1-\alpha^2) h_i^T C_{i,j}^S h_j \\ & + (u_m^-)^T \mathcal{C}_m^- u_m^- + (u_m^+)^T \mathcal{C}_m^+ u_m^+ + 2(u_m^-)^T \mathcal{C}_m^\pm u_m^+ - \alpha (h_i^T B_{i,m}^+ u_m^+) \\ & - \alpha (h_i^T B_{i,m}^- u_m^-) - \alpha (h_j^T B_{j,m}^- u_m^-) - \alpha (h_j^T B_{j,m}^+ u_m^+) - \alpha ((u_m^-)^T (B_{i,m}^-)^T h_i) \\ & - \alpha ((u_m^+)^T (B_{i,m}^+)^T h_i) - \alpha ((u_m^-)^T (B_{j,m}^-)^T h_j) - \alpha ((u_m^+)^T (B_{j,m}^+)^T h_j). \end{aligned}$$

We now allow for a more compact form of $J(h,u)$ by assembling previous matrices as follows. We set

$$B_{i,m} = (B_{i,m}^- \; B_{i,m}^+) \in \mathbb{R}^{N_i \times (N_m^- + N_m^+)}, \qquad u_m = (u_m^-, u_m^+).$$

For each fixed $i = 1, ..., I$, matrices $B_{i,m}$, for m such that $S_m \in \mathcal{S}_i$, are then grouped row-wise to form the matrix $B_i \in \mathbb{R}^{N_i \times N_{\mathcal{S}_i}}$, with $N_{\mathcal{S}_i} = \sum_{S_m \in \mathcal{S}_i} (N_m^- + N_m^+)$. Matrix B_i acts on a column vector u_i obtained extracting blocks u_m, for $S_m \in \mathcal{S}_i$, from u and appending them in the same order used for $B_{i,m}$, as the action of a suitable operator $R_i : \mathbb{R}^{N^T} \mapsto \mathbb{R}^{N_{\mathcal{S}_i}}$ such that $u_i = R_i u$. Finally, let $B \in \mathbb{R}^{N^F \times N^T}$ be defined by

$$B = \begin{pmatrix} B_1 R_1 \\ \vdots \\ B_I R_I \end{pmatrix}.$$

Let now $G^h \in \mathbb{R}^{N^F \times N^F}$ be defined blockwise as follows: for $i = 1, ..., I$ we set

$$G^h_{ii} = (1 + \alpha^2) C_{i,i}, \qquad G^h_{ij} = (\alpha^2 - 1) C^S_{i,j} \text{ if } j \in J_i \text{ (0 elsewhere)},$$

where, fixed F_i, J_i collects the indices j such that $|\bar{F}_j \cap \bar{F}_i| > 0$. Since, obviously, $j \in J_i$ if and only if $i \in J_j$, and due to the straightforward property $(G^h_{ij})^T = G^h_{ji}$, we have that G^h is a symmetric matrix. Next, let us define the matrix $G^u \in \mathbb{R}^{N^T \times N^T}$ blockwise as $G^u = \text{diag}(\mathcal{C}_m, m = 1, ..., M)$. With these definitions at hand, the functional J is rewritten

$$J(h, u) := \frac{1}{2} \left(h^T G^h h - \alpha h^T B u - \alpha u^T B^T h + u^T G^u u \right)$$

being $h \in \mathbb{R}^{N^F}$ obtained appending vectors h_i, $i = 1, ..., I$.

We finally note that, setting

$$G = \begin{pmatrix} G^h & -\alpha B \\ -\alpha B^T & G^u \end{pmatrix}$$

and $w = (h, u)$, J can be simply written as $J = \frac{1}{2} w^T G w$, with G straightforwardly symmetric, due to previous considerations, and positive semidefinite by construction.

Constraints (2.2.6) are written as a unique linear system as follows: For all $i = 1, ..., I$ define the matrix $A_i \in \mathbb{R}^{N_i \times N_i}$ as

$$\begin{aligned} (A_i)_{k\ell} &= \sum_{E \in \mathcal{T}_{i,\delta}} \left(\int_{F_i} \mathbf{K}_i \nabla \mathcal{P}^E \phi_{i,k} \nabla \mathcal{P}^E \phi_{i,\ell} \, \mathrm{d}F_i + S^E(\phi_{i,k}, \phi_{i,\ell}) \right) \\ &+ \alpha \sum_{S \in \mathcal{S}_i} \int_S \phi_{i,k|_S} \phi_{i,\ell|_S} \, \mathrm{d}\gamma, \quad k, \ell = 1, \ldots, N_i \end{aligned}$$

where the operators $\mathcal{P}^E$ and S^E are defined by (2.3.3) and (2.3.6), respectively.

For each fracture F_i, we set $N^i_{\mathcal{S}_i} = \sum_{S_m \in \mathcal{S}_i} N^\star_m$ as the number of DOFs on traces of F_i on the F_i "side", and we define matrices $\mathcal{B}_i \in \mathbb{R}^{N_i \times N^i_{\mathcal{S}_i}}$ grouping row-wise matrices $B^\star_{i,m}$, with m spanning traces in $\mathcal{S}_i$, and setting for each m either $\star = +$ or $\star = -$ according to which one of the two "sides" of trace S_m is on F_i. Matrices $\mathcal{B}_i$ act on a column vector u'_i containing all the $N^i_{\mathcal{S}_i}$ control DOFs corresponding to the traces of F_i, obtained extracting blocks $u^\star_m$, for $S_m \in \mathcal{S}_i$, from u and appending them in the same order used in the definition of $\mathcal{B}_i$. Again, this can be obtained as the action of a suitable operator $R'_i : \mathbb{R}^{N^T} \mapsto \mathbb{R}^{N^i_{\mathcal{S}_i}}$ such that $u'_i = R'_i u$. In practice, R'_i extracts only sub-vectors $u^\star_m$ from u corresponding to control functions on the "correct side" of the trace.

The algebraic formulation of the primal equations (2.2.6) is then

$$A_i h_i = \tilde{q}_i + \mathcal{B}_i\, u'_i, \qquad i = 1, ..., I, \tag{2.4.2}$$

where $\tilde{q}_i$ accounts for the term q_i in (2.2.6) and for the boundary conditions on the fracture F_i.

We set $A = \mathrm{diag}(A_i,\ i = 1, ..., I) \in \mathbb{R}^{N^F \times N^F}$ and define $\mathcal{B} \in \mathbb{R}^{N^F \times N^T}$ as

$$\mathcal{B} = \begin{pmatrix} \mathcal{B}_1\, R'_1 \\ \vdots \\ \mathcal{B}_I\, R'_I \end{pmatrix}$$

Setting $q = (\tilde{q}_1, \ldots, \tilde{q}_I) \in \mathbb{R}^{N^F}$, constraints (2.4.2) are then written $Ah - \mathcal{B}\, u = q$.

The problem under consideration is therefore reformulated as the following equality constrained quadratic programming problem:

$$\min\ J(h,u) \quad = \quad \frac{1}{2}\left(h^T G^h h - \alpha h^T B u - \alpha u^T B^T h + u^T G^u u\right) \tag{2.4.3}$$

$$\text{s.t.} \qquad Ah - \mathcal{B}\, u = q. \tag{2.4.4}$$

2.4.2 Solving the optimization problem

The first order optimality conditions for problem (2.4.3)-(2.4.4) are the following:

$$\begin{pmatrix} G^h & -\alpha B & A^T \\ -\alpha B^T & G^u & -\mathcal{B}^T \\ A & -\mathcal{B} & 0 \end{pmatrix} \begin{pmatrix} h \\ u \\ -p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix} \tag{2.4.5}$$

being p the vector of Lagrange multipliers.

The previous saddle point problem is, for real applications, a very large scale problem, with highly sparse blocks, as A, G^u are block diagonal matrices, G^h, B and $\mathcal{B}$ are block-sparse.

By (formally) using the linear constraint for eliminating the unknown h as

$$h = A^{-1}(\mathcal{B}\,u + q), \tag{2.4.6}$$

we obtain the following equivalent unconstrained problem :

$$\begin{aligned} \min\ \hat{J}(u) \quad := \quad & \frac{1}{2}u^T(\mathcal{B}^T\,A^{-T}G^hA^{-1}\,\mathcal{B} + G^u - \alpha\,\mathcal{B}^T\,A^{-T}B - \alpha B^TA^{-1}\,\mathcal{B})u \\ & + q^TA^{-T}(G^hA^{-1}\,\mathcal{B} - \alpha B)u. \end{aligned}$$

For further convenience we rewrite $\hat{J}(u) = \frac{1}{2}u^T\hat{G}u + \hat{q}^Tu$. A gradient-based method for the minimization of the functional requires the computation of the gradient of $\hat{J}$:

$$\begin{aligned} \nabla\hat{J}(u) \quad = \quad & (\mathcal{B}^T\,A^{-T}G^hA^{-1}\,\mathcal{B} + G^u - \alpha(\mathcal{B}^T\,A^{-T}B + B^TA^{-1}\,\mathcal{B}))u + \\ & (\mathcal{B}^T\,A^{-T}G^h - \alpha B^T)A^{-1}q. \end{aligned}$$

or, equivalently, $\nabla\hat{J}(u) = \hat{G}u + \hat{q}$.

The gradient can be written in terms of some auxiliary variables as follows. Re-arranging previous expression, we obtain

$$\nabla\hat{J}(u) = \mathcal{B}^T\,A^{-T}G^hA^{-1}(\mathcal{B}\,u + q) + G^uu - \alpha\,\mathcal{B}^T\,A^{-T}Bu - \alpha B^TA^{-1}(\mathcal{B}\,u + q)$$

and recalling (2.4.6), one has

$$\nabla\hat{J}(u) = \mathcal{B}^T\,A^{-T}G^hh + G^uu - \alpha\,\mathcal{B}^T\,A^{-T}Bu - \alpha B^Th.$$

Now set $p := A^{-T}(G^hh - \alpha Bu)$, i.e. given h and u, p solves

$$A^Tp = G^hh - \alpha Bu. \tag{2.4.7}$$

With these definitions, we may write

$$\nabla\hat{J}(u) = \mathcal{B}^T\,p + G^uu - \alpha B^Th. \tag{2.4.8}$$

Note that setting to zero the previous expression for obtaining stationary points for $\hat{J}(u)$, and collecting such equation together with (2.4.6) and (2.4.7), we obtain system (2.4.5).

Concerning the numerical solution of the optimization problem, we mention here two possible approaches. The first one consists in solving the linear system (2.4.5). An iterative solver is clearly a recommended choice, and `symmlq` [64] would be a suitable choice; this approach has been used in [21]. Another approach consists in applying an iterative solver to the minimization of $\hat{J}(u)$. We focus here on this second approach, sketching the conjugate gradient method applied to the minimization of $\hat{J}(u)$. In the algorithm, let us denote by g_k the gradient $\nabla\hat{J}(u_k)$ at step k and by d_k the descent direction.

Conjugate gradient method

1. Choose an initial guess u^0
2. Compute h_0 and p_0 solving (2.4.6) and (2.4.7) and g_0 by (2.4.8)
3. Set $d_0 = -g_0$, $k = 0$
4. While $g_k \neq 0$
   - 4.1. Compute λ_k with a line search along d_k
   - 4.2. Compute $u_{k+1} = u_k + \lambda_k d_k$
   - 4.3. Update $g_{k+1} = g_k + \lambda_k \hat{G} d_k$
   - 4.4. Compute $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$
   - 4.5. Update $d_{k+1} = -g_{k+1} + \beta_{k+1} d_k$
   - 4.6. $k = k + 1$

Due to linearity, Step 4.3 is equivalent to compute $g_{k+1} = \hat{G}u_{k+1} + \hat{q}$. Indeed,

$$g_{k+1} = \hat{G}u_{k+1} + \hat{q} = \hat{G}(u_k + \lambda_k d_k) + \hat{q} = \hat{G}u_k + \hat{q} + \lambda_k \hat{G} d_k = g_k + \lambda_k \hat{G} d_k.$$

Nonetheless, we remark that this step is clearly performed without forming matrix $\hat{G}$, but rather computing vector $y_k = \hat{G}d_k$ through the following steps:

1. Solve $At = \mathcal{B}\, d_k$
2. Solve $A^T v = G^h t - \alpha B d_k$
3. Compute $y_k = \mathcal{B}^T\, v + G^u d_k - \alpha B^T t$

Furthermore, since $\hat{J}$ is quadratic, the stepsize λ_k in Step 4.1 can be computed via an exact line search. Given a descent direction d_k, we compute λ_k such that it minimizes the function $\phi(\lambda) := \hat{J}(u_k + \lambda d_k)$. Straightforward computations show that one has

$$\lambda_k = -\frac{d_k^T g_k}{d_k^T \hat{G} d_k}. \tag{2.4.9}$$

The stepsize λ_k is therefore computed without much effort, as quantity $\hat{G}d_k$ is the same needed in Step 4.3.

We remark that the most expensive part of the method is given by the solution of the linear systems with coefficient matrix A (which actually equals A^T). Nevertheless, we recall that matrix A is actually symmetric positive definite, block diagonal with each block defined on a fracture. The systems are therefore decomposed in as many small "local" systems as the number of fractures. Right-hand-sides of the local systems gather information both from the current fracture, and from the intersecting fractures, which are typically small in number. Hence, these independent linear systems can be efficiently solved on parallel computers.

2.5 VEM implementation and numerical results

In this section we address some implementation issues concerning the use of VEM in conjunction with the optimization approach described in section 2.4. In addition, we present some numerical results in order to show the viability of the VEM for the simulation of discrete fracture networks and to highlight the effectiveness of the overall method in this context. Simpler test problems focused on particular implementation issues anticipate some numerical results on more complex DFNs.

2.5.1 VEM for DFN

We start describing the procedure for obtaining the computing mesh on the fracture network. Let us recall that each fracture in a DFN is represented by a 2D polygonal domain and is intersected by other fractures of the network in a set of traces. As a first step, triangular meshes are generated on each fracture independently, without taking into account trace positions or conformity requirements of any kind. Next, we proceed independently on each fracture and whenever a trace intersects one element edge, a new node is created. New nodes are also created at trace tips. If the trace tip falls in the interior of an element, the trace is prolonged up to the opposite mesh edge. Intersected elements are then split into two new "sub-elements", which become elements

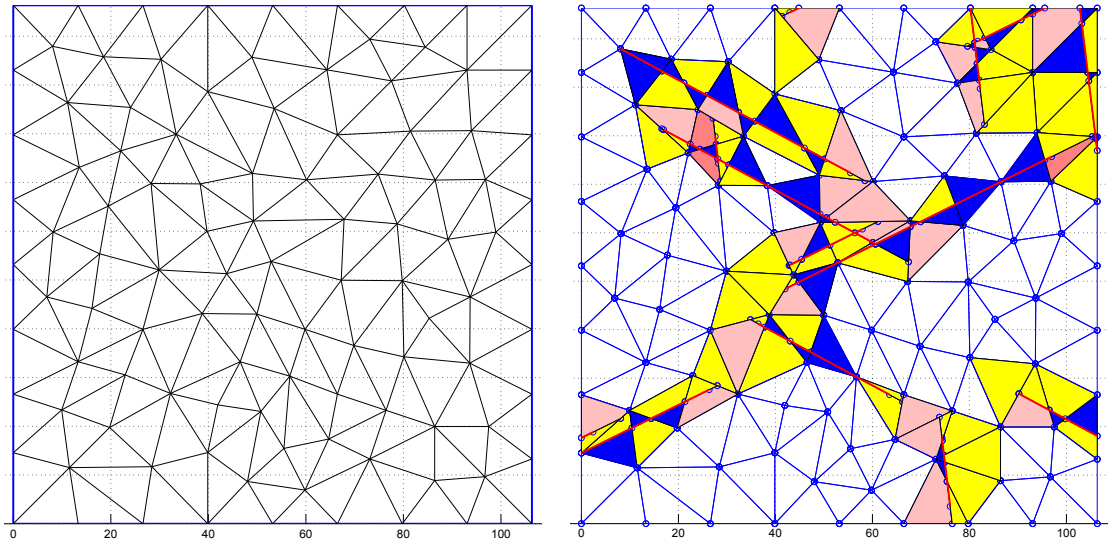


FIGURE 2.5.1: Mesh example. Left: original triangulation. Right: mesh for VEM

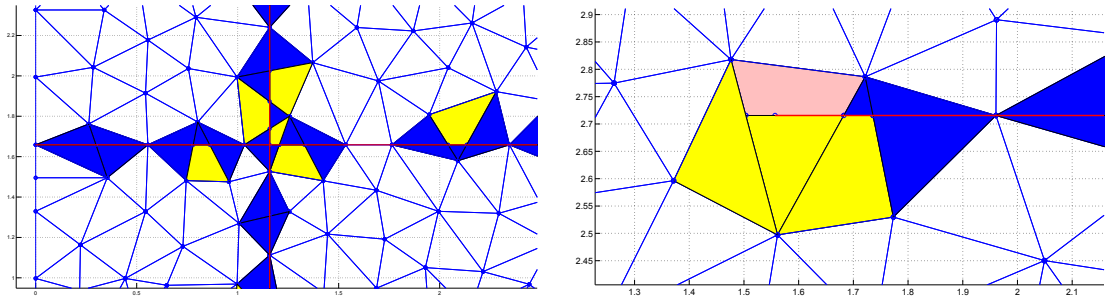


FIGURE 2.5.2: Left: detail of a mesh around a trace intersection. Right: detail of a mesh around a trace tip

in their own right, as shown in Figures 2.5.1 and 2.5.2 that represent the two phases of the process described above. In these pictures, coloured elements are the new virtual elements, whereas blank elements are the original triangular elements. Elements with up to 6 edges are introduced in these examples. In the figures, each color corresponds to a different number of edges in the element. The reader might refer to the PDF file to zoom in the pictures for a more detailed view.

The polygonal mesh obtained with the procedure described is possibly improved through the displacement of some nodes. Namely, when a node falls very close to a trace, it can be moved onto the trace itself, and therefore reducing the number of element edges and total degrees of freedom. The mesh improvement process is performed as detailed in the following. The distance of each node of intersected elements from the nearest trace is compared to a given mesh dependent tolerance. If the distance of the node to the closest trace is below the tolerance, then the node is moved to its projection on the trace. Vertices of the fractures always remain fixed and nodes in the border are only moved provided that they remain on the same border in order to avoid changing the

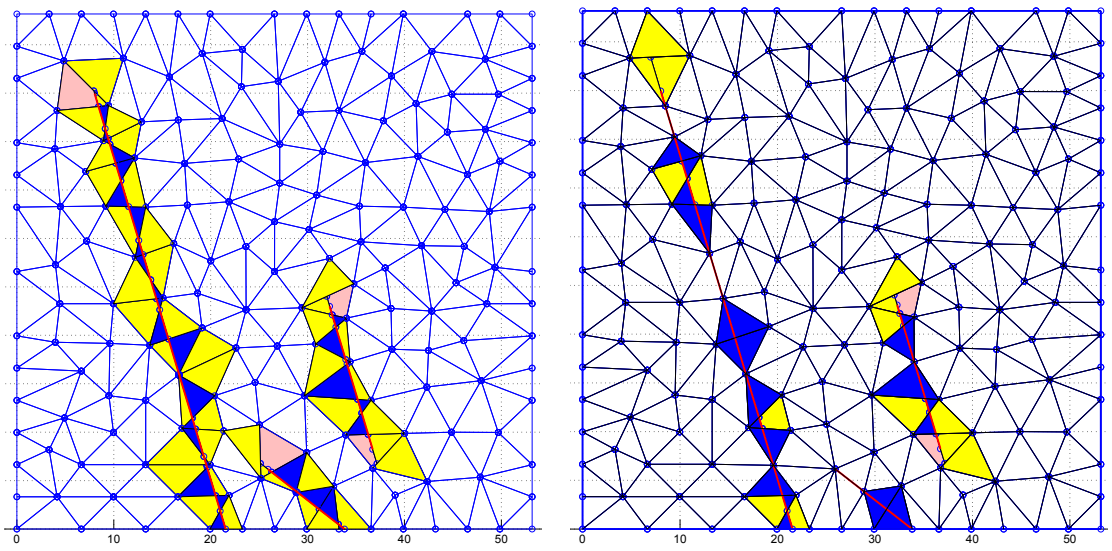


FIGURE 2.5.3: Left: example of VEM mesh without modification. Right: Same mesh after modifications

shape of the fracture. This procedure is performed independently for every fracture, and although not strictly necessary, it is advisable. The effect of this additional mesh modification is shown in Figure 2.5.3.

Since VEM basis functions are not known in the interior of mesh elements in general, we resort to the following mesh-dependent L^2 and H^1 norms commonly used in the context of mimetic finite differences, and defined $\forall u \in V_{i,\delta}$ and for all $i = 1, \dots, I$, respectively as:

$$||u||^2_{0,\delta} = \sum_{E \in \mathcal{T}_{i,\delta}} \left(\frac{|E|}{\partial E} \sum_{e \subset \partial E} |e| \left(\frac{u_h(v_i) + u_h(v_e)}{2} \right)^2 \right),$$

$$||u||^2_{1,\delta} = \sum_{E \in \mathcal{T}_{i,\delta}} \left(|E| \sum_{e \subset \partial E} \left(\frac{u_h(v_i) - u_h(v_e)}{|e|} \right)^2 \right),$$

where v_i and v_e are the initial and final point of the edge, respectively.

2.5.2 Test problems

We first propose two test problems aimed at evaluating VEM approximation capabilities in the DFN context by means of applying them to very simple configurations representative of common situations in DFN simulations. In these test cases, a single problem of the form (2.2.1) is solved, i.e. a single fracture F is considered, assigning u on the traces. In the first case, two intersecting traces are present in F, completely crossing the domain, while a single trace ending inside the domain is studied in the second problem.

The proposed numerical results show very good approximation capabilities of virtual elements in dealing with these geometrical configurations.

2.5.2.1 Problem 1

The first test problem, labeled P1, displays two traces intersecting each other inside the domain. The domain is a single rectangular fracture $F \subset \mathbb{R}^2$ with two traces S_1 and S_2 defined by:

$$F = \left\{(x,y) \in \mathbb{R}^2 : x \in (0,3), y \in (0,1)\right\},$$
$$S_1 = \left\{(x,y) \in \mathbb{R}^2 : x - y - 1 = 0\right\}, \qquad S_2 = \left\{(x,y) \in \mathbb{R}^2 : 2 - x - y = 0\right\}.$$

The domain is shown in Figure 2.5.4 with a coarse mesh with parameter $\delta_{max} = 0.2$ along with a detail of trace intersection. Here and in the sequel δ_{max} denotes the square root of the maximum element area for the initial triangulation on each fracture. For this mesh, the original triangular element containing trace intersection is split into four new elements, two triangles and two quadrilaterals.

The problem is set as follows:

$$\begin{aligned}
-\Delta H &= -\Delta H^{ex} && \Omega \setminus \mathcal{S},\\
H &= 0 && \text{on } \partial F,\\
U_1 = f_{S_1} &= \left[\!\left[\frac{\partial H^{ex}}{\partial \hat{\nu}_{S_1}}\right]\!\right]_S && \text{on } S_1,\\
U_2 = f_{S_2} &= \left[\!\left[\frac{\partial H^{ex}}{\partial \hat{\nu}_{S_2}}\right]\!\right]_S && \text{on } S_2,
\end{aligned}$$

with

$$H^{ex}(x,y) = \begin{cases}
xy(y-1)(x-y-1)(x+y-2)/7 & \text{in } A_1,\\
(1-y)(x-y-1)(x+y-2) & \text{in } A_2,\\
y(x-y-1)(x+y-2) & \text{in } A_3,\\
y(1-y)(x-3)(x-y-1)(x+y-2)/5 & \text{in } A_4,
\end{cases}$$

where A_1, A_2, A_3 and A_4 denote the four regions in which F is divided by the traces, as indicated in Figure 2.5.4. Values of f_{S_1} and f_{S_2} are

$$f_{S_1}(x,y) = \begin{cases}
1/(7\sqrt{2})(2-x-y)\,(7 - x(6+x) + 20y & \\
\quad +2x(1+x)y - 5xy^2 + y^3) & x+y-2 \le 0\\
1/(5\sqrt{2})(2-x-y)\,(-8 + y(1+y)(11+y) & \\
\quad +x^2(-1+2y) - x(1+y(4+5y))) & x+y-2 > 0,
\end{cases}$$

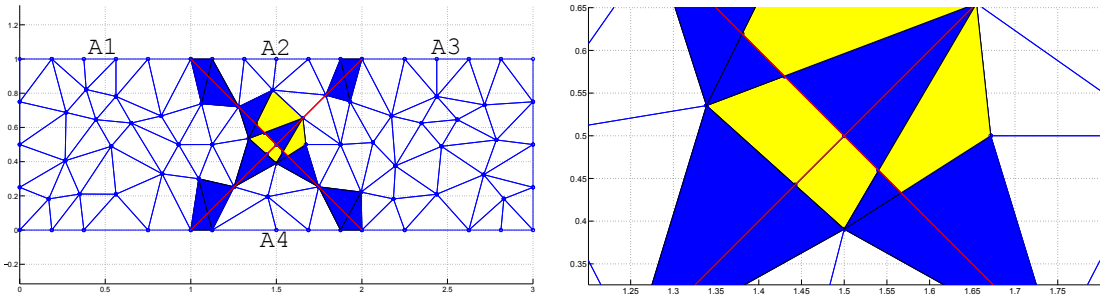


FIGURE 2.5.4: Problem P1. Left: Domain with coarse grid $\delta_{max} = 0.2$. Right: a detail of trace intersection

and

$$f_{S_2}(x,y) = \begin{cases} 1/(5\sqrt{2})(-1+x-y)\,(-16-(-10+x)x+38y \\ \quad +2(-7+x)xy+5(-3+x)y^2+y^3) & y-x+1 \le 0 \\ 1/(7\sqrt{2})(-1+x-y)\,\big(-28+x^2(-1+2y) \\ \quad +y(23+(-3+y)y)+x(9+y(-8+5y))\big) & y-x+1>0. \end{cases}$$

In Figure 2.5.5 the numerical solution obtained on a fine mesh with parameter $\delta_{max} = 0.05$ is displayed. This problem has been solved using both the VEM and the XFEM for the space discretization, as described in [20–22]. Figure 2.5.6 reports, for both space discretizations, errors computed versus the number of DOFs. We remark that, when applying the two approaches, we always start from the same triangular mesh. The XFEM deals with irregularities in the solution along traces by adding suitable enrichment functions (see [21, 22] and references therein), resulting the two methods in a different number of DOFs, when the same mesh parameter is used. Computed convergence rates are close to the expected ones both in the L^2 and the H^1 mesh-dependent norms, and both for the VEM and for the XFEM: namely, L^2 norm convergence rate is 1.03 for the VEM and 0.99 for the XFEM, whereas the H^1 norm convergence rate is 0.49 both for the VEM and for the XFEM. The L^2 norm of the error on the restriction of the solution to the traces is also reported (label 'L2H on trace' in the legend), and displays a convergence rate of 1.0 for the VEM and 0.91 for the XFEM. As a whole, the two space discretizations yield a comparable level of accuracy, and the intersection between traces is easily handled by the VEM on a polygonal mesh with very good approximation properties.

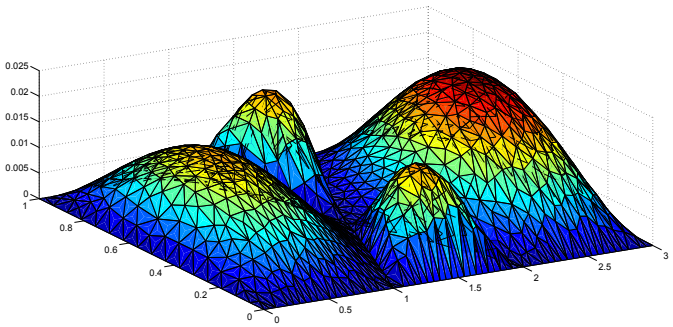


FIGURE 2.5.5: Problem P1: Approximate solution on a mesh with $\delta_{max} = 0.05$

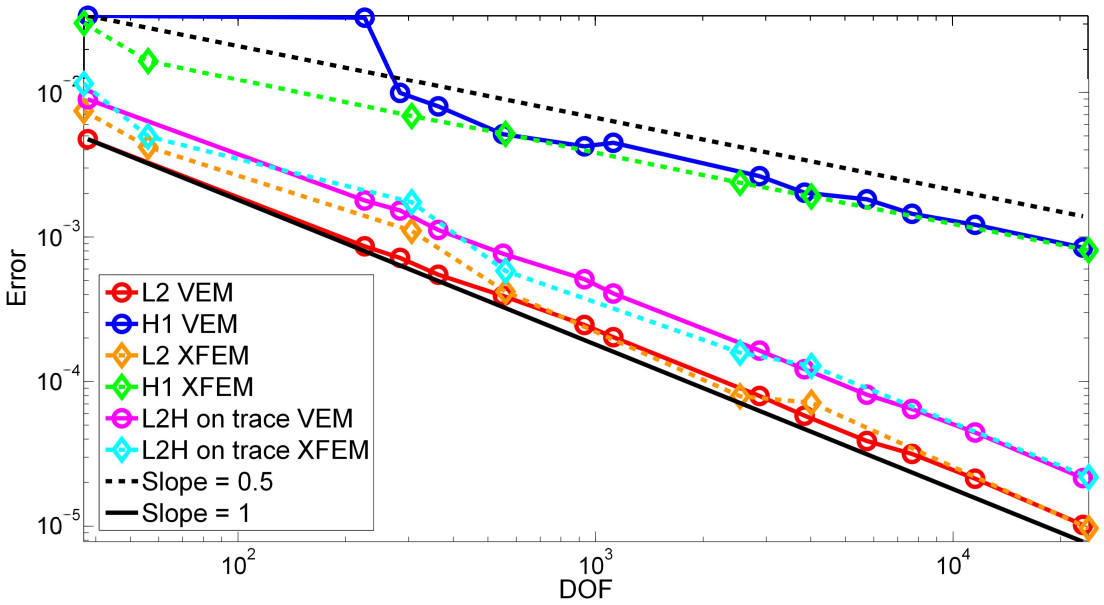


FIGURE 2.5.6: Problem P1: Error behavior

2.5.2.2 Problem 2

Let us define the domain F for the second test problem $P2$ as

$$F = \left\{(x, y) \in \mathbb{R}^3 : -1 < x < 1,\ -1 < y < 1,\ z = 0\right\},$$

with a single trace $S = \left\{(x, y) \in \mathbb{R}^2 : y = 0 \text{ and } -1 \leq x \leq 0\right\}$ ending in the interior of F. This test problem has also been considered in [21]. Here we set out to show the behaviour of virtual elements in handling the non-smooth behaviour of the solution around trace tips. Let us introduce the function $H^{ex}(x, y)$ in F as:

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

where arctan2(x, y) is the four-quadrant inverse tangent, giving the angle between the positive x-axis and point (x, y), and differs from the usual one-argument inverse tangent

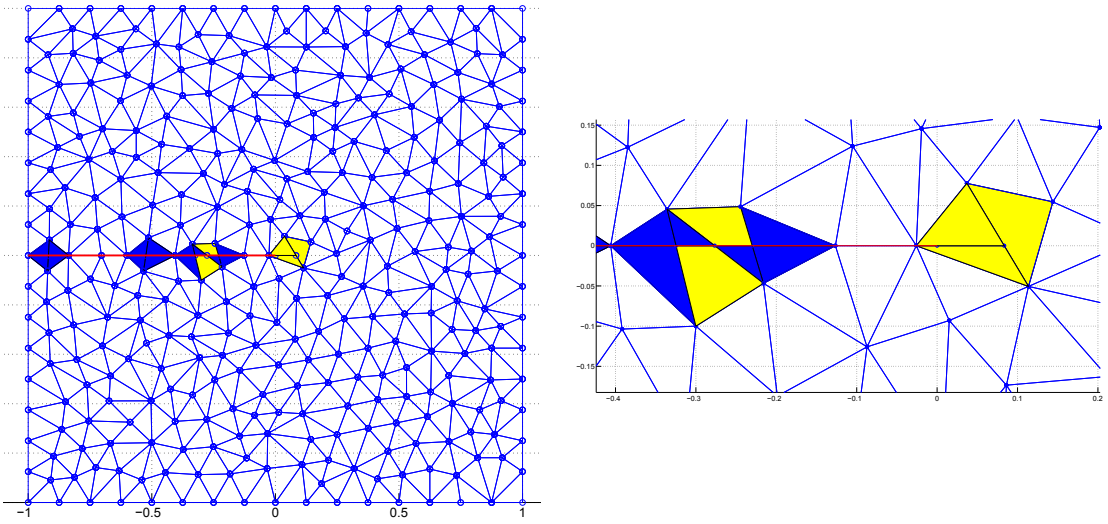


FIGURE 2.5.7: Problem P2: Domain meshed with $\delta_{max} = 0.1$. Right: a detail of elements near trace tip

arctan(·) for placing the angle in the correct quadrant.

The problem is defined by the system:

$$\begin{aligned} -\Delta H &= -\Delta H^{ex} && \text{on } \Omega \setminus S, \\ H &= 0 && \text{on } \partial F, \\ U &= x - x^3 && \text{on } S, \end{aligned}$$

where U is the prescribed value of the jump of fluxes across the trace S.

Figure 2.5.7 shows the VEM mesh and the resulting elements near the tip. In this implementation of the method, the tip becomes a new node of the triangulation, and three new four-sided elements are generated. Two of them are obtained from the original triangle that contained the trace tip, while the third one appears when the node given by the intersection between the prolonged trace and the opposite mesh element is added to the corresponding neighbouring triangle that becomes a quadrilateral.

The approximate solution is shown in Figure 2.5.8. In Figure 2.5.9 we report errors computed both with the L^2 and with the H^1 mesh dependent norms, both for the VEM and for the XFEM. Computed convergence rates are, also for this test problem, quite similar for the two space discretizations: 1.05 in the L^2 norm, and 0.51 in the H^1 norm for the VEM; 1.02 in the L^2 norm, and 0.47 in the H^1 norm for the XFEM. The Figure also reports the errors on the restriction of H to the trace S, computed in the L^2 norm. Computed convergence rate are in this case 0.85 for the VEM and 0.96 for the XFEM. As for problem P1, the approximation properties of the two space discretizations are therefore quite similar. As a whole, also this geometrical configuration including a trace

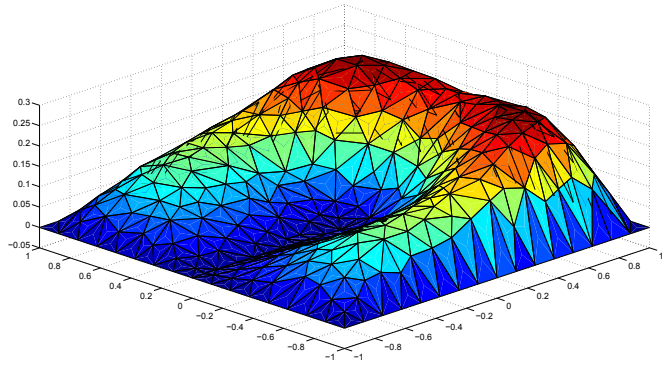


FIGURE 2.5.8: Problem P2: Approximate solution with VEM obtained with a mesh with $\delta_{max} = 0.1$

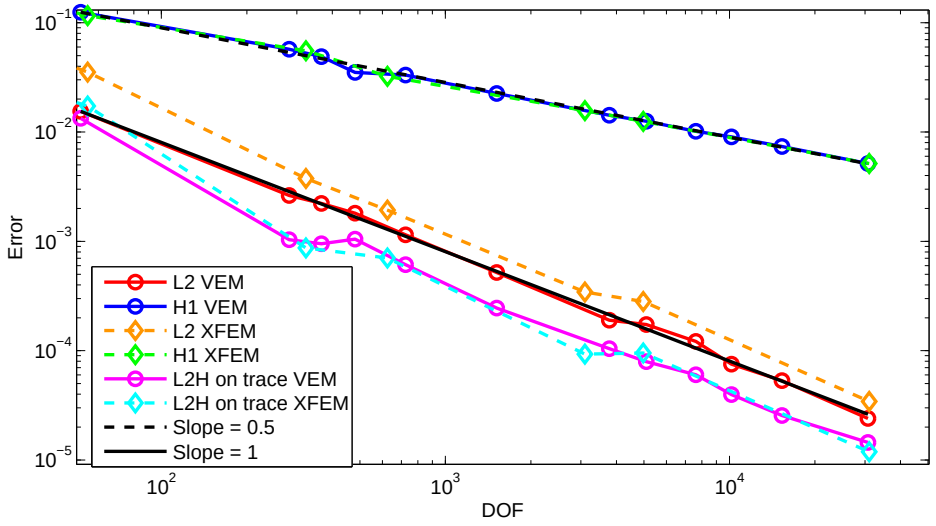


FIGURE 2.5.9: Problem P2: Error behavior

tip is effectively handled by the VEM, thanks to the flexibility in using polygonal mesh, without affecting the approximation capabilities if compared, e.g., with extend finite elements.

2.5.3 DFN problems

In this section we deal with networks of fractures, addressing both simple DFN problems and more complex and realistic problems. Computations are perfomed using the PDE-constrained optimization approach described, in conjunction with virtual element space

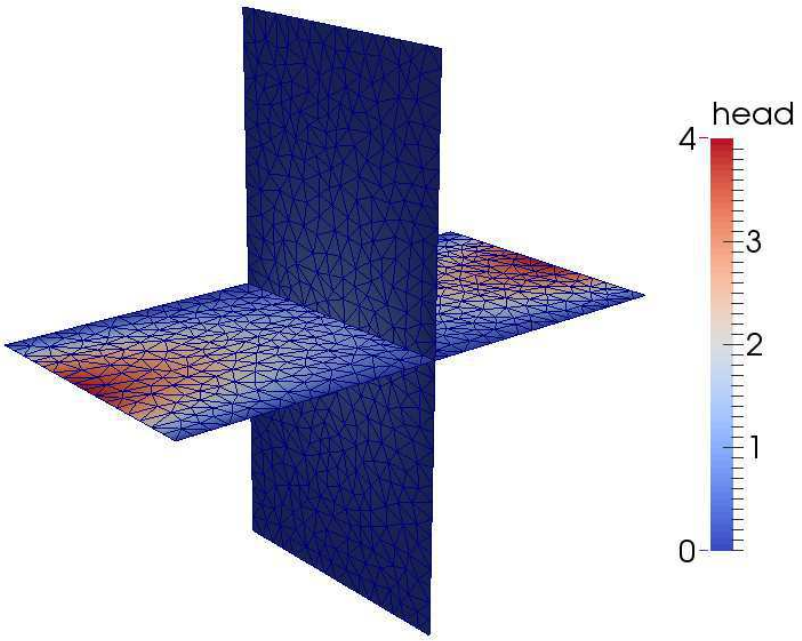


FIGURE 2.5.10: DFN2: spatial distribution of fractures and the obtained solution for the hydraulic head

discretization. The general DFN problem is set as follows:

$$\begin{aligned} -\Delta H &= q && \Omega \setminus \mathcal{S}, \\ H_{|\Gamma_D} &= H^D && \text{on } \Gamma_D, \\ \frac{\partial H}{\partial \hat{\nu}} &= G^N && \text{on } \Gamma_N, \end{aligned} \tag{2.5.1}$$

with reference to the nomenclature introduced in section 2.2.

2.5.3.1 DFN2

Here we analyze a very simple DNF consisting of two identical fractures that intersect each other orthogonally, as can be seen in Figure 2.5.10 where the domain Ω is depicted.

Fractures 1 and 2 and the trace S are defined as:

$$\begin{aligned} F_1 &= \left\{(x,y,z) \in \mathbb{R}^3 : z \in (-1,1), y \in (0,1), x = 0\right\}, \\ F_2 &= \left\{(x,y,z) \in \mathbb{R}^3 : x \in (-1,1), y \in (0,1), z = 0\right\}, \\ S &= \left\{(x,y,z) \in \mathbb{R}^3 : x = 0, y \in (0,1), z = 0\right\}. \end{aligned}$$

Homogeneous Dirichlet boundary conditions are imposed on the edges corresponding to $z = 0$ and $z = 1$ of F_1 and to $y = 0$ and $y = 1$ of F_2 . On the remaining edges

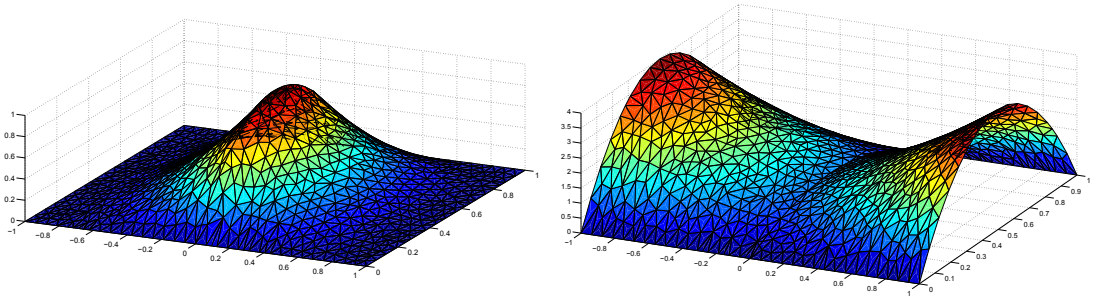


FIGURE 2.5.11: DFN2: approximate solution for fracture 1 (left) and fracture 2 (right)

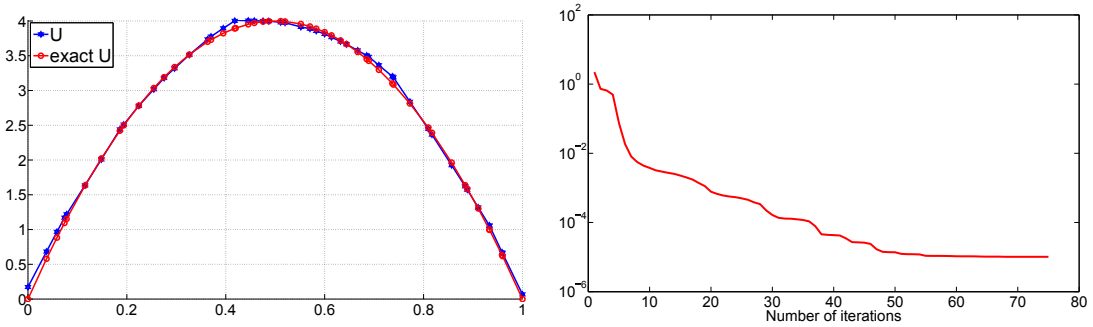


FIGURE 2.5.12: DFN2: Left: comparison between exact and approximate flux. Right: values of J versus number of iterations

we set homogeneous Neumann conditions for fracture F_1, and a non-constant Neumann boundary condition for fracture F_2 given by $G^N = 16y(1-y)^2$on Γ_N. With this definition of the problem, the exact solutions for the hydraulic head H^{ex} and the trace flux U are:

$$\begin{aligned} H_1^{ex}(x,y,z) &= \begin{cases} 4y(1-y)(z-1)^2 & \text{for } z \geq 0 \\ 4y(1-y)(z+1)^2 & \text{for } z < 0 \end{cases} \\ U_1^{ex}(x,y,z) &= 16y(1-y) \\ H_2^{ex}(x,y,z) &= \begin{cases} 4y(1-y)(x+1)^2 & \text{for } x \geq 0 \\ 4y(1-y)(x-1)^2 & \text{for } x < 0 \end{cases} \\ U_2^{ex}(x,y,z) &= -16y(1-y). \end{aligned}$$

In Figure 2.5.11 we present the results obtained for the hydraulic head on fracture F_1 (left) and F_2 (right) using a mesh size $\delta_{max} = \sqrt{0.002}$. Figure 2.5.12 shows the comparison of the obtained flux with the exact solution and the trend of the minimization of functional J against iteration number. Here, we have performed a number of iterations large enough to let J reach stagnation at its minimum. The computed flux relative to the minimum of the functional approximates the exact solution well.

Error norms are computed for the solution on the fractures in terms of the mesh-dependent L^2 and H^1 norms and are shown in Figure 2.5.13 against the number of degrees

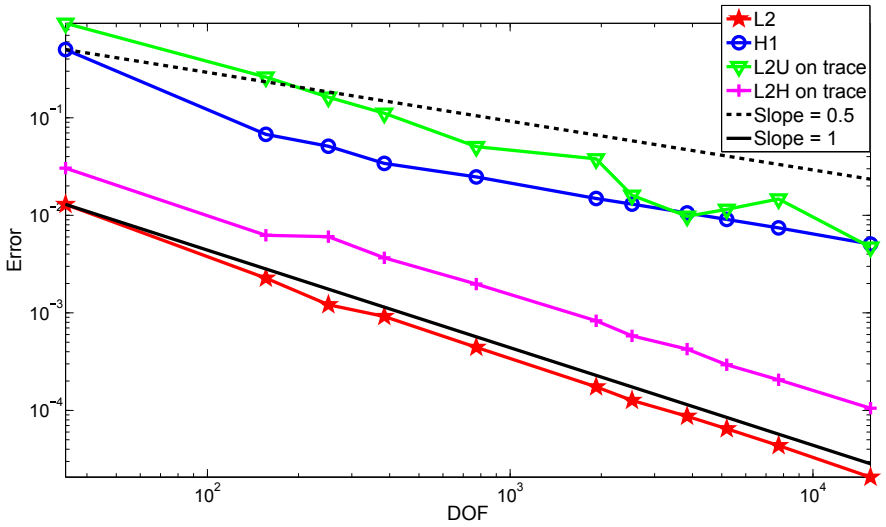


FIGURE 2.5.13: DFN2: error behaviour

of freedom. Errors for the flux on the trace and for the restriction of the solution h on the trace are also evaluated and displayed on the same figure. Convergence rates are of 1.05 and of 0.51 for the solution error in the L^2 and H^1 mesh dependent norms respectively, while a slope of 0.91 is shown for the L^2 error norm relative to the flux and a slope of 0.94 for the L^2 error norm of h at the trace. The results obtained show very good approximation properties of the VEM in conjunction with the proposed optimization method. Effectiveness of the method in handling more complex configurations is shown with the examples that follow.

2.5.3.2 DFN7

This problem consists of 7 fractures intersecting in 11 traces. The spatial distribution of the fractures can be seen in Figure 2.5.14. The source term is $q = 0$ in equation (2.5.1).

The Dirichlet boundary Γ_D is given by only two fracture edges: namely, constant Dirichlet boundary condition $H^D = 3$ is set on one edge of fracture F_3 (see Figure 2.5.14) and $H^D = 7$ is set on one edge of fracture F_7. On all the remaining boundaries of the network we set homogeneous Neumann conditions.

Due to the disposition of the fractures and the boundary conditions, the exact solution to this DFN problem is piecewise affine and displays a slope change at each trace (the jump in the slope corresponding to flux exchange). In this problem we show the capability of the VEM discretization, combined with the optimization approach, to correctly catch the solution in the space of discrete functions.

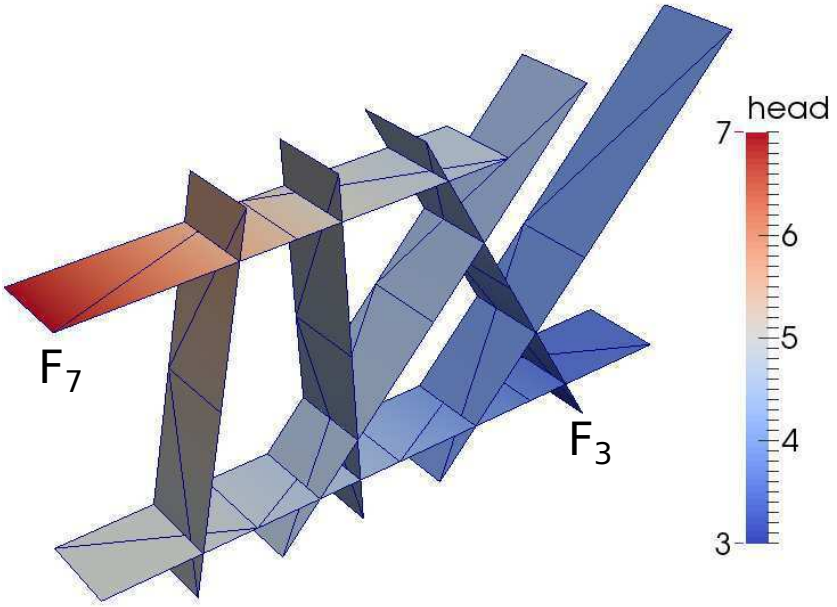


FIGURE 2.5.14: DFN7: spatial distribution of fractures and the obtained solution for the hydraulic head

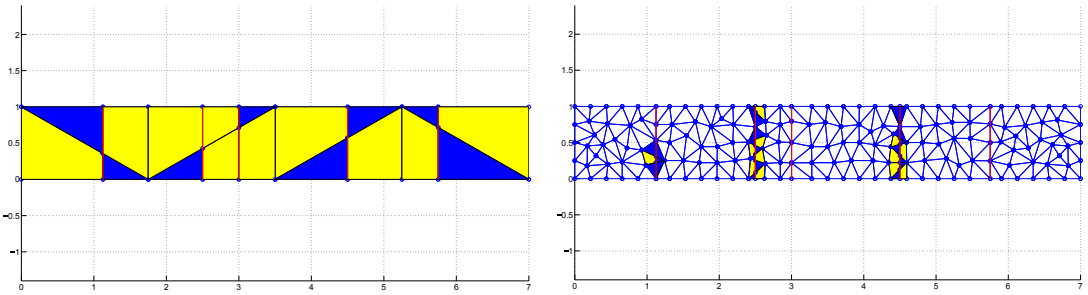


FIGURE 2.5.15: DFN7: mesh on F_6 with parameter $\delta_{max} = 1.2$ (left) and finer mesh with $\delta_{max} = 0.2$ (right)

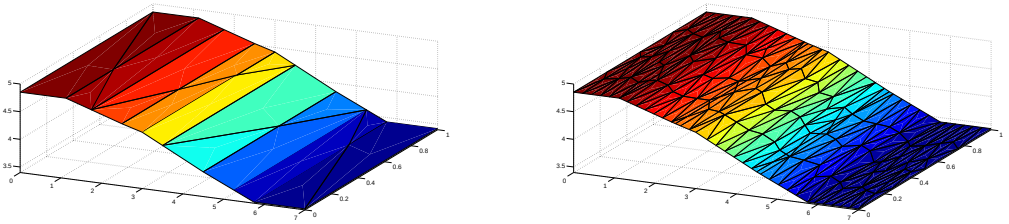


FIGURE 2.5.16: DFN7: solutions obtained for fracture 6 with coarse (left) and fine (right) mesh

Results are shown for a very coarse mesh (from 8 to 18 elements for each fracture) and for a finer mesh with $\delta_{\max} = 0.2$. See Figure 2.5.15 for a detail of the meshes for fracture 3.

Table 2.5.1 details the flux exchange in fractures and traces for the solution on the finer mesh. Rows correspond to traces and columns to fractures. The last row contains

TABLE 2.5.1: Flux data for the DFN7 configuration with flux mismatches across traces (last column) and flux balance on fractures (last row)

DFN7								
	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	
T1	-0.036			0.036				-9.8e-12
T2	-0.17					0.17		4.6e-12
T3	0.21						-0.21	-1.6e-12
T4		-0.24				0.24		-1.6e-12
T5		0.24					-0.24	-1.1e-11
T6			0.064	-0.064				-2.7e-12
T7			0.039		-0.039			-8.9e-12
T8			0.34			-0.34		1.1e-11
T9			0.31				-0.31	4.8e-12
T10				0.029		-0.029		8.3e-12
T11					0.039	-0.039		8.1e-13
	-2.1e-14	4.4e-14	0.7505	1e-14	4.2e-16	-1.4e-14	-0.7505	-5.9e-12

the sum of all the incoming and outgoing flow for each fracture, while the last column shows the balance in flux exchange between the two fractures that share a trace. An almost perfect balancing of the fluxes can be seen, both within fractures and in trace exchanges. Fracture F_7 acts as a source that provides 0.7505 of flux to the system (negative values represent flux leaving the fracture), which leaves the system at fracture F_3 with an approximately 0 unbalance reported in the bottom-right cell of the table. All other fractures show a quasi non-existent net flow, which agrees with the homogeneous Neumann boundary condition.

2.5.3.3 DFN36

We end the section with a realistic (though rather small) DFN consisting of 36 fractures intersecting in 65 traces. The spatial distribution of the fractures can be seen in Figure 2.5.17. Assuming meters as unit of length, fracture size spans from $2.8 \times 10^3 \mathrm{m}^2$ to $1.2 \times 10^4 \mathrm{m}^2$.

The Dirichlet boundary is composed by two edges of two fractures, namely Γ_D is composed by the borders of fracture F_1 and F_2 indicated in Figure 2.5.17, prescribing constant value Dirichlet conditions, $H_1^D = 100$ and $H_2^D = 0$. Homogeneous Neumann boundary conditions are set on all the remaining boundaries. With these boundary conditions fracture F_1 is a source of hydraulic head, F_2 is a sink fracture and all other fractures are insulated. Also in this case we set $q = 0$ in (2.5.1).

The problem is solved on several meshes, with $2\mathrm{m}^2 < \delta^2_{\max} < 50\mathrm{m}^2$. In Figure 2.5.18 the detail of a mesh with $\delta^2_{\max} = 30\mathrm{m}^2$ on a selected fracture and the corresponding obtained solution are shown.

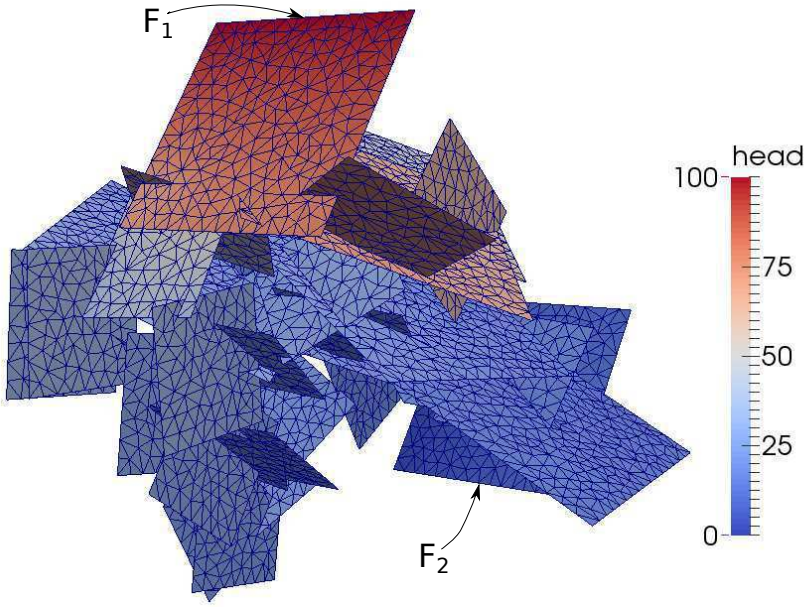


FIGURE 2.5.17: DFN36: Spatial distribution of fractures and the obtained solution for the hydraulic head

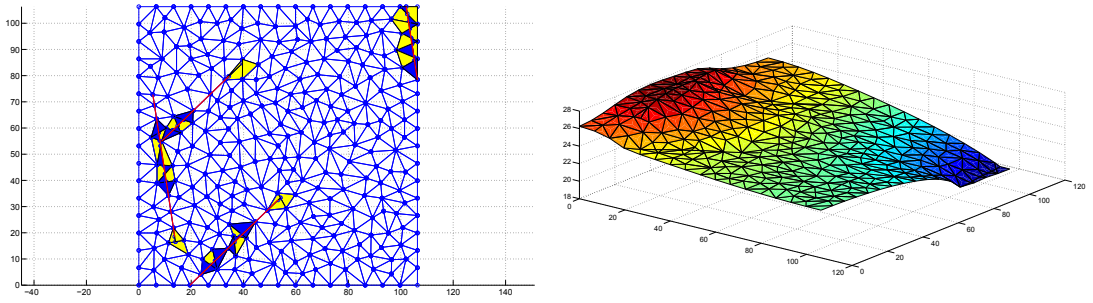


FIGURE 2.5.18: DFN36: Left: Mesh with maximum element size of 30m^2 on a selected fracture. Right: Solution on the same grid

The quality of the obtained solution can be evaluated in terms of two indicators, representing the mismatch errors in the continuity condition and in the flux balance condition on the traces per unit of trace length, defined respectively as:

$$\Delta_{\text{cont}} = \frac{\sqrt{\sum_{m=1}^{M} \|h_{i|_{S_m}} - h_{j|_{S_m}}\|^2}}{\sum_{m=1}^{M} |S_m|},$$

$$\Delta_{\text{flux}} = \frac{\sqrt{\sum_{m=1}^{M} \|u_i^m + u_j^m - \alpha(h_{i|_{S_m}} + h_{j|_{S_m}})\|^2}}{\sum_{m=1}^{M} |S_m|}.$$

These mismatch errors are reported in Table 2.5.2 for different mesh sizes. Namely, we report values obtained with both the VEM and the XFEM based space discretizations. The table also reports the number of degrees of freedom in the two cases, corresponding to each mesh parameter. We remark that the number of DOFs for u is the same in

TABLE 2.5.2: DFN36: Δ_{cont} and Δ_{flux} for various mesh sizes

		VEM			XFEM		
$\delta^2_{\max}$	u dof	h dof	Δ_{flux}	Δ_{cont}	h dof	Δ_{flux}	Δ_{cont}
50	776	4091	9.515e-04	9.432e-04	5772	1.039e-03	9.521e-04
30	942	6048	9.621e-04	8.394e-04	8106	1.147e-03	1.181e-03
12	1342	13967	6.736e-04	6.514e-04	16932	7.358e-04	8.189e-04
5	1885	30782	5.972e-04	6.083e-04	34958	5.930e-04	7.019e-04
2	2862	74107	4.847e-04	3.949e-04	80403	4.342e-04	4.664e-04

the two cases, as we use on the traces a finite element discretization which is induced by the intersection points among the initial triangular mesh element edges (the same for the two approaches) and the trace itself. On the other hand, the number of DOFs for h is different for the two approaches here adopted, and is in general smaller for the VEM. This is due to the fact that the XFEM deals with totally non-conforming meshes through the introduction of suitable enrichment functions in triangles close to the traces, thus yielding a bit larger number of DOFs. Note that this larger number of DOFs for the XFEM is required for handling a total non-conforming mesh, but it does not yield more accurate mismatch errors with respect to the VEM approach. As a whole, a good accuracy is obtained with both approaches, and the mismatch errors reduce with mesh refinement.

2.6 Conclusions

The very recent Virtual Element Method is coupled with the optimization based algorithm presented in [20–22] for the numerical simulation of DFNs on large scales. The flexibility of virtual elements in handling meshes with elements of fairly general polygonal shape allows an easy mesh generation process, reliable and independent on each fracture, suitable for the optimization approach used. The resulting method is robust as can approach any DFN with arbitrary fracture density, and efficient, since it provides an easy parallel approach to the simulation of large networks. The numerical results reported show the viability and effectiveness of the VEM for the simulation of DFNs.

Chapter 3

A globally conforming method

The contents of this chapter follow closely those found in the article "A globally conforming method for solving flow in discrete fracture networks using the Virtual Element Method" published in *Finite Elements in Analysis and Design*, Volume 109, February 2016, Pages 23–36.

3.1 Introduction

In this chapter, we aim to provide an easy, natural way for generating conforming meshes for complex DFN problems using the VEM. The proposed approach is a generalization of traditional conforming finite elements, keeping the method as simple and streamlined as possible. Some of the ideas presented here where present in the previous chapter, that introduced Virtual Elements (VEM) to DFNs. There, the VEM is used on locally conforming meshes and an optimization approach is adopted to handle the non-conformity of the global mesh. Here both local and global conformity is enforced, and classical approaches, borrowed from the domain decomposition methods, can be used to solve the problem. We make absolutely no assumptions on the meshing procedure, which is done independently for each fracture and without any consideration of the position of the traces. Traces are not modified in any way, and using some of the features of the VEM, local and global conformity for the mesh is obtained by means of splitting the original elements of the meshes independently generated on each fracture into polygons of an arbitrary number of vertices.

Using Lagrange multipliers we obtain a hybrid system that can be solved with different methods, including FETI algorithms for domain decomposition.

Section 3.2 provides the formulation of the DFN problem in the present context, whereas a brief summary of the VEM is reported in section 3.3, and in section 3.4 the proposed method is described in detail. Numerical results are presented in section 3.5, where some convergence results are given and the applicability of the method to DFNs is discussed.

3.2 The continuous problem

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

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

We define the following functional spaces:

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

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

and

$$V = \left\{ v : v_{|F_i} \in V_i,\ \forall i = 1, \ldots, N,\ \gamma_T(v_{|F_i}) = \gamma_T(v_{|F_j}),\ \forall T \in \mathcal{T}_i,\ \{i, j\} = \mathcal{I}(T) \right\},$$

where γ_T is the trace operator onto T. It is then possible to formulate the DFN problem, given by the Darcy's law in its weak form on the fractures with additional constraints of continuity of the hydraulic head across the traces: for $i = 1, \ldots, N$, find $H_i \in V_i^D$ such

that $\forall v \in V$

$$\begin{aligned} \sum_{i=1}^{N} \int_{F_i} \mathcal{K}_i \nabla H_i \nabla v_{|F_i} \mathrm{d}F_i &= \sum_{i=1}^{N} \left(\int_{F_i} f_i v_{|F_i} \mathrm{d}F_i + \langle g_i^N, v_{|\Gamma_{N_i}} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{N_i}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{N_i})} \right), \\ \gamma_T(H_i) &= \gamma_T(H_j), \ \forall T \in \mathcal{T}, \ \{i,j\} = \mathcal{I}(T) \end{aligned}$$

where $\mathcal{K}_i$ is the fracture transmissivity tensor, that we assume is constant on each fracture. The second equation represents the continuity of the hydraulic head across traces. On each fracture of the DFN the following bilinear form $a_i : V_i \times V_i \mapsto \mathbb{R}$ is defined as:

$$a_i(H_i, v_{|F_i}) = \int_{F_i} \mathcal{K}_i \nabla H_i \nabla v_{|F_i} \mathrm{d}F_i. \tag{3.2.1}$$

3.3 The Virtual Element Method

This section provides a quick overview of the VEM, recalling the main features useful in the present context. We refer the reader to the original paper [10] for a proper introduction and to [12] for a guide on implementation. Further developments can be found in [2], [32], [15] and [14]. The VEM has also been applied to problems in elasticity [11], plate bending [26], the Stokes problem [3] and has sparked interest in other applications as well.

Borrowing ideas from the Mimetic Finite Difference method [9, 55], the VEM can be regarded as a generalization of regular finite elements to meshes made up by polygonal elements of any number of edges. The discrete functional space on each element has, in general, not only polynomial functions but also other functions that are only known at a certain set of degrees of freedom. Given a bilinear form to be approximated with the VEM, our goal is to build a discrete bilinear form that coincides with the exact one when at least one of the arguments is a polynomial. For the other cases, a rough approximation that scales in a desired way is enough to obtain the desired convergence qualities of the method.

Given a domain $F \subset \mathbb{R}^2$, a mesh τ_h on F, made of polygons $\{E\}$ with mesh parameter h (i.e. the square root of the maximum element area), and the space of the polynomials of maximum order k, $\mathcal{P}_k$, let us define the local space $V_{k,h}^E$ for a given polynomial degree k as:

$$V_{k,h}^E = \left\{ v_h \in H^1(E) : v_{h|\partial E} \in C^0(\partial E), \ v_{h|e} \in \mathcal{P}_k(e) \ \forall e \subset \partial E, \ \Delta v_h \in \mathcal{P}_{k-2}(E) \right\}$$

where ∂E is the border of E, and e an edge.

From the above definition it is clear that the space $\mathcal{P}_k(E)$ is a subset of $V_{k,h}^E$. We define the following degrees of freedom for each element E:

- The value of v_h at the vertices of E;
- The value of v_h at $k-1$ internal points on each edge of E;
- The moments $\frac{1}{|E|}\int_E v_h m_\alpha$ for $|\alpha| \leqslant k-2$,

where m_α, with $\alpha = (\alpha_1, \alpha_2)$, represent scaled monomials of the type

$$m_\alpha = (\frac{x - x_c}{h_E})^{\alpha_1} (\frac{y - y_c}{h_E})^{\alpha_2},$$

and (x_c, y_c) and h_E are the centroid and the diameter of the element E respectively. Different choices for the second type of degree of freedom is possible instead of point values, e.g. edge moments. We have chosen point values on Gauss-Lobatto nodes on edges for numerical integration purposes. The selected set of degrees of freedom is unisolvent [10], and therefore, given an element E with n_v vertices, we have that the dimension of $V_{k,h}^E$ is $\#V_{k,h}^E = n_v k + \frac{k(k-1)}{2}$. We finally choose a basis for $V_{k,h}^E$, made of functions ϕ_i with $i = 1, ..., \#V_{k,h}^E$, such that, calling $\mathrm{dof}_j(v)$, for $j = 1, \ldots, \#V_{k,h}^E$ the j-th degree of freedom applied to v, we have $\mathrm{dof}_j(\phi_i) = \delta_{ij}$, being δ_{ij} the Kronecker delta. The global virtual element space is:

$$V_{k,h} = \left\{ v_h \in H^1(F) : v_h|_E \in V_{k,h}^E \ \forall E \in \tau_h \right\},$$

and we can easily check that the chosen degrees of freedom on the edges of each element allow to easily enforce continuity of any function $v_h \in V_{k,h}$ on the internal edges of the partition τ_h.

Let us now consider the restriction of the bilinear form (3.2.1) to a mesh element E, $a_i^E(.,.)$. We aim at building a discrete bilinear form $a_{i,h}^E : V_{k,h}^E \times V_{k,h}^E \mapsto \mathbb{R}$ having the previously stated polynomial consistency, i.e. the discrete bilinear form has to coincide with the exact one when at least one of the arguments is a polynomial of maximum degree k. To this end let us consider the projector operator of order k on E:

$$\Pi_{E,k}^{\nabla} : V_{k,h}^E \longrightarrow \mathcal{P}_k(E)$$

such that

$$\Pi_{E,k}^{\nabla} q_k = q_k \text{ for all } q_k \in \mathcal{P}_k(E),$$

defined by the equations

$$\int_E \nabla q_k \cdot \nabla v_h = \int_E \nabla q_k \cdot \nabla \Pi^\nabla_{E,k} v_h \text{ for all } q_k \in \mathcal{P}_k(E),$$
$$\int_E \Pi^\nabla_{E,k} v_h = \int_E v_h.$$

The projection $\Pi^\nabla_{E,k} v_h$ can be uniquely defined starting from the degrees of freedom of v_h using integration by parts [12] and represents an orthogonality condition in the H^1 inner product. The first equation defines the projection up to a constant, which is defined by the second equation. Other options for the second equation exist [14]. For order $k = 1$, it can be taken as

$$\frac{1}{N^v} \sum_{i=1}^{N^v} \Pi^\nabla_{E,k} v_h(\mathcal{V}_i) = \frac{1}{N^v} \sum_{i=1}^{N^v} v_h(\mathcal{V}_i)$$

where $\mathcal{V}_i$ are the vertices of the element and N^v its number.

Remark 3.1. In the case of a more complex equation than the Laplacian (or even the Laplacian with non-constant coefficients), other projectors have to be considered [14].

Let us now take any symmetric, positive definite bilinear form $S^E_{i,h} : V^E_{k,h} \times V^E_{k,h} \mapsto \mathbb{R}$, such that there exist c_0 and c_1 positive constants, independent of the element E and its diameter, that verify

$$c_0 a^E(v_h, v_h) \leq S^E_{i,h}(v_h, v_h) \leq c_1 a^E(v_h, v_h) \qquad \forall v_h \in V^E_{k,h} \text{ with } \Pi^\nabla_{E,k} v_h = 0.$$

This implies that $S^E_{i,h}$ *scales* like $a^E_i(v_h, v_h)$, and then the local discrete bilinear form $a^E_{i,h}$ is set as

$$\begin{aligned} a^E_{i,h}(u_h, v_h) \quad = \quad & a^E_i(\Pi^\nabla_{E,k} u_h, \Pi^\nabla_{E,k} v_h) + \\ & S^E_{i,h}(u_h - \Pi^\nabla_{E,k} u_h, v_h - \Pi^\nabla_{E,k} v_h) \; \forall u_h, v_h \in V^E_{k,h}. \end{aligned}$$

The first terms ensures the *consistency* and the second one the *stability* of the form. Finally, the complete discrete bilinear form becomes

$$a_{i,h}(u_h, v_h) = \sum_{E \in \tau_h} a^E_{i,h}(u_h, v_h) \qquad \forall u_h, v_h \in V_{k,h}.$$

A possible choice for the bilinear form $S^E_{i,h}$ is the usual Euclidean product in $\mathbb{R}^{\#V^E_{k,h} \times \#V^E_{k,h}}$ between two vectors whose components are the values of the functions at the degrees of freedom. A stiffness matrix K_i is associated to the discrete bilinear form $a_{i,h}$, defined

as :

$$(K_i)_{pq} = a_{i,h}(\phi_q, \phi_p), \text{ for } p, q = 1, ..., \#V_{k,h}.$$

In general it is not true that the VEM stiffness matrix approximates the exact stiffness matrix as if it were computed numerically.

For the right hand side with load term f, it is enough for optimal convergence [12] to consider

$$\begin{aligned} (f, v_h) &= \sum_{E \in \tau_h} \int_E f \Pi^0_{E,k-1} v_h \qquad \text{for order } k = 1, 2, \\ (f, v_h) &= \sum_{E \in \tau_h} \int_E f \Pi^0_{E,k-2} v_h \qquad \text{for order } k \geq 3, \end{aligned}$$

where $\Pi^0_{E,k}$ is the the full L^2 projection on the polynomials of degree k.

3.4 Problem implementation

3.4.1 Mesh generation

Mesh generation is done independently for each fracture regardless of traces and their positions. The process of mesh generation consists of three steps: the first task is the generation of a baseline triangulation of each fracture, not necessarily conforming to trace disposition, and independent on each fracture; the second step is the generation of a fracture-local conforming mesh, splitting the triangles of the baseline mesh into polygons conforming to the traces; finally on each fracture F_i, nodes are added on the traces $T \in \mathcal{T}_i$ corresponding to the nodes of the intersecting fracture F_j with $\{i, j\} = \mathcal{I}(T)$, $\forall T \in \mathcal{T}_i$, thus gaining global conformity. The three steps are depicted in Figure 3.4.1, and, the second and third steps are further described in full details in the next paragraphs.

3.4.1.1 Local conformity

Local conformity is obtained as in the previous work [17]. Every time a trace intersects an edge of the triangulation, a new node is created there. Nodes are also created at trace tips. If a trace tip is inside a triangular element, we extend the geometrical segment coinciding with the trace up to the nearest edge of the triangulation, thereby creating

a new edge and a new node. The trace is not modified, being now a subset of the extended segment. By doing this, we split the original elements of the triangulation into new convex "sub-elements", which are elements of the mesh in their own right. The end result is a mesh of polygonal elements for which all traces are covered by element edges, see Figures 3.1(a) and 3.1(b), where element colouring indicates the number of edges. A careful inspection of those Figures reveals all of the situations described above.

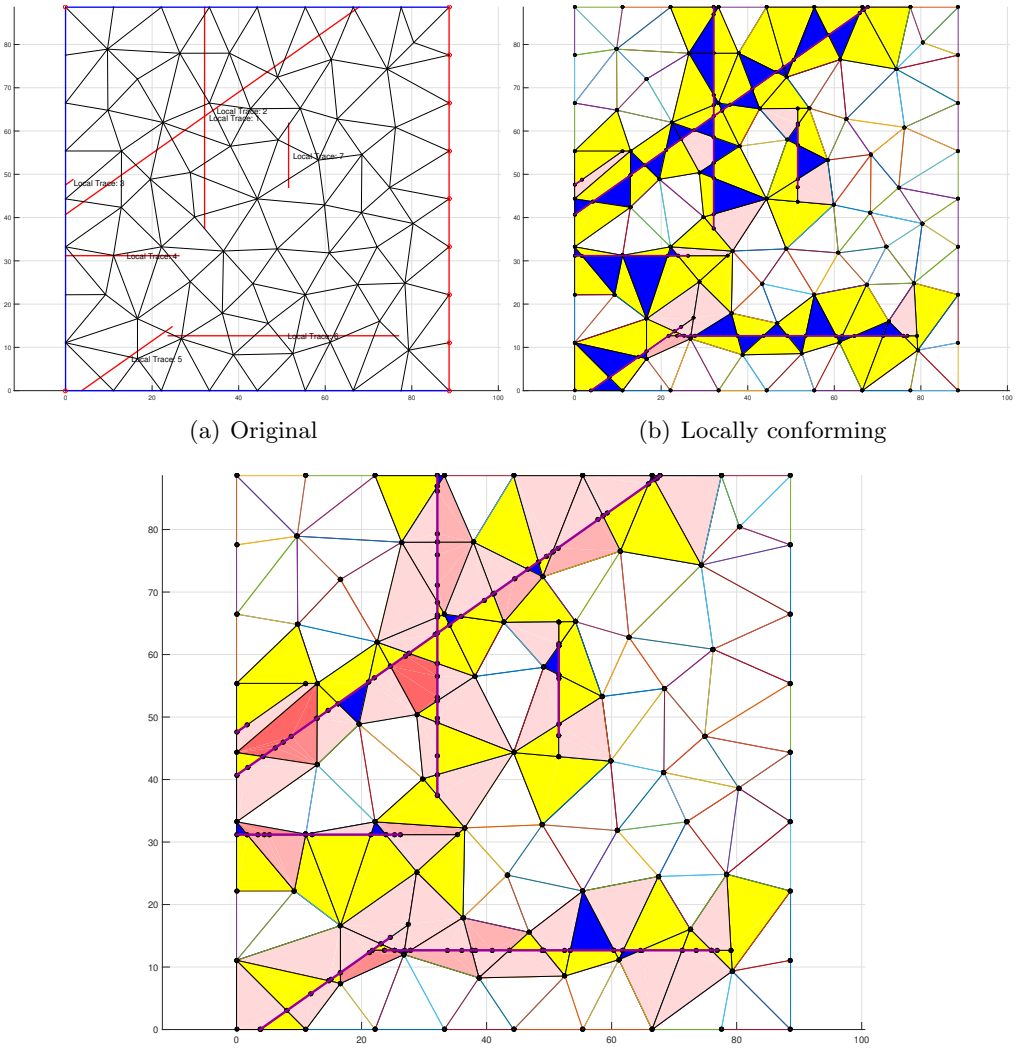
Remark 3.2. An optional mesh modification has been implemented that rearranges some of the nodes of the baseline triangulation before the splitting process, so as to make them coincide with nearby traces, trace tips and trace intersections. This leads to better shaped elements and fewer DOFs for the final mesh and it is not computationally demanding.

3.4.1.2 Global conformity

After obtaining the locally conforming mesh the subsequent step is to ensure that all the nodes on the traces are included in the meshes of both fractures that share the trace. These nodes are the ones shared by more than one fracture. This is the most important feature of the method we are proposing and takes full advantage of VEM versatility. Given a trace T shared by fractures F_i and F_j, we define $U_T^{F_i}$ as the set of all nodes on the trace T in fracture F_i and analogously $U_T^{F_j}$ for F_j. The procedure used to obtain the global conforming mesh guarantees that both trace tips are included and that the discretization includes all nodes on the traces and covers it precisely. The complete trace discretization is then $U_T = U_T^{F_i} \cup U_T^{F_j}$. What remains now is to simply add the set of nodes $U_T \setminus U_T^{F_i}$ on the corresponding elements of fracture F_i and analogously for fracture F_j. This can be done since the VEM allows for elements of arbitrary number of edges and 180° angles between them. The final globally conforming mesh is shown in Figure 3.1(c) and is identical to the previous mesh except for the new added nodes on the traces and a change in element colouring that is an indication of the increment in the number of edges and DOFs.

3.4.2 Imposing matching conditions

For every fracture F_i, with $i = 1, ..., N$, we call n_{dof_i} the number of DOFs of fracture F_i and we assemble the stiffness matrix $K_i \in \mathbb{R}^{n_{dof_i} \times n_{dof_i}}$ following the procedure described in section 3.3. Then we construct the column vectors $f_i \in \mathbb{R}^{n_{dof_i}}$ as the vector of load values (including terms arising from non-homogeneous boundary conditions) and h_i as the vector of nodal values of the discrete solution. We note that the matrix K_i is singular for fractures with pure Neumann boundary conditions. For the complete DFN we have:



(a) Original

(b) Locally conforming

(c) Globally conforming

FIGURE 3.4.1: Original mesh, VEM mesh and final globally conforming mesh

$$K = \begin{pmatrix} K_1 & 0 & \cdots & 0 \\ 0 & K_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & K_N \end{pmatrix}, \; f = \begin{pmatrix} f_1 \\ \vdots \\ \vdots \\ f_N \end{pmatrix} \text{ and } h = \begin{pmatrix} h_1 \\ \vdots \\ \vdots \\ h_N \end{pmatrix}.$$

In order to obtain the saddle point linear system for the complete DFN we have to impose matching conditions for the nodes on the traces that guarantee the continuity condition of the hydraulic head. We do that by means of Lagrange multipliers λ_t, for $t = 1, ..., n_{dof_t}$. They are introduced for each node on the traces in a non-redundant way (see [50]) which means that in the case of two intersecting traces, i.e. three fractures sharing a single point in space (as in the example of section 3.5.1.2), only two multipliers are added. To each index $t = 1, \ldots, n_{dof_t}$ corresponds a node on a trace T that is shared by fractures F_i and F_j, and we denote by $dof_i(t)$ the corresponding global DOF for node

t on F_i and analogously by $dof_j(t)$ the DOF on F_j. We define $N^h = \sum_{i=1}^{N} n_{dof_i}$, and the row vector $\mathfrak{L}_t \in \mathbb{R}^{N^h}$ as:

$$\mathfrak{L}_t = \Big(0 \quad \cdots \quad 0 \quad \overset{dof_i}{1} \quad 0 \quad \cdots \quad 0 \quad \overset{dof_j}{-1} \quad 0 \quad \cdots \quad 0\Big). \tag{3.4.1}$$

Finally, we set $\mathfrak{L} \in \mathbb{R}^{n_{dof_t} \times N^h}$ as the matrix:

$$\mathfrak{L} = \begin{pmatrix} \mathfrak{L}_1 \\ \vdots \\ \vdots \\ \mathfrak{L}_{n_{dof_t}} \end{pmatrix}.$$

The final linear system is:

$$\begin{bmatrix} K & \mathfrak{L}^T \\ \mathfrak{L} & 0 \end{bmatrix} \begin{bmatrix} h \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{3.4.2}$$

This saddle point problem has a unique solution as it can be easily proven resorting to classical results of quadratic programming [61].

When the dimensions of the system 3.4.2 are large, the use of an iterative method and of a preconditioner is advised. We briefly recall the one-level FETI method for domain decomposition as described in [49] here implemented. In this method the primal variables are determined in terms of the Lagrange multipliers. More precisely, we define a block diagonal matrix R as

$$R = \begin{pmatrix} R_1 & 0 & \cdots & 0 \\ 0 & R_2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & R_N \end{pmatrix}$$

where each sub-matrix R_i, for $i = 1, \ldots, N$ is such that its columns form a basis of the kernel of K_i, $ker(K_i)$, so that $ker(K) = range(R)$. In the case of the Laplacian operator, R_i corresponds to constant solutions for the subdomains with pure Neumann boundary conditions. Subdomains with Dirichlet boundary conditions have a unique solution and therefore have no contribution for R. It can be shown that

$$h = K^*(f - \mathfrak{L}^T\lambda) + R\alpha$$

where K^* is the pseudoinverse of K and the vector α depends on λ but not on the primal variables h. This means that if we solve a system for λ, this completely determines the solution. In order to solve this system for λ, a choice of several preconditioners is possible.

We give a brief outline of the procedure to obtain the Dirichlet preconditioner for the one-level FETI, denoted M^{-1}. Let us define $\mathcal{K}^t$ as the sum of transmissivity values of the fractures that share the node associated with the degree of freedom t. We first multiply the coefficient $(\mathfrak{L})_{t,dof_i(t)}$ by $\mathcal{K}_i/\mathcal{K}^t$ and the coefficient $(\mathfrak{L})_{t,dof_j(t)}$ by $\mathcal{K}_j/\mathcal{K}^t$. This takes into account the relative weight of the transmissivity coefficient of each fracture with respect to the sum of the transmissivity coefficients of the fractures associated with that node. We collect then the new coefficients in a matrix $\mathfrak{L}_D$. Then, for each fracture we denote by τ the set of fracture DOFs corresponding to nodes placed on the traces, and by ζ the set of the remaining DOFs and we can rearrange matrices K_i to obtain:

$$\tilde{K}_i = \begin{bmatrix} K_i^{(\zeta\zeta)} & K_i^{(\tau\zeta)T} \\ K_i^{(\tau\zeta)} & K_i^{(\tau\tau)} \end{bmatrix}.$$

The local Schur complement S_i is defined as:

$$S_i = K_i^{(\tau\tau)} - K_i^{(\tau\zeta)}(K_i^{(\zeta\zeta)})^{-1}K_i^{(\tau\zeta)T}.$$

If we call S the block diagonal Schur complement matrix of the whole system, the Dirichlet preconditioner for the one-level FETI is:

$$M^{-1} = \mathfrak{L}_D S \mathfrak{L}_D^T.$$

This is called Dirichlet preconditioner as a consequence of the fact that for each application of the preconditioner a local Dirichlet problem has to be solved. The lumped preconditioner is defined similarly as:

$$M^{-1} = \mathfrak{L}_D K^{(\tau\tau)} \mathfrak{L}_D^T,$$

where $K^{(\tau\tau)}$ is the block diagonal matrix made up by the local $K_i^{(\tau\tau)}$. We note that in order to define inner products for the Preconditioned Conjugate Gradient (PCG) FETI algorithm, a symmetric, positive definite matrix Q is used [49]. In our experiments we have considered $Q = M^{-1}$.

3.5 Numerical results

In this section we present some numerical results, beginning with convergence results for benchmark problems and VEM spaces of various orders. We also compare the results obtained with this approach to the results of a validated XFEM based method on a medium size DFN [22, 23]. We conclude showing some examples of numerical instabilities arising mainly with the higher order VEM approximation spaces for certain particularly adverse geometrical configurations. All of the results were obtained using a constant transmissivity tensor $\mathcal{K} = 1$ for all fractures.

3.5.1 Convergence results

The error norms used for the convergence curves are the usual L^2 and H^1 norms. The error is computed by taking the projection of the discrete solution on the space of polynomials, since the values of the discrete solution are only known at the DOFs and are not explicitly known inside the elements (see [14]):

$$Err^2_{\mathrm{L}^2} = \sum_{E \in \mathcal{T}_\delta} ||H - \Pi^\nabla_{E,k} h_E||^2_{\mathrm{L}^2(E)},$$
$$Err^2_{\mathrm{H}^1} = \sum_{E \in \mathcal{T}_\delta} ||H - \Pi^\nabla_{E,k} h_E||^2_{\mathrm{H}^1(E)}$$

where $\Pi^\nabla_{E,k}$ is the projection operator of order k as defined in section 3.3, H is the exact solution and h_E is the discrete solution restricted to element E.

The flux incoming in a fracture through the traces is computed as the jump of the conormal derivative of the discrete solution across the traces. For every trace we fix a tangential orientation and a normal unit vector obtained by clockwise rotating by 90° the tangent vector of the trace in the fracture plane. For every mesh edge $e \subset T$, i.e. an edge included in trace T, we consider a unique normal vector $\mathbf{n}_{e,i}$ in F_i with an orientation given by the normal vector fixed for the trace, and we define the flux incoming in the fracture F_i through the edge e, named $u_{e,i}$, as follows:

$$u_{\mathrm{left},e,i} = \nabla \Pi^\nabla_{E_l,k} h_{E,i} \cdot \mathbf{n}_{e,i},$$
$$u_{\mathrm{right},e,i} = -\nabla \Pi^\nabla_{E_r,k} h_{E,i} \cdot \mathbf{n}_{e,i},$$
$$u_{e,i} = u_{\mathrm{left},e,i} + u_{\mathrm{right},e,i},$$

where E_l and E_r are the elements to the left and to the right of the trace that share the edge e, respectively.

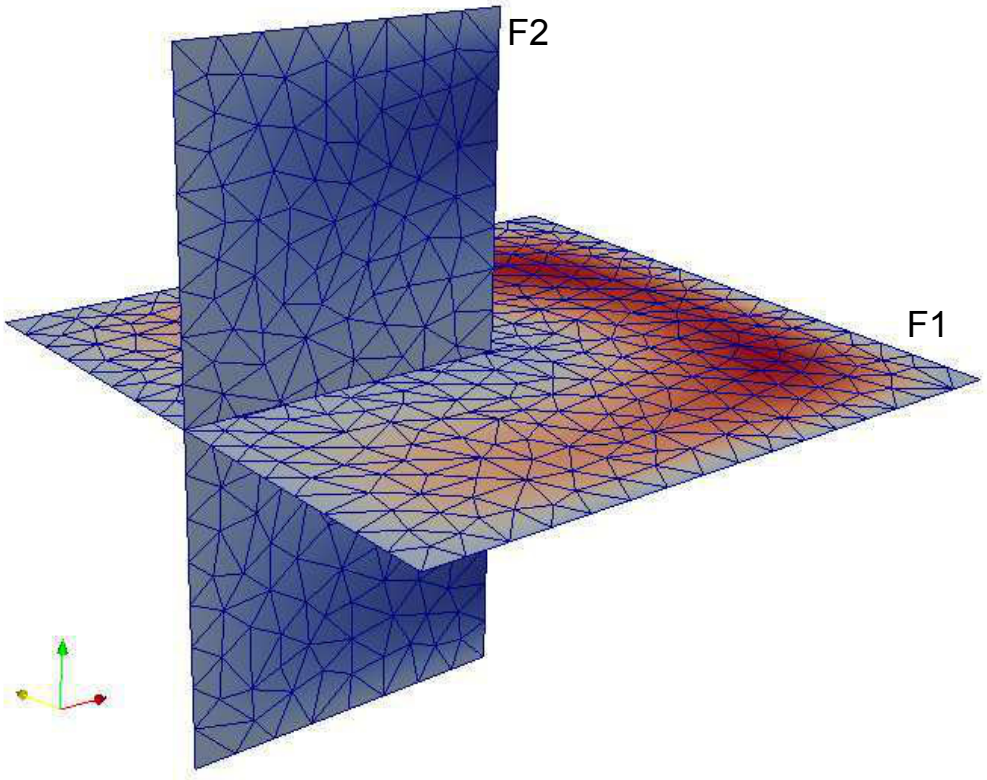


FIGURE 3.5.1: Spatial distribution of fractures for benchmark problem 1

The flux entering in the fracture F_i through trace T is then obtained by repeating this procedure over all the mesh edges in F_i belonging to T:

$$u_{T,i} = \sum_{e \subset T} u_{e,i}.$$

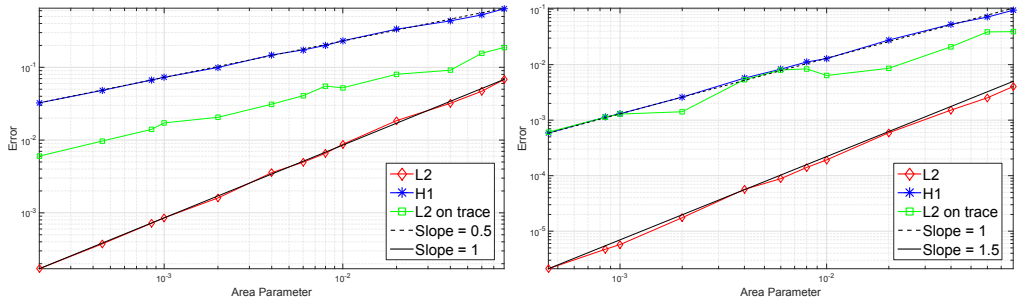
The L^2 error of the flux on the trace is then:

$$ErrU^2_{L^2} = ||U_{T,i} - u_{T,i}||^2_{L^2(T)},$$

where $U_{T,i}$ is the exact incoming flux in F_i through trace T.

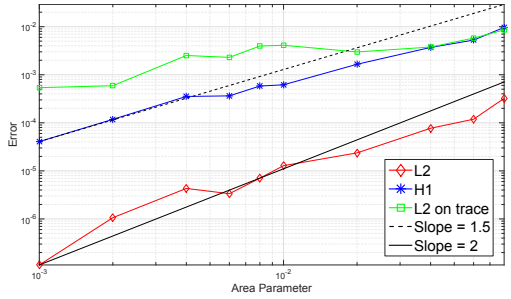
3.5.1.1 Benchmark problem 1

This first problem has been considered before in the context of the XFEM (eXtended finite elements) [21] and of the VEM in chapter 2 as a single-fracture problem. Nevertheless, it remains interesting for the fact that it includes a trace tip inside the domain and the exact solution is known. In this work the problem is considered as a 2-fracture DFN, as shown in Figure 3.5.1 and the error calculations and convergence curves are shown for the first fracture, F_1.

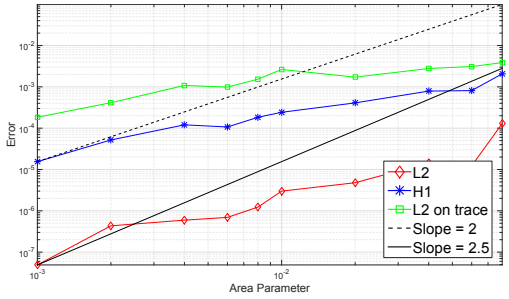


(a) Order 1

(b) Order 2

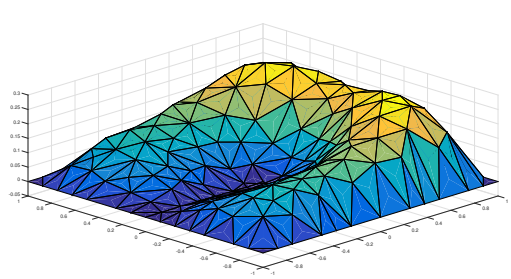


(c) Order 3

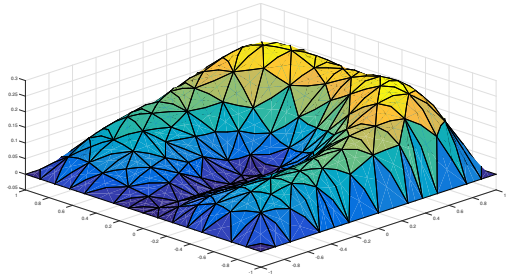


(d) Order 4

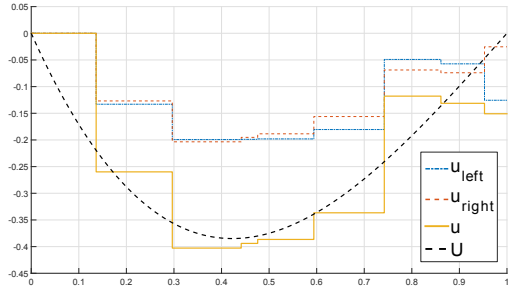
FIGURE 3.5.2: Convergence curves for benchmark problem 1 - Fracture 1



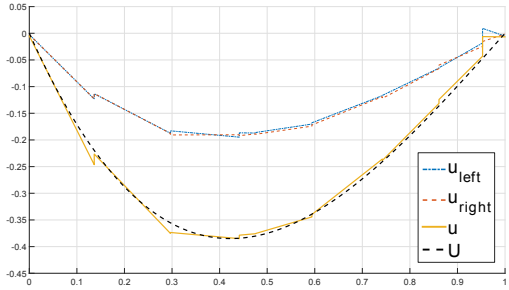
(a) h - Order 1



(b) h - Order 2



(c) u - Order 1



(d) u - Order 2

FIGURE 3.5.3: Solutions for benchmark problem 1 - Fracture 1

Let us define the domains F_1 and F_2 as

$$F_1 = \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq x \leq 1,\ -1 \leq y \leq 1,\ z = 0\right\},$$
$$F_2 = \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq x \leq 0,\ -1 \leq z \leq 1,\ y = 0\right\},$$

with a single trace $T = \left\{(x,y) \in \mathbb{R}^3 : y = 0, z = 0 \text{ and } -1 \leq x \leq 0\right\}$ ending in the interior of F_1 (Figure 3.5.1).

Exact solutions for F_1 and F_2 are given by $H_1^{ex}(x,y)$ and $H_2^{ex}(x,y)$:

$$H_1^{ex}(x,y,z) = -\cos\left(\frac{1}{2}\arctan2(x,y)\right)(x^2-1)(y^2-1)(x^2+y^2)$$
$$H_2^{ex}(x,y,z) = -\cos\left(\frac{1}{2}\arctan2(x,y)\right)(z^2-1)(x^2-1)(z^2+x^2)$$

where $\arctan2(x,y)$ is the arc-tangent function with 2 arguments, that returns the appropriate quadrant of the computed angle.

The problem is then:

$$\begin{aligned} -\Delta H &= -\Delta H_1^{ex} \text{ on } F_1 \setminus T, \\ H &= 0 \text{ on } \partial F_1, \end{aligned}$$

$$\begin{aligned} -\Delta H &= -\Delta H_2^{ex} \text{ on } F_2 \setminus T, \\ H &= (z^2 - z^4)\cos(\pi/4) \text{ on } \partial F_2^D \\ H &= 0 \text{ on } \partial F_2 \setminus \partial F_2^D. \end{aligned}$$

where $\partial F_2^D = \left\{(x,y,z) \in \mathbb{R}^3 : x = 0, y = 0, -1 \leq z \leq 1\right\}$ is the boundary of F_2 with non-homogeneous Dirichlet boundary conditions.

Convergence curves for the VEM of orders from 1 to 4 are shown in Figure 3.5.2. The expected rates of convergence are obtained for orders 1 and 2, whereas a slower rate of convergence for orders 3 and 4 was obtained as a consequence of the insufficient regularity of the exact solution in the sense of Sobolev spaces.

Numerical solutions for the hydraulic head H_1 with the VEM of orders 1 and 2 are shown in Figure 3.5.3 a) and b). In Figure 3.5.3 c) and d), we present a comparison between the exact solution and the approximate solution of the flux incoming in F_1, as well as its left and right components. Note how the approximation of the trace flux U is piecewise constant for order 1 VEM and piecewise linear for order 2 VEM, and the

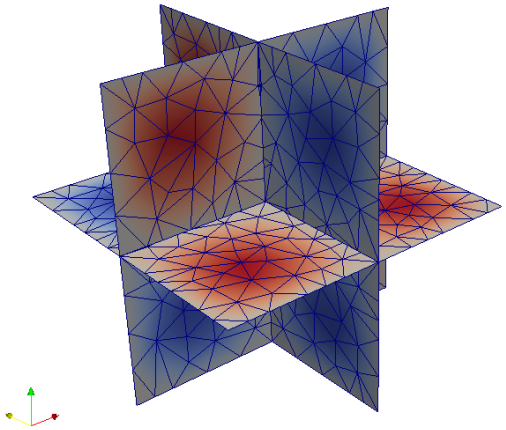
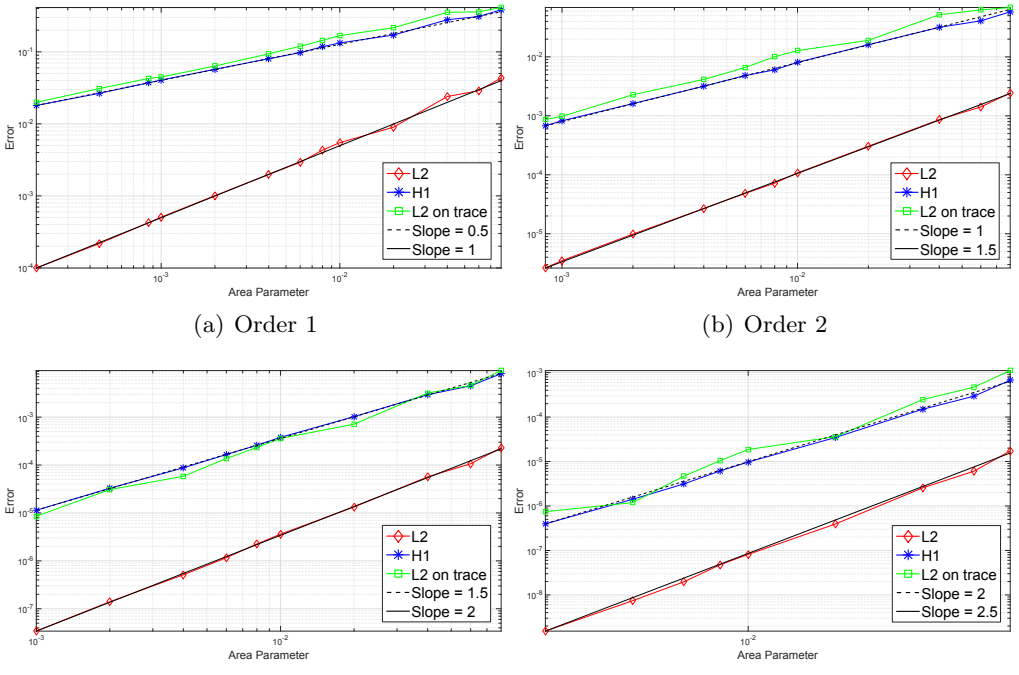


FIGURE 3.5.4: Spatial distribution of fractures for benchmark problem 2

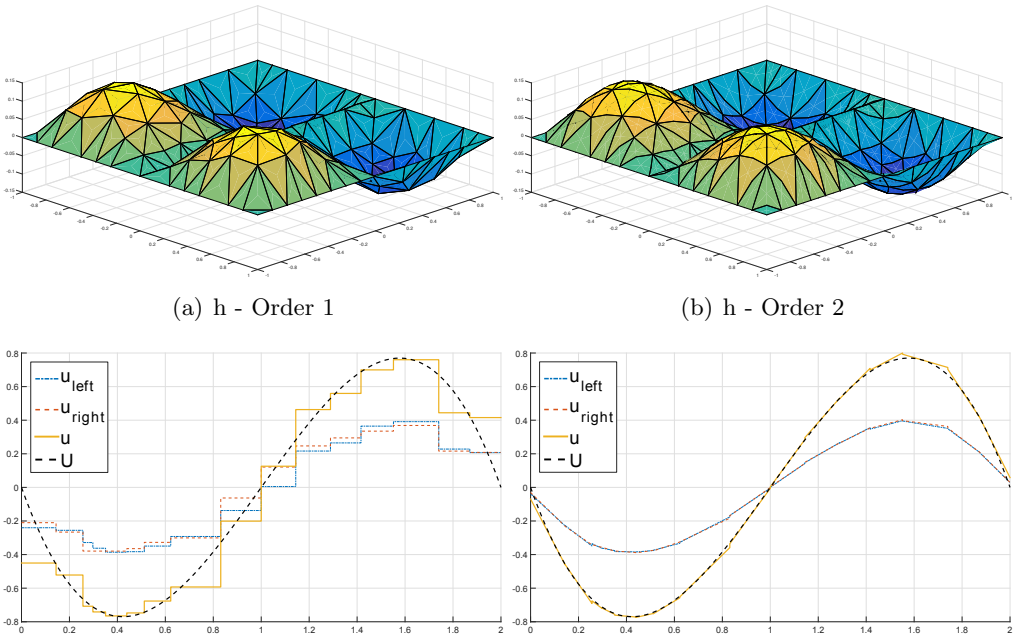


(a) Order 1 (b) Order 2

(c) Order 3 (d) Order 4

FIGURE 3.5.5: Convergence curves for benchmark problem 2 - Fracture 1

approximation of the exact flux (dashed line) with the VEM of second order is greatly improved.



(a) h - Order 1 (b) h - Order 2

(c) u - Order 1 (d) u - Order 2

FIGURE 3.5.6: Solutions for benchmark problem 2 - Fracture 1 and trace 1

3.5.1.2 Benchmark problem 2

This problem shows the performance of the proposed approach in presence of trace intersections. The considered system consists of 3 fractures and 3 traces as shown in Figure 3.5.4:

$$
\begin{aligned}
F_1 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq x \leq 1,\ -1 \leq y \leq 1,\ z = 0\right\},\\
F_2 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq y \leq 1,\ -1 \leq z \leq 1,\ x = 0\right\},\\
F_3 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq z \leq 1,\ -1 \leq x \leq 1,\ y = 0\right\},
\end{aligned}
$$

$$
\begin{aligned}
T_1 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq x \leq 1,\ y = 0,\ z = 0\right\},\\
T_2 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq y \leq 1,\ z = 0,\ x = 0\right\},\\
T_3 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \leq z \leq 1,\ x = 0,\ y = 0\right\}.
\end{aligned}
$$

Note that all of the three traces intersect in a single point $P = (0,0,0)$ in space (as it is always the case for the intersection of 3 planar fractures).

Exact solutions are known for all fractures:

$$H_1^{ex}(x,y) = |x|(1+x)(1-x)y(1+y)(1-y),$$
$$H_2^{ex}(y,z) = y(1+y)(1-y)|z|(1+z)(1-z),$$
$$H_3^{ex}(z,x) = z(1+z)(1-z)x(1+x)(1-x).$$

Note that H_1^{ex} and H_2^{ex} are not C^1 in the whole fracture, but, for each of the 4 subdomains defined by the traces in each fracture, they are polynomials of degree 6.

The problem is then:

$$\begin{aligned} -\Delta H &= 6|x|y(x^2+y^2-2) \text{ on } F_1 \setminus \mathcal{T}_1, \\ -\Delta H &= 6|y|z(y^2+z^2-2) \text{ on } F_2 \setminus \mathcal{T}_2, \\ -\Delta H &= 6zx(z^2+y^2-2) \text{ on } F_3 \setminus \mathcal{T}_3, \\ H &= 0 \text{ on } \partial F_1 \cup \partial F_2 \cup \partial F_3. \end{aligned}$$

Convergence curves for the VEM of orders from 1 to 4 are shown in Figure 3.5.5 and solutions for order 1 and 2 are reported in Figure 3.5.6. In contrast with benchmark problem 1, the expected convergence speed is achieved for all orders, since now the exact solution has C^∞ regularity on each of the subdomains defined by the traces and the mesh for the numerical solution is conforming to the traces. This is a sufficient condition for optimal convergence rates, [42, 72]. The error in the discrete solution for VEM of order 6 is $||H-h||^2_{\mathrm{L}^2} = 3.53e{-}19$, $||\partial_x(H-h)||^2_{\mathrm{L}^2} = 5.09e{-}18$ and $||\partial_y(H-h)||^2_{\mathrm{L}^2} = 5.85e{-}18$, being then of the same order of the round-off error in double precision. This confirms that the discrete solution coincides numerically with the exact solution.

3.5.2 DFN - 27 fractures

Let us consider the DFN shown in Figure 3.5.7 consisting of 27 fractures. A sink fracture F_1 and a source fracture F_2 are defined, both having a non homogeneous Dirichlet boundary conditions on one edge of their boundary and homogeneous Neumann boundary conditions on the remaining edges. All other fractures have homogeneous Neumann boundary conditions and are therefore insulated on their boundaries. In absence of an exact solution, the difference Δ between the flux entering the system from F_2 (the source fracture), "So", and the flux leaving it from F_1 (sink fracture), "Si", is considered for assessing the quality of the obtained numerical approximation.

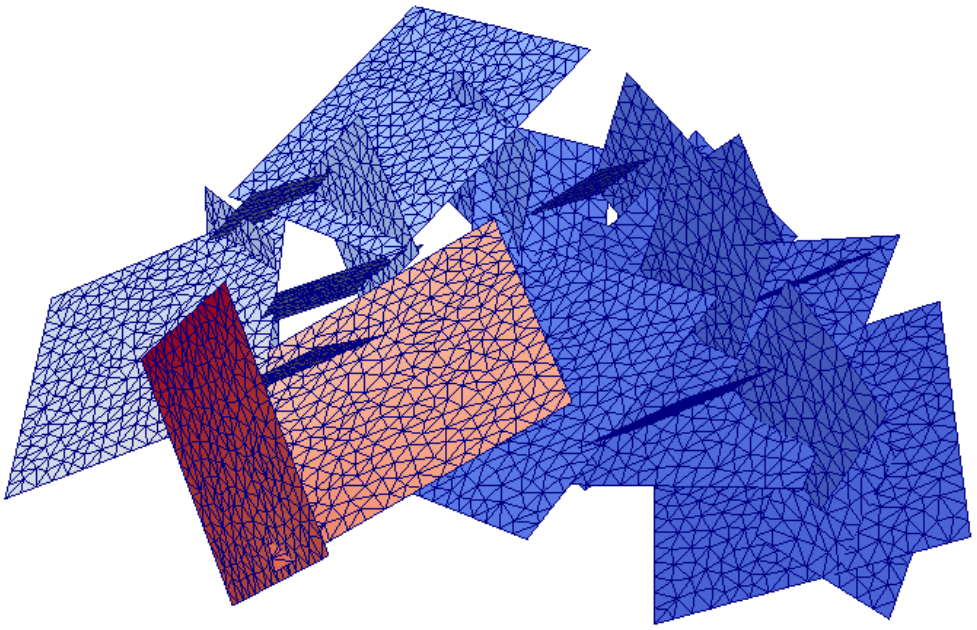


FIGURE 3.5.7: DFN27: spatial distribution of fractures for a DFN with 27 fractures

TABLE 3.5.1: DFN27: net flux in source (So) and sink (Si) fractures and flux mismatch Δ for various mesh sizes and VEM orders

	mesh 150			mesh 120		
Method	Si	So	Δ	Si	So	Δ
VEM-1	8.75	-8.22	0.53	8.70	-7.92	0.78
VEM-2	11.23	-9.78	1.45	11.16	-10.05	1.09
VEM-3	11.60	-10.36	1.23	11.64	-10.60	1.04
VEM-4	11.88	-10.76	1.12	11.89	-10.92	0.98
	mesh 90			mesh 60		
Method	Si	So	Δ	Si	So	Δ
VEM-1	9.01	-7.75	1.26	9.73	-8.32	1.42
VEM-2	11.18	-10.03	1.08	11.40	-10.26	1.14
VEM-3	11.64	-10.73	0.91	11.80	-10.89	0.9
VEM-4	11.91	-10.99	0.92	12.03	-11.17	0.86
	mesh 30			mesh 15		
Method	Si	So	Δ	Si	So	Δ
VEM-1	10.56	-8.51	2.05	10.71	-9.49	1.23
VEM-2	11.83	-10.77	1.06	11.91	-11.00	0.91
VEM-3	12.11	-11.25	0.86	12.13	-11.53	0.59
VEM-4	12.26	-11.48	0.78	10.21	-13.01	-2.81
	mesh 10			mesh 5		
Method	Si	So	Δ	Si	So	Δ
VEM-1	10.98	-9.18	1.81	11.36	-10.26	1.12
VEM-2	12.00	-11.09	0.90	12.12	-11.65	0.47

It should be noted that the methodology presented in this work does not guarantee nor aims to have local mass conservation in each fracture, since this is not explicitly imposed on any fracture. This means that the global mass conservation is well described, but the "local" flux balances (i.e., on each individual fracture) can be somewhat less accurate. On the other hand, these fracture flux balances are expected to improve with finer meshes as the method is converging to the solution. On the whole, the method can be seen as basically solving the DFN problem in one very complex 3D domain in space, that may however still be thought as a set of bidimensional domains.

Table 3.5.1 shows the net flux in the source and sink fractures, Si and So, respectively, as well as the difference Δ for mesh parameters (area of the largest element of the mesh) ranging from 5 to 150 and orders of the VEM space from 1 to 4. Only orders 1 and 2 are considered on the two finer meshes.

After extensive numerical experiments a trend emerged in the results; for order 1, convergence can be quite slow in the flux variable on these coarse meshes and displays oscillations, this can be attributed to the fact that the approximation of the flux is only piecewise constant and the projection of the VEM space functions for each element is onto a polynomial space of degree one, regardless of the number of edges of the element. Moving to higher order discretization spaces, the approximation of the flux improves. A marked improvement is obtained with second order VEM with respect to the first order, probably due to the piecewise linear structure of U. Further increasing the VEM order has a less noticeable effect, with practically no gain in moving to a third or fourth order approximation. In addition, higher order discretizations might suffer from numerical instabilities due to very badly shaped elements. This is for example the case for the fourth order approximation on the mesh size 15, where instabilities cause a degenerate discrete solution as shown by the parameter Δ reported in Table 3.5.1. Further details on possible causes of instabilities are discussed later in Paragraph 3.5.4.

Remark 3.3. When tackling a new DFN, a good practice would be to run it the first time with a coarse mesh and first order elements. The values of h and of u already provide a reliable indication of the order of magnitude of the correct solution, and using the flux values on each fracture one can establish a rule for selecting the fractures for which a mesh refinement is advisable. Fractures with less important contribution to the total flux through the DFN do not require a finer mesh. Afterwards, a new simulation can be launched with second order elements and the new adapted mesh.

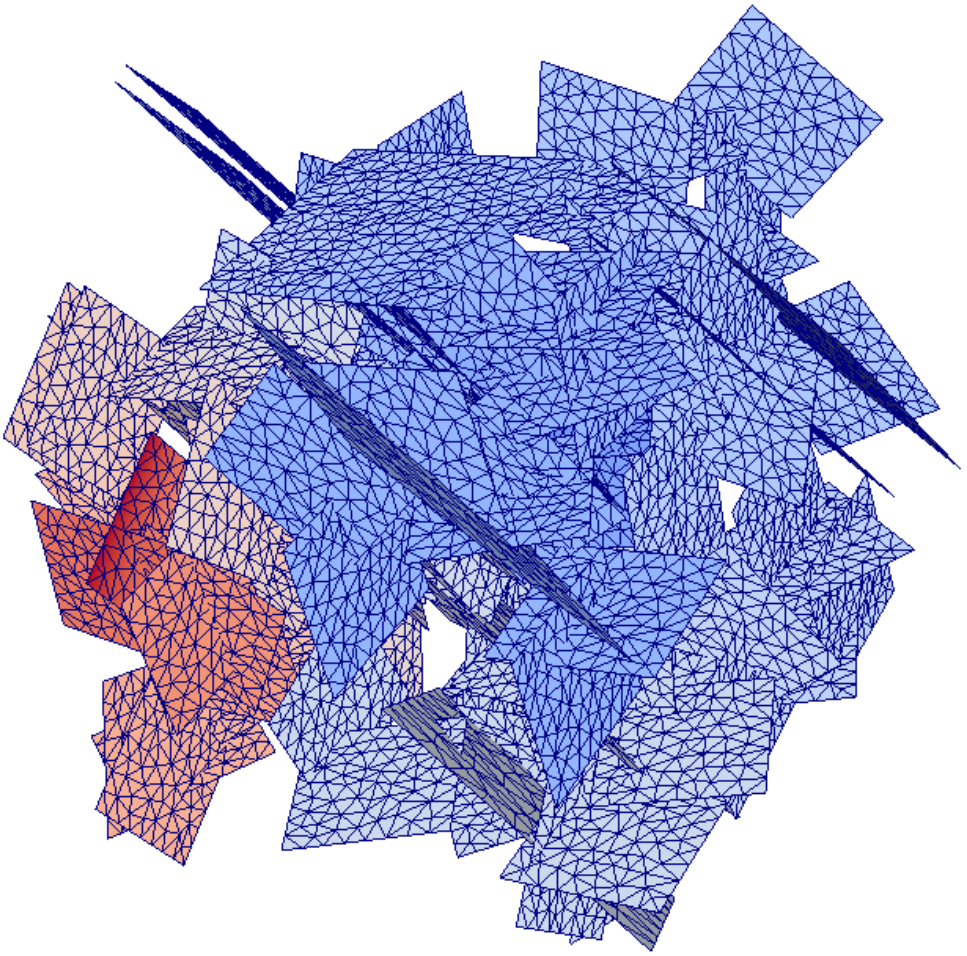
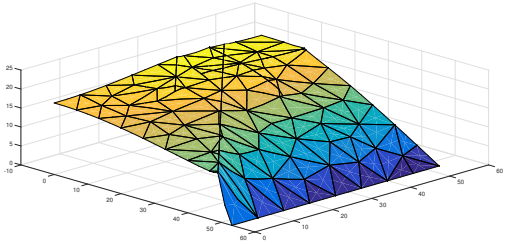
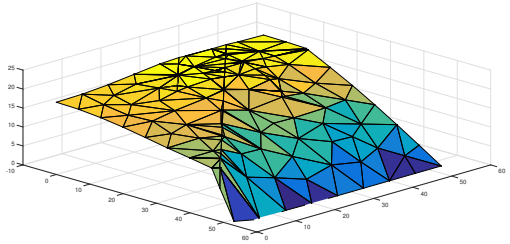


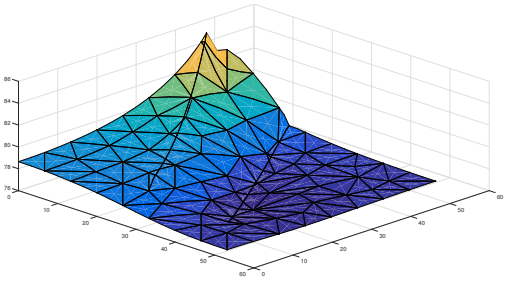
FIGURE 3.5.8: DFN116: spatial distribution of fractures for a DFN with 116 fractures



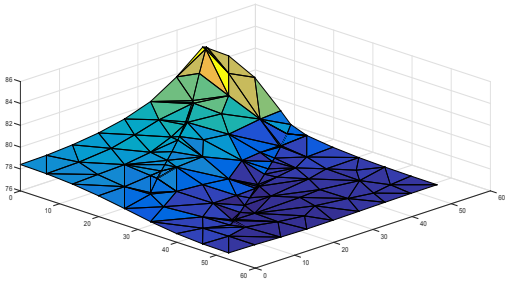
(a) h for fracture 1 (VEM - order 2)



(b) h for fracture 1 (XFEM)



(c) h for fracture 57 (VEM - order 2)



(d) h for fracture 57 (XFEM)

FIGURE 3.5.9: DFN116: large DFN comparison

TABLE 3.5.2: Comparison of iterations for different choices of Q and preconditioner M^{-1}

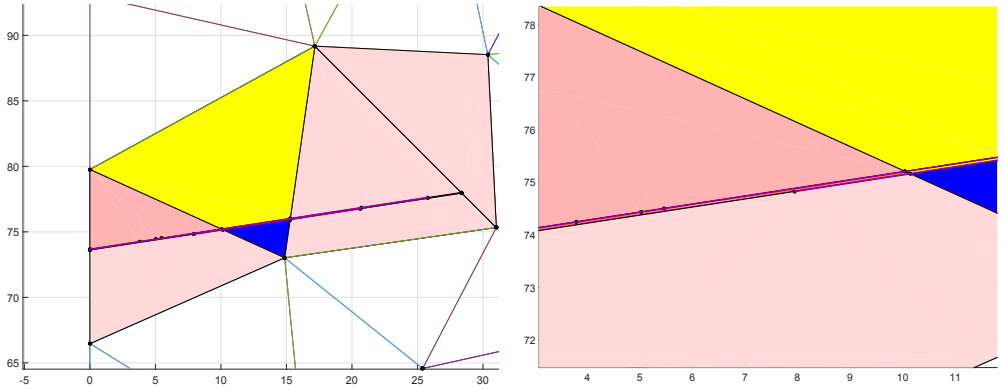
			CG	Lumped	Dirichlet
Method/Area	Total DOF	Trace DOF	Iter	Iter	Iter
VEM-1/150	7209	2047	137	106	72
VEM-1/90	9220	2524	152	118	77
VEM-1/30	19116	4182	29891	138	80
VEM-1/5	75672	9833	NC	238	113
VEM-2/150	25028	3869	181	259	77
VEM-2/90	34038	4823	4537	286	74
VEM-2/30	79736	8139	NC	357	112

3.5.3 DFN - 116 fractures

We now consider a DFN consisting of 116 fractures, as shown in Figure 3.5.8. Dirichlet boundary conditions are imposed on a source and sink fracture whereas all other fractures have homogeneous Neumann boundary conditions. In Figure 3.5.9 we plot the solution for the sink fracture and for a selected fracture with insulated boundaries. As a comparison, results are shown for both the VEM approach of order 2 depicted in the present work and for the XFEM based optimization approach described in [22], starting from the same baseline mesh. A very good agreement between the solutions can be appreciated in the figure. Good agreement was also obtained for VEM of orders 1 and 3.

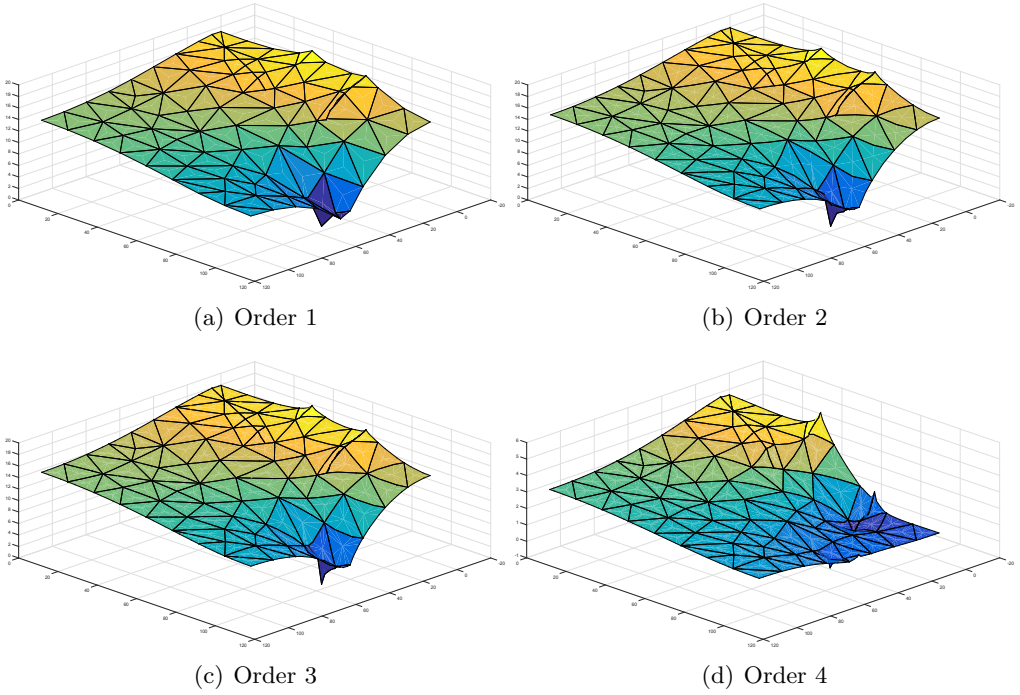
In Table 3.5.2, we report the behaviour of 2 preconditioning techniques. Different mesh parameters and VEM of order 1 and 2 are considered. The table displays the number of iterations required by the conjugate gradient (CG) routine compared to the performances of the preconditioned algorithm with the Lumped and Dirichlet preconditioners. For the non preconditioned CG algorithm, a rapid increase in the iteration number with mesh refinement can be appreciated for both orders 1 and 2. As expected, the increase in iterations with a preconditioner is much smaller, with the Dirichlet preconditioner performing better than the Lumped preconditioner.

The notable improvement renders almost imperative the use of a preconditioner, since the reduction in iteration number far outweighs the extra computational cost that arises from the computation of the preconditioner. Cases marked with NC stand for no convergence after 1 million iterations.



(a) Mesh (b) Detail

FIGURE 3.5.10: DFN116: detail of two very close and almost parallel traces



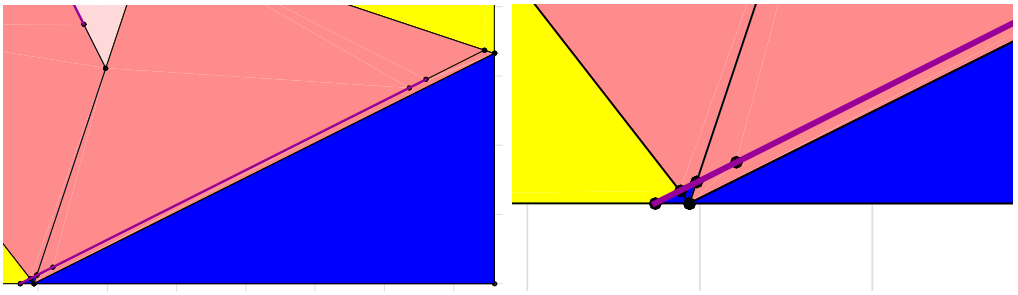
(a) Order 1 (b) Order 2

(c) Order 3 (d) Order 4

FIGURE 3.5.11: DFN116: comparison of results for problematic situations

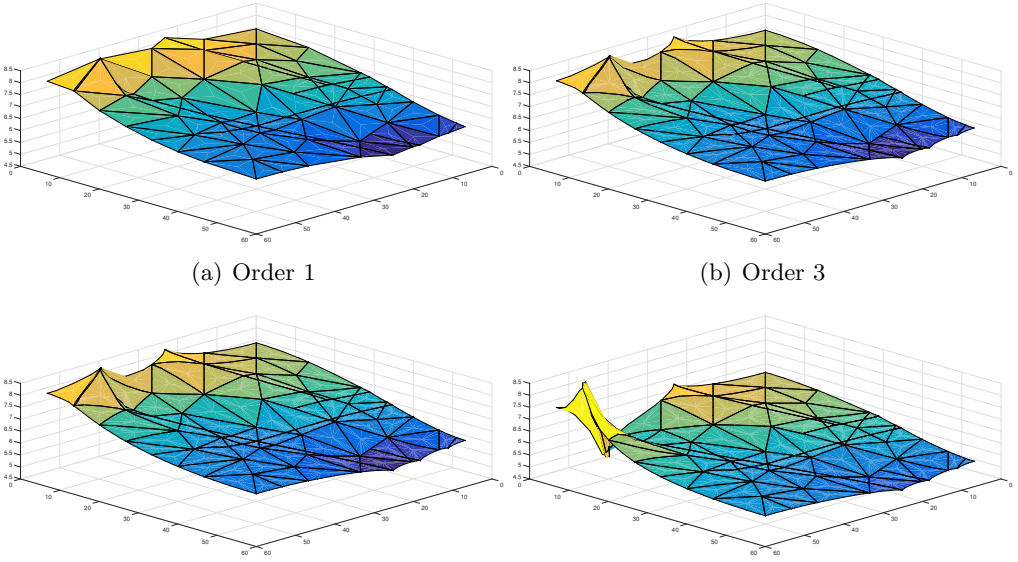
3.5.4 A survey of troublesome situations

In this subsection we describe some situations that arose in the simulations that have proven to be difficult to handle numerically. The monomial basis for the space of polynomials is notoriously bad conditioned, and the situation worsens with increasing orders. We believe that this is the cause of the issues we are presenting in this section, and they appear in elements with unsuitable shapes. Some of these issues can be prevented if a mesh modifying procedure as mentioned in Remark 3.2 is used.



(a) Mesh (b) Detail

FIGURE 3.5.12: DFN27: detail of an unfortunate disposition of a mesh edge and a trace



(a) Order 1 (b) Order 3

(c) Order 4 (d) Order 5

FIGURE 3.5.13: DFN27: comparison of results for problematic situations

A first example is related to the DFN with 116 fractures, where a fracture has two traces that are almost parallel and very close to each other, as in Figure 3.5.10. This inevitably leads to elements with a bad aspect ratio, since any attempt to obtain an adequate mesh would require a very large number of small elements to fill the space between the two traces. The solution is stable up to VEM of order 3, while when using a fourth order approximation the obtained solution drastically changes (see Figure 3.5.11), and even falls below zero, which is not compatible with the imposed boundary conditions, necessarily leading to a solution bounded between 0 and 100. As a reference, one particularly problematic mesh element has an almost rectangular shape and an area of 0.58, with a length of 10.26 in one direction and 0.058 in the other (a 177 ratio). This is a degenerate octagon and for order 4 it has 38 DOFs (Figure 3.5.10). We remark that this particular configuration can be successfully dealt with VEM of orders from 1 to 3, and problems only appear with order 4 and higher.

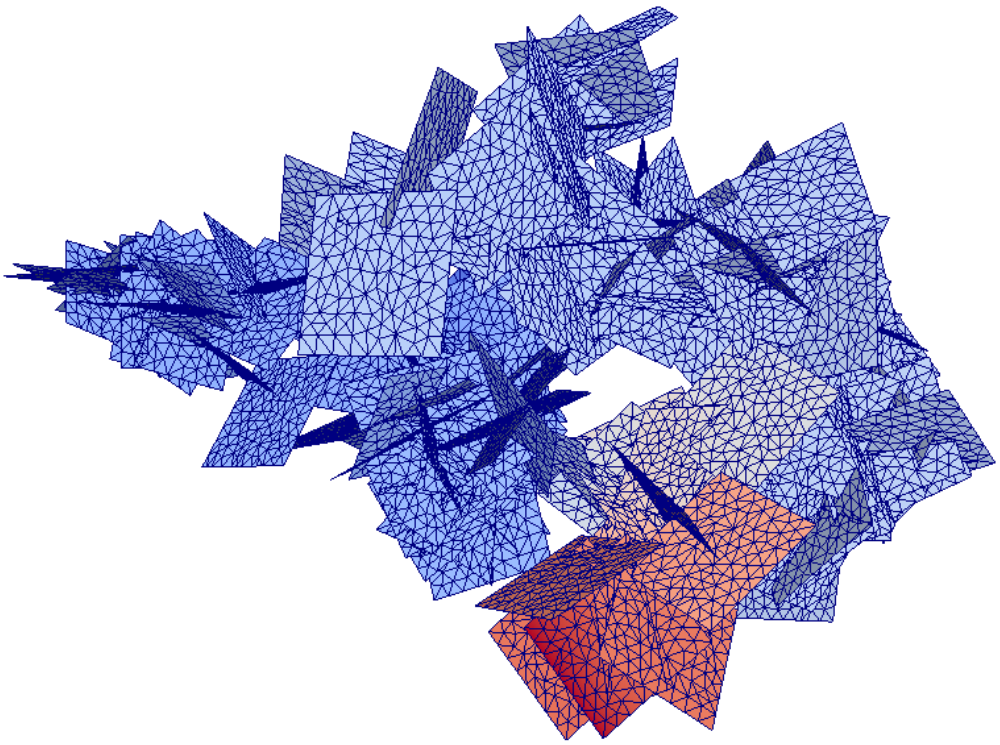
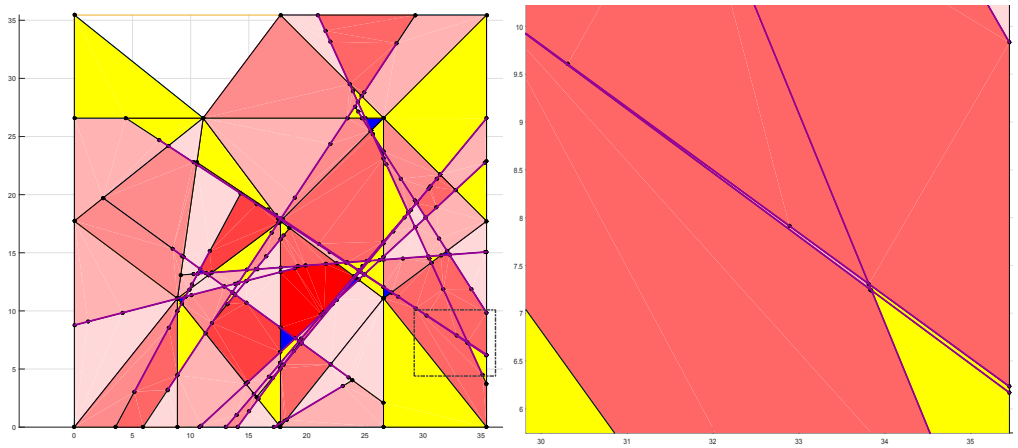


FIGURE 3.5.14: DFN130: spatial distribution of fractures for a DFN with 130 fractures

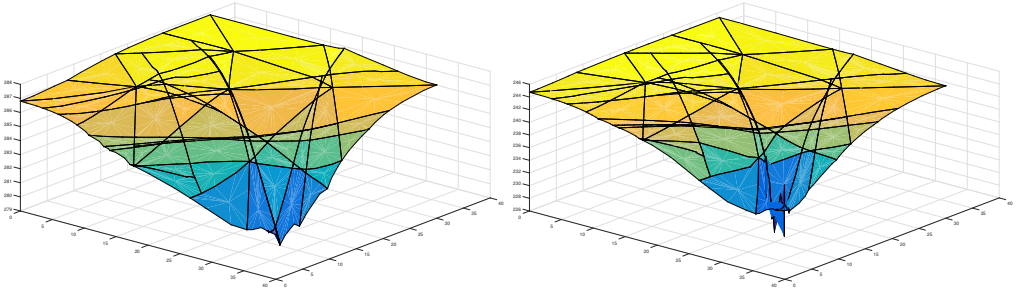
A second documented problematic configuration, occurring on the DFN 27 problem, concerns badly shaped elements due not to the geometry of the DFN but to an unfortunate starting mesh, and is such that it may not be present with either a finer or a coarser mesh. This situation could be prevented applying the mesh smoothing process described in Remark 3.2. The situation is depicted in Figure 3.5.12, where we can see that the edge of an element is very close to a trace and has originated elements much more stretched in one direction than in the other. Furthermore, a very small element was generated next to the stretched element. The solution for VEM of order 5 becomes numerically unstable in this case, as shown by Figure 3.5.13. We remark that the major source of instability in this case is again the elongated element and not the neighboring small element.

Finally, we present the last case that is part of a medium size DFN with 130 fractures, shown in Figure 3.5.14, that includes parallel traces very close to each other, large disparity between trace lengths, highly heterogeneous element areas, element angles of less than 1 degree and complex trace intersections among other complications. More precisely, we have for the whole DFN that: minimum angle $= 0.41°$, maximum trace length ≈ 45, minimum trace length ≈ 0.01 and largest number of traces in a fracture $= 24$. An adequate globally conforming triangular mesh for this system would be quite difficult to obtain, if not impossible. With our approach, meshing can be done as usual (Figure 3.5.15) although it may lead to elements with undesirable shapes. It can be



(a) Mesh (b) Detail

FIGURE 3.5.15: DFN130: detail of two traces meeting at a very small angle



(a) Order 3 (b) Order 4

FIGURE 3.5.16: DFN130: comparison of results for problematic situations

seen that irregularities in the solution were present only starting from VEM of order 4 approximations, again at a very elongated element between two traces meeting at a very small angle (Figure 3.5.16). The solution shows an uneven and rough behaviour that is further propagated to other fractures that have traces in common, and was not present in the solution obtained with the VEM of order 3.

3.6 Conclusions

In this work we have presented a novel method that constitutes a natural generalization of conforming Finite Elements for Discrete Fracture Network flow simulations. Local and global conformity is obtained using some of the features of the Virtual Element Method, and most importantly, global conformity is achieved without any constraints in the initial meshing process, that is performed independently for each fracture, nor any modification of DFN geometry. Convergence curves were presented as well as results for

DFNs of small and medium scale, and the method has been shown to be robust enough to handle complex geometrical situations that arise in randomly generated DFNs.

After extensive numerical experiments, the following patterns were noticed: in general, all methods give a good approximation for the hydraulic head H, and due to how the problem was implemented, continuity of H for the whole DFN is guaranteed. Even with VEM of order 1 the solutions are reliable for this variable, and this is due to the fact that we are using the primal formulation of the problem and the local conformity of the mesh allows for a more accurate representation of the jump of the derivative of H along the traces. In the case of the flux exchanged at the traces, U, the situation is different; only starting with a somewhat fine mesh can acceptable results be obtained for order 1. Order 2 on the other hand, shows a marked improvement that can be attributed to the larger number of DOF but also to the improved approximation of the gradient of H and consequently of U. We remark that U is not obtained directly, but deriving the projection onto a polynomial space of the computed primal variable H.

Concerning the use of discretizations with increasing polynomial accuracy, for this application, we discourage going beyond order 2 based on the obtained results. Higher orders are not only less stable numerically on strongly distorted meshes, but also much more computationally expensive, and the improvement in accuracy is often not considerable. In fact, the exact solution of a DFN does not have in general high regularity and a cubic approximation of H and a quadratic approximation for U might be excessive. As we have seen however, whenever regularity is guaranteed, convergence for higher orders is as good as expected.

Simple FETI algorithms for domain decomposition were successfully implemented and show promise for possible parallelization of the resulting linear system. They prove to be nearly indispensable if a large system is to be solved due to the achievable reduction in the number of iterations required to solve the system.

Finally, much of the work done here in obtaining the globally conforming meshes as well as the idea for imposing matching conditions between corresponding degrees of freedom can be readily applied with few alterations to an implementation of a mixed formulation of the original problem using mixed Virtual Elements and will the subject of future work.