Development and analysis of a novel optimization approach for the simulation of the flow in large scale discrete fracture networks with non-conforming finite elements / Sviluppo ed

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

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

Settore disciplinare: MATEMATICA - MAT/08

Tesi di Dottorato

Sviluppo ed analisi di un nuovo approccio per la simulazione del flusso in network discreti di fratture su larga scala basato su un metodo di ottimizzazione ed elementi finiti su griglie non conformi

Stefano Scialò

Tutore
Prof. Stefano Berrone

Coordinatore
Prof. Lamberto Rondoni

Marzo 2014

POLITECNICO DI TORINO

SCUOLA DI DOTTORATO

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

Thesis submitted for the degree of Doctor of Philosophy

Development and analysis of a novel optimization approach for the simulation of the flow in large scale discrete fracture networks with non-conforming finite elements

Stefano Scialò

Supervisor
Prof. Stefano Berrone

Coordinator
Prof. Lamberto Rondoni

March 2014

Dedicated to my family

Acknowledgements

The author is grateful to Stefano Berrone and Sandra Pieraccini for their constant support and valuable advices.

The author would also like to thank Alfio Borzì, Claudio Canuto, and Corrado Fidelibus for their remarks and suggestions.

Finally a special thank goes to my colleagues and friends Anna Cattani, Chiara Giverso and Moreno Ursino for many fruitful discussions, and to Matias Ferdinando Benedetto who contributed to the results in the last part of the present thesis.

Contents

Introduction

The objective of the work presented here is the description and the investigation of a novel numerical tool for the effective simulation of fluid flows in underground poro-fractured media at large scales.

This subject is of interest for several practical applications. In the context of an increasing concern towards environmental friendly industrial applications, sequestration and underground storage of CO_2 is currently under investigation as a potential way to reduce emissions of greenhouse gases. CO_2 is pumped in underground basins, where over geological time-scales it reacts with underground minerals forming stable carbonate mineral forms. Numerical models are a valuable tool for geologists and engineers to assess safety and viability of specific geological sites, in order to study the risk of dangerous leakages of gases through rock faults, or the dispersion of CO_2 in the atmosphere due to filtration through rock pores. CSS and a deeper research on the subject towards cost-efficiency and safety is currently promoted by European Commission, see Directive 2009/31/CE and the CSS website [7].

Countries that import natural gas for energetic purposes need to store huge quantities in order to face both typical fluctuations of request and unforeseeable long periods of scarce supply. Natural gas storage is usually performed in depleted geological reservoirs, or in large underground basins. Numerical tools can be used to assess the viability of geological sites, to predict seepage of fluid and the mechanical response of the rock faults to variations of pressure during the cyclical gas pumping in and out. As an example, Italy highly relies on imported gas for energy supply and currently has fourteen storage sites with a trend towards an increase of storing capacity ([12]).

The exploitation of an oil field requires a detailed assessment of soil properties and geological characteristics of the ground at reservoir scale. This is a lengthy and expensive process, requiring a large number of wells and soil analysis. Numerical codes are widely

used in this process, and more efficient algorithms are required in order to improve predictions and thus reduce costs.

The recent exploitation of shale gas in the United States requires horizontal drilling of the soil and the generation of fractures in the impermeable shale formations in order to extract the natural gas trapped therein. These techniques have a high environmental risk, linked to the intense use of water for the generation of fractures, the emission of large quantities of greenhouse gases in the atmosphere, the contamination of marine and underground water, and therefore require a careful management ([24]). Numerical models capable to combine the simulation of underground flows with rock mechanics and chemistry can be a valuable tool in this field.

Underground flow numerical simulations find other possible application in the management and monitoring of surface and sub-surface water resources or in the analysis of the transport and diffusion of pollutant species in the underground.

The present work focuses on the description of a new numerical model for the definition of the hydraulic head distribution in Discrete Fracture Networks (DFNs). Discrete fracture networks are a well established model to simulate hydrological processes in underground rock agglomerates, [14, 19, 8, 11, 4, 9, 15, 3]. A DFN consists of a set of intersecting planar polygons resembling the fractures in a rock matrix. The explicit representation of rock fractures is the major characteristic of these models, that are therefore preferred to continuum-like models when the fracture pattern represents the preferential flow path. This is the case when faults in the rock matrix have a higher permeability than the surrounding rocks. On the other hand, continuum models or hybrid continuum-fracture models are used when the sole fracture network is not sufficient to characterize the flow behaviour. In continuum models the flow is described as occurring in a continuous porous medium, in which the presence of fractures is accounted for the definition of a suitable permeability tensor ([16]).

Location, orientation, size and hydrological properties, such as the permeability tensor, of the fractures of a DFN resembling a specific geological site are defined by means of probability density functions, whose parameters are obtained through laboratory analyses on samples from probing or boreholes [2, 5, 1]. The quantity of interest is the hydraulic head in the fractures, representing the sum of the pressure head and of elevation. Hydraulic head is evaluated by means of the Darcy law and low order finite elements are usually employed to numerically solve the problem (see for example ROCKFLOW, [22]).

The classical approach described above has two major drawbacks that limit the use of DFN models for large scale applications. Firstly, DFNs of huge dimensions might count up to millions of fractures, thus requiring a very high computational effort, and additionally, repeated simulations are usually required to overcame the uncertainty due to the stochastic nature of input data. Secondly, the generation of a good quality mesh suitable for finite elements might result infeasible for intricate DFN configurations. This is connected to the fact that fractures in DFNs intersect with arbitrary orientation and the finite element triangulation need to be conformal to fracture intersections, usually called traces. As a consequence elongated elements with poor aspect ratio might be generated to match fractures intersecting with narrow angles, thus compromising the accuracy of the solution. In many cases, due to the conformity requirement, triangulation codes might even fail in generating a mesh [21].

The method described herein tackles both these difficulties by splitting the problem on the whole DFN in many small sub-problems on each fracture that can be solved independently from each other, and resorting to the minimization of a cost functional to enforce the compatibility conditions at fracture intersections. In such a way the complexity of the initial problem can be handled more efficiently in parallel computers in an easy and straightforward way, and the meshing process can be performed independently on each fracture, removing the constraint of triangulations conformal with fracture intersections.

Different discretization strategies are possible. The solution can be obtained using standard finite elements on each fracture, or through the use of special finite elements in order to improve the accuracy near the traces, where the solution is expected to have a discontinuous co-normal derivative and standard FEM on meshes non conforming to the traces would not correctly reproduce this non-smooth behaviour. Alternatives consist in using the eXtended Finite Element Method (XFEM) that allows a full non-conformity between mesh elements and traces and relies on additional basis functions to represent kinks in the solution, and the Virtual Element Method (VEM) that allows a partial non-conformity and an easy meshing procedure thanks to the use of elements with an arbitrary number of edges. Within the proposed approach a mixing of these discretization strategies is possible, improving flexibility in dealing with complex DFN configurations.

A large part of the research activity in the field of DFN simulations focuses on the problems identified above. In order to reduce problem complexity, in [6, 18] the authors

suggest to describe the DFN as a system of mono-dimensional pipes that connect the traces with the neighbouring fractures, without affecting the topology of the network. Different approaches rely on mortar methods to ease the meshing process allowing a partial non-conformity with the traces. In [23] mortar methods are used in conjunction with mixed finite elements, while in [20, 21] traces are modified to conform locally to element edges, but allowing nonconformity with the discretization on the intersecting fracture that is handled with mortar methods. Geometrical minor modification of the DFN are also proposed in other works, such as [13]. A different approach is proposed in [17], where the solution of 3D fracture networks is reduced to a system of differential problems on the traces, organized such that it is possible to obtain successive levels of approximations, according to the accuracy required. In [10] benchmark DFN configurations are provided and the authors envisage models with non-conforming meshes and a domain decomposition approach as a promising strategy for large scale simulations.

Overview

The present thesis has the structure of a collection of journal articles and is divided into three parts: the first part is devoted to the presentation of the mathematical statements of method, proposed both in the continuous and discrete formulations. Also the algorithm used to obtain a numerical solution is described, along with a large number of numerical results that show the viability and efficiency of the proposed method. The first part is constituted by Chapters 1-4 that report fully three published articles and a fourth work currently under review, co-authored by the author. In Chapter 1 is reproduced the following article:

Berrone S., Pieraccini S. and Scialò S., *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM Journal on Scientific Computing, 35(2), B487-B510.

In Chapter 2 is reproduced:

Berrone S., Pieraccini S. and Scialò S., *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM Journal on Scientific Computing, 35(2), A908-A935;

in Chapter 3:

Berrone S., Pieraccini S. and Scialò S, *An optimization approach for large scale simulations of discrete fracture network flows*, Journal of Computational Physics, 256, 838-853

and in Chapter 4:
Berrone S., Pieraccini S. and Scialò S, *The eXtended Finite Element Method for Subsurface Flow Simulations*, Under review.

The second part is constituted by unpublished material and is organized as follows. In Chapter 5, numerical results on complex DFN configurations are provided both with standard finite elements on nonconforming grids and with the XFEM on the same grids to improve solution representation. A preliminary investigation on the scalability properties of the algorithm end this Chapter. An analysis on a possible strategy of preconditioning the conjugate gradient method for DFN simulations is proposed in Chapter 6.

The third part is constituted by Chapter 7 that reports an article in preparation on a preliminary investigation of the method in conjunction with the Virtual Element Method as an alternative to the XFEM or FEM:
Benedetto, M., Berrone S., Pieraccini S. and Scialò S, *The Virtual Element Method for Discrete Fracture Network simulations*, In preparation.

Bibliography

[1] M. Abbie and J. Bedford, *Rock Mechanics: New Research*, Nova Science Publishers, Inc., 1999.

[2] P. M. Adler, *Fractures and Fracture Networks*, Kluwer Academic, Dordrecht, 1999.

[3] A. Baghbanan and L. R. Jing, *Hydraulic properties of fractured rock masses with correlated fracture length and aperture*, Int. J. Roch Mech. Min., 44 (2007), pp. 704–719.

[4] I. I. Bogdanov, V. V. Mourzenko, J. F. Thovert, and P. M. Adler, *Effective permeability of fractured porous media in steady state flow*, Water Resour. Res., 39 (2003), pp. 1023–1038.

[5] O. Bour, P. Davy, C. Darcel, and N. Odling, *A statistical scaling model for fracture network geometry, with validation on a multiscale mapping of a joint network (Hornelen Basin, Norway)*, J. Geophys. Res., 107 (2002).

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

[7] *Carbon Capture and Geological Storage - European Community.* `http://ec.europa.eu/clima/policies/lowcarbon/ccs/index_en.htm`, Septmber 2013.

[8] M. Cravero and C. Fidelibus, *A code for scaled flow simulations on generated fracture networks*, Comput. Geosci., 25 (1999), pp. 191–195.

[9] V. Cvetkovic, S. Painter, N. Outters, and J. O. Selroos, *Stochastic simulation of radionuclide migration in discretely fractured rock near the Aspo Hard Rock Laboratory*, Water Resour. Res., 40 (2004), p. W02404.

[10] J.-R. de Dreuzy, G. Pichot, B. Poirriez, and J. Erhel, *Synthetic benchmark for modeling flow in 3D fractured media*, Computers & Geosciences, 50 (2013), pp. 59 – 71.

[11] J. R. D. Dreuzy, P. Davy, and O. Bour, *Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 2., permeability of networks based on log-normal distribution of apertures*, Water Resour. Res., 37 (2001), pp. 2079–2095.

[12] *Elenco dei pozzi e delle centrali si stoccaggio - Ministero dello Sviluppo Economico, Italy.* `http://unmig.sviluppoeconomico.gov.it/unmig/stoccaggio/pozzi/centrali.asp`, September 2013.

[13] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[14] J. C. S. Long, J. S. Remer, C. R. Wilson, and P. A. Witherspoon, *Porous media equivalents for networks of discontinuous fractures*, Water Resour. Res., 23 (1982), pp. 645–658.

[15] L. Martinez-Landa and J. Carrera, *A methodology to interpret cross-hole tests in a granite block*, J. Hydrol., 325 (2006), pp. 222–240.

[16] S. P. Neuman, *Trends, prospects and challenges in quantifying flow and transport through fractured rocks*, Hydrogeol. J., 13 (2005), pp. 124–147.

[17] B. Nœtinger and N. Jarrige, *A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks*, J. Comput. Phys., 231 (2012), pp. 23–38.

[18] A. W. Nordqvist, Y. W. Tsang, C. F. Tsang, B. Dverstop, and J. Andersson, *A variable aperture fracture network model for flow and transport in fractured rocks*, Water Resource Res., 28 (1992), pp. 1703–1713.

[19] A. W. Nordqvist, Y. W. Tsang, C.-F. Tsang, B. Dverstorp, and J. Andersson, *Effects of high variance of fracture transmissivity on transport and sorption at different scales in a discrete model for fractured rocks*, J. Contam. Hydrol., 22 (1996), pp. 39–66.

[20] G. Pichot, J. Erhel, and J.-R. De Dreuzy, *A mixed hybrid Mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629–1643.

[21] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM J. Sci. Comput., 34 (2012), pp. B86–B105.

[22] *Rockflow*. `http://www.rockflow.uni-hannover.de`, September 2013.

[23] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.

[24] Q. Wang, X. Chen, A. N. Jha, and H. Rogers, *Natural gas from shale formation - the evolution, evidences and challenges of shale gas revolution in united states*, Renewable and Sustainable Energy Reviews, 30 (2014), pp. 1 – 28.

Part I

Presentation of the method

Chapter 1

A PDE-constrained optimization formulation for discrete fracture network flows

Abstract We investigate a new numerical approach for the computation of the 3D flow in a discrete fracture network that does not require a conforming discretization of partial differential equations on complex 3D systems of planar fractures. The discretization within each fracture is performed independently of the discretization of the other fractures and of their intersections. Independent meshing process within each fracture is a very important issue for practical large scale simulations making easier mesh generation. Some numerical simulations are given to show the viability of the method. The resulting approach can be naturally parallelized for dealing with systems with a huge number of fractures.

1.1 Introduction

Efficient numerical simulations of subsurface fluid flows in fractured rocks are of interest for many applications ranging from water resources management, contaminant transport and dissemination, oil prospecting and enhanced oil/gas recovery. Among the major difficulties are intrinsic heterogeneity, directionality of the medium and multiscale nature of the phenomena, as well as uncertainty in the medium properties. A Discrete Fracture Network (DFN) is a complex 3D structure obtained intersecting planar frac-

tures. DFN models are frequently preferred to more conventional continuum models as basis for simulations. A classical approach to the problem is to model fractures as planar ellipses or polygons and stochastically generate DFNs with probabilistic distributions of density, aspect ratio, orientation, size, aperture of fractures and hydrologic properties [9] and to simulate the flow through the obtained networks. Intensive numerical simulations with several configurations of DFNs and physical parameters are then performed in order to tackle the issue of uncertainty. The flow pattern strongly depends on density and size of fractures and for large scale simulations different approaches are possible. For dense fracture networks and continuous distribution of size and aspect ratios, flow can be modeled as the flow in an equivalent continuous porous medium where the fracture network pattern leads to the definition of a suitable permeability tensor. For sparse fracture networks with some large fractures that discontinuously increase directionality of the flow, an explicit representation of the fracture network is more reliable. In both cases a stochastic approach to the uncertainty of the parameters is needed and this requires many simulations, so that efficiency and large applicability of numerical algorithms are fundamental issues.

Here the steady flow in a given DFN is considered assuming the rock matrix impervious and no longitudinal flow in the intersection between the fractures. These intersections are called *traces* and are always segments.

In DFN simulations the first classical numerical challenge is to provide a good-quality conforming mesh for this 3D structure to be used for the discretization of the flow equations. Conformity of the mesh requires that for each trace a unique discretization is introduced, which is shared by all the discretizations of the fractures intersecting along the trace. Conformity on the traces and good quality of the meshes for a completely arbitrary DFN can be obtained only with the introduction of a huge number of elements independently of the required accuracy of the numerical solution. In [28], a mixed non-conforming finite element method on a conforming mesh is proposed. In [20], an adaptive approach to the conforming mesh generation requiring adjustments of trace spatial collocations is proposed. Local modifications of the mesh or of the fracture network in order to preserve conformity of the meshes or alignment of meshes along the traces are considered in several works as [17, 28]. In [11], a method to generate a good-quality conforming mesh on the network system is proposed. In [23, 24], a mixed hybrid mortar method is proposed allowing nonconformities of the meshes on the fractures, but requiring that the traces are contained in the set of the edges of

each fracture triangulation. Resorting to mortar methods the discretization of each fracture can lead to a different discretization of the traces. A different approach to the simulation of the flow in the fracture network is based on its modelization with a system of mono-dimensional pipes that are aligned along the fractures and mutually connect the centers of the fracture intersections with the surrounding fractures. The resulting mesh of pipes still reflects the topological properties of the fracture network [6, 22]. An accurate definition of pipe properties within the fracture system has been obtained by means of a boundary element method in [10]. However, the geometrical simplification implies errors in the assessment of the fluid flow regime, depending on the complexity and geometrical properties of the underlying DFN, thus the resort to a full discretization is preferred.

Specific commercial codes based on FEM are available, also simulating the fluid flow in the rock blocks [19]; contributions can be found in literature for the extension to coupled problems with deformable blocks and fractures, even in conjunction with other methods as BEM (e.g. [12]). However, these codes suffer for a strong computational demand: the discretization in fact leads frequently to the generation of huge or poor-quality meshes.

Problem model allows discontinuities of fluxes of hydraulic head through the traces when fluxes of hydraulic head leave a fracture to reach a different fracture at the common trace. In the previous approaches these discontinuities can be modeled if they are localized at edges between elements or at the border of each piece of fracture.

In this paper a new method is proposed, which relies on the reformulation of the problem as a PDE-constrained optimization problem. Following this approach, fracture meshes are not required to match along traces and any kind of mesh conformity along traces is skipped, thus making the mesh generation process an easy task, attainable with a standard mesh generator. Furthermore, the problem on the overall DFN can be decoupled in several local problems on the fractures, thus allowing a great potential for a possible parallel implementation. Discontinuities of fluxes of hydraulic head can occur on arbitrary traces with respect to the triangulation and the used finite elements allow to catch these discontinuities of the fluxes also inside elements. This can be obtained introducing suitable Extended Finite Elements (XFE).

The paper is organized as follows. In Section 1.2, we recall the physical model and governing equations, and introduce the continuous optimization problem that leads to the solution on the network system. In Section 1.3 we recall basics on extended

finite elements of the type considered herein, and give some details for the application to DFNs. In Section 1.4 a discrete formulation of the optimization problem is given, which leads to an equality constrained Quadratic Programming problem. Finally, in Section 1.5 numerical results are discussed in order to prove the viability, reliability and effectivity of the method.

Notations. In the paper, we will frequently use the following notations. We will use capital letters for continuous unknowns (as for example the hydraulic head H) and lower case letters for the corresponding finite dimensional approximation (e.g. h). We will use the same lower case letter for the vector of degrees of freedom, the difference being clear from the context. Roman capital letters will be used for functional spaces. Given functions g_i, for i belonging to some index set I, the symbol $\prod_{i\in I} g_i$ denotes the tuple of functions $(g_1, g_2, ..., g_{\#I})$, being $\#I$ the cardinality of I.

1.2 Description of the problem

1.2.1 The continuous problem

Let us consider an open planar polygonal fracture $\omega \subset \mathbb{R}^2$ and let us introduce on ω a tangential coordinate system $\hat{x}$. Following [1], the problem of subsurface flow through ω can be written as:

$$-\nabla \cdot (\mathbf{K}\,\nabla H) = q \qquad \text{in } \omega, \tag{1.1}$$

$$H_{|\Gamma_D} = H_D \qquad \text{on } \gamma_D, \tag{1.2}$$

$$\frac{\partial H}{\partial \hat{\nu}} = G_N \qquad \text{on } \gamma_N, \tag{1.3}$$

where $\partial\omega = \gamma_D \cup \gamma_N$ is the boundary of ω and $\gamma_D \cap \gamma_N = \emptyset$, $\gamma_D \neq \emptyset$. The scalar function $H = \mathcal{P} + \zeta$ is the hydraulic head, $\mathcal{P} = p/(\varrho g)$ is the pressure head, p is the fluid pressure, g is the gravitational acceleration and ϱ is the fluid density. The variable ζ is the elevation, and $\mathbf{K} = \mathbf{K}(\hat{x})$ is the fracture transmissivity tensor and is a symmetric and uniformly positive definite tensor. The symbol $\frac{\partial H}{\partial \hat{\nu}}$ denotes the outward co-normal derivative of the hydraulic head:

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

with $\hat{n}$ unit vector outward normal to the boundary γ_N.

The definition of the hydraulic head in a Discrete Fracture Network Ω should require the solution of problem (1.1)-(1.3) in a system of intersected polygonal fractures in the space. In order to define 3D fractures F_i, let us consider a set of open planar polygons $\{\omega_i\}_{i\in\mathfrak{I}}$, being $\mathfrak{I}$ the set of their indices, and let $\bar{F}_i \subset \mathbb{R}^3$ be the image of the closure of a polygon $\omega_i \subset \mathbb{R}^2$ through an affine mapping $T_i(\hat{x}_i) = b_i + Q_i(\hat{x}_i - \hat{x}_{0,i})$ where $\hat{x}_{0,i}$ is the coordinate of a given vertex of the polygon ω_i in the local planar reference system $\hat{x}_i$, and b_i is the position of the same vertex in the 3D space. We assume that $Q_i^T Q_i$ is the identity matrix, such that the differential operators defined on the tangential reference system in F_i are equivalent to the operators defined on the planar fracture ω_i. Let Ω be the 3D set

$$\Omega = \bigcup_{i\in\mathfrak{I}} F_i,$$

and let $\partial\Omega$ denote its boundary. Given two fractures, the intersection of their closure is either an empty set or a set of non vanishing segments called *traces* (vanishing segments are not considered as no flux exchange among fractures takes place in these intersections). Let $\mathcal{S}$ denote the set of all the traces, and assume traces in $\mathcal{S}$ are indexed by a set of indices $\mathfrak{M}$, with cardinality $\sharp\mathfrak{M}$.

In the sequel, we make the following assumptions on the DFN:

1. $\bar{\Omega}$ is a connected set;

2. each trace S_m, $m \in \mathfrak{M}$, is shared by exactly two polygonal fractures F_i and F_j, $i \neq j$: $S_m \subseteq \bar{F}_i \cap \bar{F}_j$;

3. on each fracture, the transmissivity tensor $\mathbf{K}_i(\hat{x}_i)$ is symmetric and uniformly positive definite.

Given a trace S_m we denote by $I_{S_m} = \{i, j\}$ the set of indices i and j of the fractures F_i and F_j sharing the trace; for further convenience, we also introduce the sorted couple $c_m = (i, j)$ with $i < j$. For each fracture F_i, we denote by $\mathcal{S}_i$ the set of traces shared by F_i and other fractures.

In order to define the problem on the DFN, let us consider a set of open subfractures $f_l,\ l \in \mathfrak{L}$, obtained splitting each fracture in such a way that each trace is part of the boundary of some subfractures and $S_m \cap f_l = \emptyset,\ \forall m \in \mathfrak{M}, \forall l \in \mathfrak{L}$, see Figure 1.1. Note that the traces belong to the boundary of the subfractures, but they do not necessarily coincide with a whole edge of such boundaries, see e.g. trace S_2 in Figure 1.1. So we

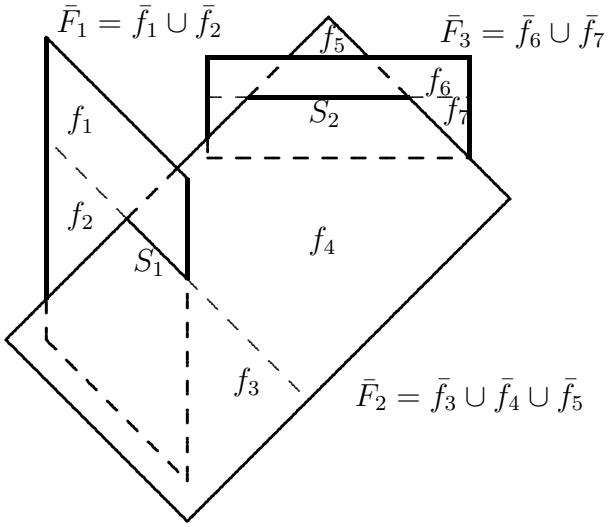


Figure 1.1: An example of DFN splitted in subfractures

have

$$\Omega = \bigcup_{l \in \mathfrak{L}} \bar{f}_l \backslash \partial\Omega.$$

Let us split $\partial\Omega$ in two parts $\Gamma_D \neq \emptyset$ and Γ_N, with $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\Gamma_D \cap \Gamma_N = \emptyset$, corresponding to Dirichlet and Neumann boundary conditions, respectively.

The global hydraulic head H in the whole connected system Ω satisfies the following equations $\forall l \in \mathfrak{L}$:

$$\nabla \cdot (\mathbf{K}_{f_l} \nabla H) = q_l, \qquad \text{in } f_l, \tag{1.4}$$

$$H_{|_{\Gamma_D \cap \partial f_l}} = H_D, \qquad \text{on } \Gamma_D \cap \partial f_l, \tag{1.5}$$

$$\frac{\partial H}{\partial \hat{\nu}_{\partial f_l}} = G_N, \qquad on\ \Gamma_N \cap \partial f_l, \tag{1.6}$$

with a 2D local reference system on f_l. Given a trace S_m let $\mathfrak{L}_{S_m} \subset \mathfrak{L}$ be the set of indices l such that $S_m \subset \partial f_l$. Equations (1.4)-(1.6) have to be complemented with the following coupling conditions, corresponding to the physical requirement of continuity of the hydraulic head and conservation of hydraulic fluxes across the traces:

$$H_{|_{\bar{f}_l}} = H_{|_{\bar{f}_k}}, \ \text{on } S_m, \qquad \forall S_m \in \mathcal{S},\ \forall l, k \in \mathfrak{L}_{S_m}, \tag{1.7}$$

$$\sum_{l \in \mathfrak{L}_{S_m}} \frac{\partial H_{|_{f_l}}}{\partial \hat{\nu}_{\partial f_l}} = 0, \qquad on\ S_m, \qquad \forall S_m \in \mathcal{S}\,. \tag{1.8}$$

For this formulation of the problem existence and uniqueness of the solution are known. In the following we want to focus on the whole fracture, disregarding this subfracture approach. Thus, let us denote by H_i the restriction of the hydraulic head H to the fracture F_i, $\forall i \in \mathfrak{I}$. Conditions (1.7) and (1.8) are equivalent to

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i,j \in I_{S_m},\ \forall m \in \mathfrak{M}, \tag{1.9}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_{S_m}}\right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_{S_m}}\right]\!\right]_{S_m} = 0, \qquad \text{for } i,j \in I_{S_m}, \tag{1.10}$$

where the symbol $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_{S_m}}\right]\!\right]_{S_m}$ denotes the jump of the co-normal derivative along the unique normal $\hat{n}^i_{S_m}$ fixed for the trace S_m on the fracture F_i. This jump is independent of the orientation of $\hat{n}^i_{S_m}$.

Let Γ_i be the boundary of F_i and let it be split in Γ_{iN}, the boundary with Neumann boundary condition $\frac{\partial H_i}{\partial \hat{\nu}} = G_{iN}$, and $\Gamma_{iD} \neq \emptyset$, the boundary with Dirichlet boundary condition $H_{i|_{\Gamma_D}} = H_{iD}$, satisfying $\Gamma_{iN} \cap \Gamma_{iD} = \emptyset$ and $\Gamma_{iN} \cup \Gamma_{iD} = \Gamma_i$. Let us define

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

and V_i' its dual space. The hydraulic head H_i in each fracture belongs to the space

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

and the hydraulic head H on the whole domain Ω is obtained by suitably matching via (1.9), (1.10) for $m \in \mathfrak{M}$ the solutions $H_i \in V_i^D$ for each $i \in \mathfrak{I}$, and belongs to the space

$$V^D = \mathrm{H}^1_{\mathrm{D}}(\Omega) = \left\{ v \in \prod_{i \in \mathfrak{I}} V_i^D : (v_{|_{F_i}})_{|_{S_m}} = (v_{|_{F_j}})_{|_{S_m}},\ i,j \in I_{S_m},\ \forall m \in \mathfrak{M} \right\}. \tag{1.11}$$

With a similar definition we set $V = \mathrm{H}^1_0(\Omega)$.

For the sake of simplicity of notation, in the following of this section we assume that the traces $S \in \mathcal{S}$ are disjoint.

Remark 1.1. The assumption of disjoint traces can be removed by replacing, in the sequel, each single trace S with the union of connected traces. Furthermore, in our discrete formulation, this assumption is dropped out in a natural way, see later Remark 1.2.

Let us define for each trace $S \in \mathcal{S}$ a suitable space $\mathcal{U}^S$ and

$$\mathcal{U}^{\mathcal{S}_i} = \prod_{S \in \mathcal{S}_i} \mathcal{U}^S, \qquad \mathcal{U} = \prod_{i \in \mathfrak{I}} \mathcal{U}^{\mathcal{S}_i}.$$

Moreover, for each trace $S \in \mathcal{S}$, with $I_S = \{i, j\}$, we introduce suitable variables $U_i^S \in \mathcal{U}^S$ and $U_j^S \in \mathcal{U}^S$ representing the unknown quantities $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i}\right]\!\right]_S$ and $\left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}_S^j}\right]\!\right]_S$, respectively, and for each fracture F_i let us set

$$U_i = \prod_{S \in \mathcal{S}_i} U_i^S \in \mathcal{U}^{\mathcal{S}_i}$$

i.e., U_i is the tuple of functions U_i^S with S spanning $\mathcal{S}_i$. Moreover, we set

$$U = \prod_{i \in \mathfrak{I}} U_i \in \mathcal{U}$$

as the tuple of all functions U_i^S with $S \in \mathcal{S}_i$ and $i \in \mathfrak{I}$, i.e. U is the $2\#\mathfrak{M}$-tuple of functions U_i^S on all traces in Ω.

Condition (1.10) rewrites, in terms of the new unknowns $U_i^{S_m}$, $U_j^{S_m}$ as

$$U_i^{S_m} + U_j^{S_m} = 0, \qquad \text{for } i, j \in I_{S_m}. \tag{1.12}$$

Let us introduce the following linear bounded operators and their duals:

$$\begin{gathered} A_i \in \mathcal{L}(V_i, V_i'), \qquad A_i^* \in \mathcal{L}(V_i, V_i'), \qquad A_i^D \in \mathcal{L}(V_i^D, V_i'), \\ B_i \in \mathcal{L}(\mathcal{U}^{\mathcal{S}_i}, V_i'), \qquad B_i^* \in \mathcal{L}(V_i, \mathcal{U}^{\mathcal{S}_i\prime}), \qquad B_{\Gamma_{iN}} \in \mathcal{L}(\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), V_i'), \end{gathered}$$

and the Riesz isomorphism $\Lambda_{\mathcal{U}^{\mathcal{S}_i}} : \mathcal{U}^{\mathcal{S}_i} \to \mathcal{U}^{\mathcal{S}_i\prime}$. The operators A_i, A_i^D, B_i, $B_{\Gamma_{iN}}$ are defined such that

$$\begin{aligned} \langle A_i H_i^0, v\rangle_{V_i', V_i} &= \left(\mathbf{K}\,\nabla H_i^0, \nabla v\right), \ H_i^0 \in V_i, \ v \in V_i, \\ \langle A_i^D H_i^D, v\rangle_{V_i', V_i} &= (\mathbf{K}\,\nabla H_i^D, \nabla v), \ H_i^D \in V_i^D, \ v \in V_i, \\ \langle B_i U_i, v\rangle_{V_i', V_i} &= \langle U_i, v_{|\mathcal{S}_i}\rangle_{\mathcal{U}^{\mathcal{S}_i}, \mathcal{U}^{\mathcal{S}_i\prime}}, \ U_i \in \mathcal{U}^{\mathcal{S}_i}, v \in V_i, \\ \langle B_{\Gamma_{iN}} G_{iN}, v\rangle_{V_i', V_i} &= \langle G_{iN}, v_{|\Gamma_{iN}}\rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})}, \ G_{iN} \in \mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), v \in V_i. \end{aligned}$$

Finally, let $\mathcal{R}_i\, H_{iD} \in V_i^D$ be a lifting of Dirichlet boundary condition H_{iD}.

Let us introduce $\forall i \in \mathfrak{I}$ the problem: *find* $H_i = H_i^0 + \mathcal{R}_i\, H_{iD}$, *with* $H_i^0 \in V_i$ *such that:*

$$\begin{aligned} \left(\mathbf{K}\,\nabla H_i^0, \nabla v\right) \;=\; & (q_i, v) + \langle U_i, v_{|\mathcal{S}_i}\rangle_{\mathcal{U}^{\mathcal{S}_i}, \mathcal{U}^{\mathcal{S}_i\prime}} \\ & + \langle G_{iN}, v_{|\Gamma_{iN}}\rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} - \left(\mathbf{K}\,\nabla\, \mathcal{R}_i\, H_{iD}, \nabla v\right), \quad \forall v \in V_i \end{aligned} \tag{1.13}$$

or equivalently $\forall i \in \mathfrak{I}$

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_{iN} - A_i^D\, \mathcal{R}_i\, H_{iD}. \tag{1.14}$$

The following result states the equivalence between the subfracture setting and the setting based on fractures. The proof is omitted as it straightforwardly follows from classical arguments.

Proposition 1.1. *Let* $\mathcal{U}^S = H^{-\frac{1}{2}}(S), \forall S \in \mathcal{S}$. *Then, solving* (1.13) $\forall i \in \mathfrak{I}$ *with additional conditions* (1.9), (1.12) *is equivalent to solve* (1.4)-(1.8).

1.2.2 The optimal control formulation

The formulations of the problem described in the previous section requires the exact fullfilment of some conditions which couple the solution on different fractures; this happens either in the subfracture setting given by equations (1.4)-(1.8), or with the formulation (1.13) with coupling conditions (1.9), (1.12). Hence, finding a numerical solution to the problem solving the previous sets of equations typically asks for some form of (at least partial) conformity in the meshes introduced on the fractures, see e.g. [11, 17, 20, 23, 28].

In order to circumvent this problem, we propose here a different approach. Instead of solving the overmentioned coupled differential problems, we look for the solution of a PDE constrained optimal control problem [18], the variable U being the "control variable". Let us define for each trace $S \in \mathcal{S}$ a suitable space $\mathcal{H}^S$, the spaces

$$\mathcal{H}^{\mathcal{S}_i} = \prod_{S \in \mathcal{S}_i} \mathcal{H}^S, \qquad \mathcal{H} = \prod_{i \in \mathfrak{I}} \mathcal{H}^{\mathcal{S}_i},$$

and the Riesz isomorphism $\Lambda_{\mathcal{H}^{\mathcal{S}_i}} : \mathcal{H}^{\mathcal{S}_i} \to \mathcal{H}^{\mathcal{S}_i\prime}$. The following linear bounded "observation" operators C_i^S and C_i and the dual $C_i{}^*$

$$C_i^S \in \mathcal{L}(V_i, \mathcal{H}^S), \qquad C_i \in \mathcal{L}(V_i, \mathcal{H}^{\mathcal{S}_i}) = \prod_{S \in \mathcal{S}_i} C_i^S, \qquad C_i{}^* \in \mathcal{L}(\mathcal{H}^{\mathcal{S}_i\prime}, V_i'),$$

will be defined for each choice of the spaces $\mathcal{H}^S$. For all $i \in \mathfrak{I}$, let us denote by $H_i(U_i)$ the solution to (1.13) corresponding to the value U_i for the control variable. Furthermore, fixed a fracture F_i, we denote by

$$\prod_{S \in \mathcal{S}_i} U_j^S$$

the tuple of control variables defined on fractures F_j intersecting F_i in traces $S \in \mathcal{S}_i$ and by

$$\prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right)$$

the tuple of functions $\left(C_i^S H_i(U_i) - C_j^S H_j(U_j)\right)$ as S varies in $\mathcal{S}_i$.

Let us now introduce the following differentiable functional $J : \mathcal{U} \to \mathbb{R}$:

$$\begin{aligned} J(U) &= \sum_{S\in\mathcal{S}} J^S(U) = \sum_{S\in\mathcal{S}} \left(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{H}^S} + ||U_i^S + U_j^S||^2_{\mathcal{U}^S} \right) \\ &= \frac{1}{2} \sum_{i\in\mathfrak{I}} \sum_{S\in\mathcal{S}_i} \left(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{H}^S} + ||U_i^S + U_j^S||^2_{\mathcal{U}^S} \right) \\ &= \frac{1}{2} \sum_{i\in\mathfrak{I}} \left(|| \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right) ||^2_{\mathcal{H}^{\mathcal{S}_i}} + ||U_i + \prod_{S\in\mathcal{S}_i} U_j^S||^2_{\mathcal{U}^{\mathcal{S}_i}} \right). \end{aligned} \tag{1.15}$$

Proposition 1.2. *Let us define the spaces $\mathcal{U}^S$ and $\mathcal{H}^S$ and the observation operator C_i^S on the trace S as*

$$\mathcal{U}^S = H^{-\frac{1}{2}}(S), \qquad \mathcal{H}^S = H^{\frac{1}{2}}(S), \qquad C_i^S H_i = H_{i|_S}, \ \forall S \in \mathcal{S}\,. \tag{1.16}$$

Then, the hydraulic head $H \in H^1_D(\Omega)$ is the unique exact solution of (1.4)-(1.8) if and only if it satisfies the differential problems (1.13) for all $i \in \mathfrak{I}$ and, correspondingly, $J(U) = 0$.

Proof. The existence and uniqueness of $H \in \mathrm{H}^1_{\mathrm{D}}(\Omega)$ satisfying (1.4)-(1.8) is a classical result (see for example [28] and references therein). Proposition 1.1 states that problems (1.4)-(1.8) $\forall l$ are equivalent to problems (1.13) $\forall i$, endowed with matching conditions (1.9)-(1.12), which in turn are equivalent to $J(U) = 0$. □

Based on the previous Proposition, the problem of finding the hydraulic head in the whole domain is restated here as follows: find $U \in \mathcal{U}$ solving the problem

$$\min J(U) \quad \text{subject to (1.13)}, \ \forall i \in \mathfrak{I}. \tag{1.17}$$

Proposition 1.3. *The optimal control $U \in \mathcal{U}$ providing the solution to (1.17) corresponds to*

$$\left(\Lambda_{\mathcal{U}^{\mathcal{S}_i}}\right)^{-1} {B_i}^* P_i + U_i + \prod_{S\in\mathcal{S}_i} U_j^S = 0, \qquad \forall i \in \mathfrak{I} \tag{1.18}$$

where the functions $P_i \in V_i$, $\forall i \in \mathfrak{I}$ are the solutions to the equations

$$A_i^* P_i = {C_i}^* \Lambda_{\mathcal{H}^{\mathcal{S}_i}} \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i - C_j^S H_j \right). \tag{1.19}$$

Proof. Let us differentiate the cost functional $J(U)$ with respect to the control U_i, this has effect only for $S \in \mathcal{S}_i$ and we have

$$
\begin{aligned}
&J'(U)(v_i - U_i) = \sum_{S\in\mathcal{S}_i} J^{S\prime}(U_i)(v_i - U_i) \\
&= \sum_{S\in\mathcal{S}_i} \left[2\left(C_i^S H_i(U_i) - C_j^S H_j(U_j), C_i^S(H_i(v_i) - H_i(U_i))\right)_{\mathcal{H}^S} + 2\left(U_i^S + U_j^S, v_i^S - U_i^S\right)_{\mathcal{U}^S} \right] \\
&= 2\left\langle C_i{}^*\Lambda_{\mathcal{H}^{\mathcal{S}_i}} \prod_{S\in\mathcal{S}_i} (C_i^S H_i(U_i) - C_j^S H_j(U_j)), H_i(v_i) - H_i(U_i) \right\rangle_{V_i', V_i} \\
&\qquad +2\left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}(U_i + \prod_{S\in\mathcal{S}_i} U_j^S), v_i - U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime}, \mathcal{U}^{\mathcal{S}_i}} \\
&= 2\left\langle A_i^* P_i, A_i^{-1} B_i (v_i - U_i) \right\rangle_{V_i', V_i} + 2\left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}(U_i + \prod_{S\in\mathcal{S}_i} U_j^S), v_i - U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime}, \mathcal{U}^{\mathcal{S}_i}} \\
&= 2\left\langle B_i^* P_i, v_i - U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime}, \mathcal{U}^{\mathcal{S}_i}} + 2\left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}(U_i + \prod_{S\in\mathcal{S}_i} U_j^S), v_i - U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime}, \mathcal{U}^{\mathcal{S}_i}}
\end{aligned}
$$

and this yields the thesis. □

Equations (1.13), (1.18) and (1.19) $\forall i \in \mathfrak{I}$ then provide solution to the subsurface flow in the network; nevertheless, they couple all the unknowns on the overall DFN. As an alternative approach, we propose to set up a minimization process that only requires, at each step, local solutions on the fractures. The key point of this approach is that the method only requires *decoupled* solutions of the flows on fractures, thus avoiding mesh conformity requirements. This target is attained, for example, by using a gradient-based approach, such as for example the steepest descent method. This approach requires the solution of many simple problems with a small exchange of data. The resulting algorithm is suitable for massively parallel computers and GPU-based computers.

In order to describe the minimization process leading to the solution of the continuous problem (1.17), let us define

$$
\delta U_i = \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} B_i{}^* P_i + U_i + \prod_{S\in\mathcal{S}_i} U_j^S, \ \forall i \in \mathfrak{I}, \qquad \delta U = \prod_{i\in\mathfrak{I}} \delta U_i \tag{1.20}
$$

and let $\delta H_i \in V_i$, $\forall i \in \mathfrak{I}$ be defined as the solution of the problem

$$
A_i \delta H_i = B_i \delta U_i. \tag{1.21}
$$

Proposition 1.4. *Given a control variable U, let us increment it by a step $\lambda\delta U$. The*

steepest descent method corresponds to the stepsize

$$\lambda = -\frac{||\delta U||_{\mathcal{U}}^2}{\sum_{S\in\mathcal{S}} \left(||C_i^S \delta H_i - C_j^S \delta H_j||_{\mathcal{H}^S}^2 + ||\delta U_i^S + \delta U_j^S||_{\mathcal{U}^S}^2 \right)}, \tag{1.22}$$

being $\delta U_i^S = \delta U_{i|_S}$.

Proof. Let us compute $J(U + \lambda\delta U)$. We have

$$\begin{aligned}
J(U+\lambda\delta U) &= J(U) + 2\sum_{S\in\mathcal{S}} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j), \lambda(C_i^S\delta H_i - C_j^S \delta H_j) \right)_{\mathcal{H}^S} \\
&+2\sum_{S\in\mathcal{S}} \left(U_i^S + U_j^S, \lambda(\delta U_i^S + \delta U_j^S) \right)_{\mathcal{U}^S} \\
&\qquad +\lambda^2 \sum_{S\in\mathcal{S}} ||C_i^S\delta H_i - C_j^S \delta H_j||_{\mathcal{H}^S}^2 + \lambda^2 ||\delta U_i^S + \delta U_j^S||_{\mathcal{U}^S}^2 \\
&= J(U) + 2\sum_{i\in\mathfrak{I}} \sum_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j), \lambda C_i^S \delta H_i \right)_{\mathcal{H}^S} \\
&+2\sum_{i\in\mathfrak{I}} \sum_{S\in\mathcal{S}_i} \left(U_i^S + U_j^S, \lambda\delta U_i^S \right)_{\mathcal{U}^S} + \lambda^2 \sum_{S\in\mathcal{S}} \left(||C_i^S\delta H_i - C_j^S\delta H_j||_{\mathcal{H}^S}^2 \right. \\
&\qquad \left. +||\delta U_i^S + \delta U_j^S||_{\mathcal{U}^S}^2 \right) \\
&= J(U) + 2\sum_{i\in\mathfrak{I}} \left(\prod_{S\in\mathcal{S}_i} (C_i^S H_i(U_i) - C_j^S H_j(U_j)), \lambda C_i \delta H_i \right)_{\mathcal{H}^{\mathcal{S}_i}} \\
&+2\sum_{i\in\mathfrak{I}} \left(U_i + \prod_{S\in\mathcal{S}_i} U_j^S, \lambda\delta U_i \right)_{\mathcal{U}^{\mathcal{S}_i}} + \lambda^2 \sum_{S\in\mathcal{S}} \left(||C_i^S\delta H_i - C_j^S\delta H_j||_{\mathcal{H}^S}^2 \right. \\
&\qquad \left. +||\delta U_i^S + \delta U_j^S||_{\mathcal{U}^S}^2 \right).
\end{aligned}$$

From the previous relation, recalling (1.19) we obtain

$$\begin{aligned}
&J(U+\lambda\delta U) - J(U) - \lambda^2 \sum_{S\in\mathcal{S}} (||C_i^S\delta H_i - C_j^S\delta H_j||_{\mathcal{H}^S}^2 + ||\delta U_i^S + \delta U_j^S||_{\mathcal{U}^S}^2) = \\
&= 2\lambda \sum_{i\in\mathfrak{I}} \left\langle A_i^* P_i, A_i^{-1} B_i \delta U_i \right\rangle_{V_i', V_i} + 2\lambda \sum_{i\in\mathfrak{I}} \left\langle \Lambda_{\mathcal{U}^S}(U_i + \prod_{S\in\mathcal{S}_i} U_j^S), \delta U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime}, \mathcal{U}^{\mathcal{S}_i}} \\
&= 2\lambda \sum_{i\in\mathfrak{I}} \left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} B_i^* P_i + U_i + \prod_{S\in\mathcal{S}_i} U_j^S, \delta U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i}, \mathcal{U}^{\mathcal{S}_i}} = 2\lambda \sum_{i\in\mathfrak{I}} ||\delta U_i||_{\mathcal{U}^{\mathcal{S}_i}}^2.
\end{aligned}$$

Then the value of λ in (1.22) vanishes the derivative of $\mathcal{J}(\lambda) := J(U+\lambda\delta U)$ with respect to λ, thus providing the minimum of the function $\mathcal{J}(\lambda)$. □

Summarizing, problem (1.17) can be solved, in the continuous framework, either solving equations (1.13), (1.18) and (1.19) or following an iterative algorithm such as

the steepest descent, in which at each iteration one step is taken along the direction δU computed by (1.20) with a stepsize λ given by (1.22).

The discrete counterparts of these two approaches are presented in Section 1.4.

1.3 The DFN problem with XFEM

In this section, we briefly account for the application of the Extended Finite Element Method (XFEM) to our context. In the first subsection, we briefly recall from literature some key points of XFEM; in the second subsection these ideas are applied to the DFN framework.

1.3.1 Description of XFEM

The XFEM [3, 8, 4] is a mesh-based numerical technique for the solution of partial differential equations in variational form, when non-smooth or discontinuous solutions are considered. The XFEM can reproduce irregularities that are arbitrarily placed in the domain, regardless of the underlying triangulation. The concept at the basis of the XFEM consists in combining the standard Finite Element (FE) approach with the Partition of Unity Method (PUM) [2], in order to overcome the limitations of FE in dealing with singularities. Customized *enrichment functions* are added to the standard FE approximation space in order to catch the non-smooth character of the solution and extend approximation capability.

In what follows only the description of the method in the case of continuous solutions with discontinuous first order derivatives (*weak* discontinuities) is reported, being the only situation of interest in our application. Customizations of the method for other cases can be found in [4, 14].

Given a problem with exact solution H in a domain $\omega \in \mathbb{R}^n$, with a sharp or weak singularity along the interface described by the manifold $S \subset \omega, S \in \mathbb{R}^{n-1}$, let $\mathcal{T}_\delta$ be a conforming triangulation on ω, and let $\mathrm{V}_\delta^{\mathrm{fem}}$ be a finite dimensional trial and test space defined on the elements of $\mathcal{T}_\delta$ and spanned by Lagrangian FE basis functions ϕ_ξ, $\xi \in \mathcal{I} = \left\{1, ..., N^{\mathrm{dof}}\right\}$:

$$\mathrm{V}_\delta^{\mathrm{fem}} = \mathrm{span}\left(\{\phi_\xi(\hat{x})\}_{\xi\in\mathcal{I}}\right). \tag{1.23}$$

Each basis function ϕ_ξ has compact support Δ_ξ.

In our applications, provided that the edges of the elements in $\mathcal{T}_\delta$ surrounding S match it exactly, the approximate solution of H with standard finite elements has the following form:

$$h_\delta^{\text{fem}}(\hat{x}) = \sum_{\xi \in \mathcal{I}} h_\xi^{\text{fem}} \phi_\xi(\hat{x}) \tag{1.24}$$

where h_ξ^{fem} is the degree of freedom corresponding to the basis function $\phi_\xi(\hat{x})$. Functions in $\mathrm{V}_\delta^{\text{fem}}$ are continuous and can have discontinuities in the first order derivatives across element edges.

Let assume Φ is a continuous bounded function on ω, $\Phi \in \mathrm{H}^1(\omega) \cap C^0(\bar{\omega})$ that well approximates the behaviour of H in a neighbourhood of S called Δ_S. With the XFEM this function is introduced into the standard FE space, thus defining a new *enriched* functional space with extended approximation capabilities. This can be done by means of the PUM, using the standard FE shape functions for the definition of a partition of unity. The new enriched functional space is:

$$\mathrm{V}_\delta^{\text{xfem}} = \text{span}\left(\{\phi_\xi(\hat{x})\}_{\xi \in \mathcal{I}}, \{\phi_\xi(\hat{x})\Phi(\hat{x})\}_{\xi \in \mathcal{J}} \right) \subset \mathrm{H}_0^1(\omega), \tag{1.25}$$

where we have identified with $\mathcal{J} \subset \mathcal{I}$ the subset of indices of functions ϕ_ξ whose support belongs to Δ_S. DOFs in $\mathcal{J}$ are called enriched DOFs and the corresponding nodes enriched nodes. Typically, as sketched in Figure 1.2 it is:

$$\mathcal{J} = \{\xi \in \mathcal{I} : \Delta_\xi \cap S \neq \emptyset\}. \tag{1.26}$$

Consequently the approximate solution h^{xfem} of the problem with the XFEM is:

$$h_\delta^{\text{xfem}}(\hat{x}) = \sum_{\xi \in \mathcal{I}} h_\xi^{\text{xfem}} \phi_\xi(\hat{x}) + \sum_{\xi \in \mathcal{J}} a_\xi^{\text{xfem}} \phi_\xi(\hat{x})\Phi(\hat{x}) \tag{1.27}$$

where h_ξ^{xfem} and a_ξ^{xfem} are the unknowns related to the standard and enriching basis functions, respectively. Since functions representing the non smooth behaviour of the solution are now present in the discrete subspace, the non smooth behaviour of the solution can be reproduced independently of the positioning of elements in $\mathcal{T}_\delta$ with respect to the interface S.

According to (1.26) only a small subset of total elements is enriched and this is a peculiarity of the XFEM if compared to PUM or other similar methods as for example the GFEM ([25, 26]). Elements in $\mathcal{T}_\delta$ may thus have a variable number of enriched nodes. In particular it is possible to group elements in three categories, following the classification used in [14] (see Figure 1.2):

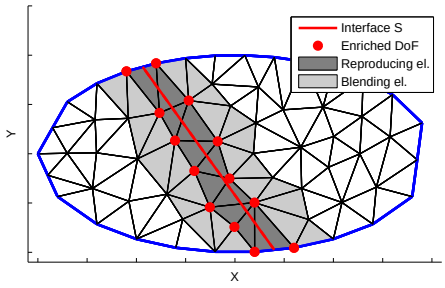


Figure 1.2: Selection of nodes in $\mathcal{J}$

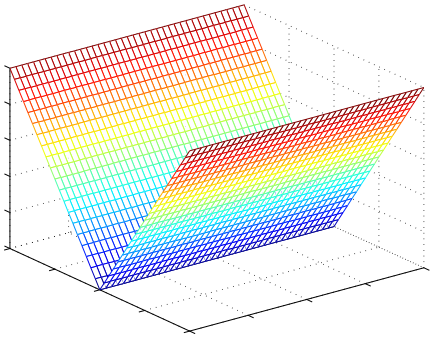


Figure 1.3: Function $\Psi(\hat{x})$

i) *standard* elements: no nodes enriched;

ii) *reproducing* elements: all nodes enriched;

iii) *blending* elements: some nodes enriched.

In reproducing elements, where all the nodes are enriched, the function Φ can be correctly reproduced, providing the desired behaviour for the discrete solution. In blending elements, instead, where only some nodes are enriched, spurious terms are introduced in the local discrete space in order to preserve continuity. This may affect the convergence rate of the method compared to the standard FE. Numerous techniques are suggested in order to prevent this issue, for example in [7, 27, 13]. In particular the *modified* XFEM suggested in [13] and adopted here, introduces a re-definition of enrichment functions and enriched DOFs in order to correctly account for the contribution of blending elements and recover the standard FE rate of convergence. We denote by $\widetilde{\Phi}$ and $\widetilde{\mathcal{J}}$ the modified version of Φ and $\mathcal{J}$ respectively, defined as:

$$\widetilde{\Phi} = \Phi(\hat{x})R(\hat{x}) \qquad \widetilde{\mathcal{J}} = \{\xi \in \mathcal{I} : \Delta_\xi \cap \Delta_S \neq \emptyset\}\,, \tag{1.28}$$

where $R(\hat{x}) = \sum_{\xi\in\mathcal{J}} \phi_\xi$. The new enrichment function $\widetilde{\Phi}$ coincides with Φ in reproducing elements where $R = 1$ and vanishes on the boundaries and outside Δ_S, where $R = 0$. Thus anywhere the enrichment function $\widetilde{\Phi}$ is non-zero it is correctly reproduced, avoiding problems related to parasitic terms.

The generalization to other kind of discontinuities follows the same outline described above, with specific re-definition of functional spaces. A comprehensive review of the XFEM/GFEM method with details of all implementation aspects is available in [14].

1.3.2 The discrete DFN problem

With reference to definitions and notation introduced in Section 1.2, we now discuss the application of the XFEM to DFN problems. For the sake of brevity we focus here on *closed* interfaces, i.e. traces entirely crossing a fracture plane, as for example the one depicted in Figure 1.2. Generalizations to other geometrical configurations of interfaces follow the same outline of this description, requiring, in some cases, the introduction of different enrichment functions. More general cases are considered in [5].

Let us consider a fracture $F \subset \mathbb{R}^2$ that has $\#\mathfrak{M}$ intersections with other fractures in Ω in the traces $S_m \in \mathcal{S}_i$, $m \in \mathfrak{M}$. The starting point for XFEM implementation is a standard finite element setting, defined by a triangulation $\mathcal{T}_\delta^F$ not necessarily conformal to the traces and the discrete test space $\mathrm{V}_{F,\delta}^{\mathrm{fem}}$ as defined by Equation (1.23). On F the exact solutions H_F, P_F and δH_F to (1.13), (1.19) and (1.21) respectively, may have a jump of fluxes (a *weak* discontinuity) across the traces in $\mathcal{S}_i$. The numerical solution of previous equations with XFEM allows the triangulation to be set on each fracture independently of the disposition and number of the traces. This is much more relevant as the number of traces increases or when traces intersect with arbitrary orientations, since in these situations a good quality mesh fitting the interfaces could hardly be produced and would require a huge number of elements, regardless of the required accuracy. Enrichment functions for *weak* discontinuities were introduced in early works with the XFEM mainly in the context of fracture mechanics. A comprehensive description can be found in [4, 27, 8, 14]. The description of each trace is performed introducing a signed distance function d_m that is defined for $\hat{x} \in F$ as the distance with sign from segments S_m [27, 4]:

$$d_m(\hat{x}) = \|\bar{x} - \hat{x}\| \frac{\hat{n}_S \cdot (\bar{x} - \hat{x})}{\|\hat{n}_S \cdot (\bar{x} - \hat{x})\|}$$

where $\bar{x}$ is the projection of $\hat{x}$ on S_m and $\hat{n}_S$ the fixed unit normal vector to S_m. The enrichment functions are built starting from the signed distance functions. For a closed interface we use the enrichment function Ψ_m defined as $\Psi_m(\hat{x}) = |d_m(\hat{x})|$. Clearly Ψ_m is a continuous function, but its first order derivatives have a jump across S_m, thus introducing the required non-smooth behaviour in the approximation (Figure 1.3). The sets of enriched DOFs, $\mathcal{J}_m$, are defined according to (1.26) for each trace.

In order to avoid problems related to blending elements, the XFEM modified version [13] is used. Functions $\widetilde{\Psi}_m$ and sets $\widetilde{\mathcal{J}}_m$ are built starting from Ψ_m and $\mathcal{J}_m$ according

to definition (1.28). The discrete approximation space is thus:

$$\mathrm{V}_{F,\delta}^{\mathrm{xfem}} = \mathrm{span}\left(\{\phi_\xi(\hat{x})\}_{\xi\in\mathcal{I}}, \{\phi_\xi(\hat{x})\Phi_m(\hat{x})\}_{m\in\mathfrak{M},\xi\in\mathcal{J}_m}\right) \subset \mathrm{H}_0^1(\omega), \tag{1.29}$$

and the discrete solution is:

$$h_{F,\delta}^{\mathrm{xfem}}(\hat{x}) = \sum_{\xi\in\mathcal{I}} h_\xi\phi_\xi(\hat{x}) + \sum_{m\in\mathfrak{M}}\sum_{\xi\in\widetilde{\mathcal{J}}_m} a_\xi^m\phi_\xi(\hat{x})\widetilde{\Psi}_m(\hat{x}). \tag{1.30}$$

We remark the additivity of the previous formula with respect to the interfaces: the previous expression does not depend on where traces are located, how close are each other, or wether or not they do intersect each other, nor on which elements the enriched functions are defined.

The numerical integration of non smooth functions is performed on sub-domains where the restriction of basis functions is regular. Gauss quadrature rule is used, adopting the number of integration nodes required by the polynomial degree of the integrands.

1.4 Discrete formulation

In this section we provide a discrete formulation of problem (1.17). For the sake of simplicity, we assume in this section homogeneous Dirichlet boundary conditions, i.e. $H_D = 0$. All the results can be extended to the general case $H_D \neq 0$, see later Remark 1.3. For simplicity of notation again, in this section, given two (or more) vectors $x \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$, we will write (x, y) denoting the vector $(x^T, y^T)^T \in \mathbb{R}^{p+q}$.

Under assumptions (1.16), the minimum of the functional $J(U)$ is characterized by conditions involving a fractional power of the Laplace operator on the traces. Hence, we develop our numerical method for the approximation of the solution adopting the following choices:

$$\mathcal{U}^S = \mathrm{L}^2(S), \qquad \mathcal{H}^S = \mathrm{L}^2(S), \qquad \forall S \in \mathcal{S}\,. \tag{1.31}$$

Remark 1.2. We remark that with these choices the assumption of disconnected traces can be removed. This is due to the following property of the L^2-norm: if S_1 and S_2 are two possibly connected traces, then $\|\cdot\|^2_{\mathrm{L}^2(S_1\cup S_2)} = \|\cdot\|^2_{\mathrm{L}^2(S_1)} + \|\cdot\|^2_{\mathrm{L}^2(S_2)}$ (see also Remark 1.1).

For all $i \in \mathfrak{I}$, let $J_i \subset \mathfrak{I}$ be the subset of indices such that, for $j \in J_i$, the fracture F_j shares a trace with F_i. Furthermore, for all $i \in \mathfrak{I}$ and for all $S \in \mathcal{S}_i$, let us fix a finite

dimensional subspace of $\mathcal{U}^S$ for the discrete approximation u_i^S of the control variable U_i^S (with a similar notation let us also denote by h_i the discrete approximation of H_i). Let us introduce a basis $\{\psi_{i,k}^S\}_{k=1,\dots,N_{i,S}}$ for this subspace, so that we write

$$u_i^S = \sum_{k=1}^{N_{i,S}} u_{i,k}^S \psi_{i,k}^S \quad \forall i \in \mathfrak{I}, S \in \mathcal{S}_i \,.$$

Replacing these expressions in (1.15), using L^2-norm and $C_i^S h_i = h_{i|_S}$, we get

$$J(u) = \frac{1}{2} \sum_{i \in \mathfrak{I}} \sum_{S \in \mathcal{S}_i} \Bigg(\int_S (\sum_{k=1}^{N_i} h_{i,k} \phi_{i,k|_S} - \sum_{k=1}^{N_j} h_{j,k} \phi_{j,k|_S})^2 \,\mathrm{d}\gamma + \int_S (\sum_{k=1}^{N_{i,S}} u_{i,k}^S \psi_{i,k}^S + \sum_{k=1}^{N_{j,S}} u_{j,k}^S \psi_{j,k}^S)^2 \,\mathrm{d}\gamma \Bigg). \tag{1.32}$$

For all $i \in \mathfrak{I}$ and $S \in \mathcal{S}_i$, let us introduce the subset $K_{i,S} \subseteq \{1, ..., N_i\}$ of indices k of functions $\phi_{i,k}$ whose support has a nonempty intersection with S. The first integral in (1.32) rewrites as

$$\begin{aligned} I_{ij}^{S,h} \;=\; & \sum_{k \in K_{i,S}} h_{i,k}^2 \int_S \phi_{i,k|_S}^2 \,\mathrm{d}\gamma + 2 \sum_{k,\ell \in K_{i,S}} h_{i,k} h_{i,\ell} \int_S \phi_{i,k|_S} \phi_{i,\ell|_S} \,\mathrm{d}\gamma + \sum_{k \in K_{j,S}} h_{j,k}^2 \int_S \phi_{j,k|_S}^2 \,\mathrm{d}\gamma \\ & + 2 \sum_{k,\ell \in K_{j,S}} h_{j,k} h_{j,\ell} \int_S \phi_{j,k|_S} \phi_{j,\ell|_S} \,\mathrm{d}\gamma - 2 \sum_{k \in K_{i,S}} \sum_{\ell \in K_{j,S}} h_{i,k} h_{j,\ell} \int_S \phi_{i,k|_S} \phi_{j,\ell|_S} \,\mathrm{d}\gamma. \end{aligned}$$

Let us introduce vectors $h_i \in \mathbb{R}^{N_i}$, $h_i = (h_{i,1}, \dots, h_{i,N_i})^T$, $i \in \mathfrak{I}$ and setting $N^F = \sum_{i \in \mathfrak{I}} N_i$, let $h \in \mathbb{R}^{N^F}$ be obtained concatenating, for $i \in \mathfrak{I}$, vectors h_i. Hence from now on, besides denoting the discrete solution, h_i will also denote the vector of corresponding DOFs.

Next, for all $i \in \mathfrak{I}$, $S \in \mathcal{S}_i$ let us define matrices $M_i^S \in \mathbb{R}^{N_i \times N_i}$ and (for $j \in J_i$) $M_{ij}^S \in \mathbb{R}^{N_i \times N_j}$ as:

$$(M_i^S)_{k\ell} = \int_S \phi_{i,k|_S} \phi_{i,\ell|_S} \,\mathrm{d}\gamma, \qquad (M_{ij}^S)_{k\ell} = \int_S \phi_{i,k|_S} \phi_{j,\ell|_S} \,\mathrm{d}\gamma.$$

With these definitions, the first integral in (1.32) is written in compact form as

$$I_{ij}^{S,h} = h_i^T M_i^S h_i + h_j^T M_j^S h_j - 2 h_i^T M_{ij}^S h_j. \tag{1.33}$$

Let us turn to the second integral in (1.32). For a convenient compact form of this second integral, let us consider a different numbering of functions u_i^S induced by

the trace numbering. Let $S = S_m$ be a given trace, with $c_m = (i, j)$ (hence $i < j$). We denote by u_m^- the control function related to the m-th trace and corresponding to fracture F_i, and by u_m^+ the control function related to the same trace and corresponding to the other fracture, F_j. This numbering induces a different numbering also on the basis functions $\psi_{i,k}^S$, $\psi_{j,k}^S$ which can be labeled as $\psi_{m,k}^-$, $\psi_{m,k}^+$, respectively, and accordingly we set $N_m^+ = N_{i,S}$, $N_m^- = N_{j,S}$.

Then we have, for $\star = -$ or $+$,

$$u_m^\star = \sum_{k=1}^{N_m^\star} u_{m,k}^\star \psi_{m,k}^\star \quad \forall m \in \mathfrak{M}.$$

Now, let us introduce the vectors $u_m^\star \in \mathbb{R}^{N_m^\star}$, $u_m^\star = (u_{m,1}^\star, \dots, u_{m,N_m^\star}^\star)^T$, $m \in \mathfrak{M}$, $\star = -, +$, and setting $N^T = \sum_{m\in\mathfrak{M}}(N_m^- + N_m^+)$ we define $u \in \mathbb{R}^{N^T}$ as

$$u = (u_1^-, u_1^+, \dots, u_{\#\mathfrak{M}}^-, u_{\#\mathfrak{M}}^+).$$

Let us also define the following matrices:

$$\mathcal{M}_m^\star \in \mathbb{R}^{N_m^\star \times N_m^\star}, \qquad (\mathcal{M}_m^\star)_{k\ell} = \textstyle\int_S \psi_{m,k}^\star \psi_{m,\ell}^\star \,\mathrm{d}\gamma, \qquad m \in \mathfrak{M}, \quad \star = -, +$$
$$\mathcal{M}_m^\pm \in \mathbb{R}^{N_m^- \times N_m^+}, \qquad (\mathcal{M}_m^\pm)_{k\ell} = \textstyle\int_S \psi_{m,k}^- \psi_{m,\ell}^+ \,\mathrm{d}\gamma.$$

The second integral in (1.32), after some straighforward algebraic manipulation, rewrites as

$$\begin{aligned} I_{ij}^{S,u} \;=\; & \sum_{k=1}^{N_m^-} {u_{m,k}^-}^2 \int_S {\psi_{m,k}^-}^2 \,\mathrm{d}\gamma + 2\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_m^-} u_{m,k}^- u_{m,\ell}^- \int_S \psi_{m,k}^- \psi_{m,\ell}^- \,\mathrm{d}\gamma + \sum_{k=1}^{N_m^+} {u_{m,k}^+}^2 \int_S {\psi_{m,k}^+}^2 \,\mathrm{d}\gamma \\ & + 2\sum_{k=1}^{N_m^+}\sum_{\ell=1}^{N_m^+} u_{m,k}^+ u_{m,\ell}^+ \int_S \psi_{m,k}^+ \psi_{m,\ell}^+ \,\mathrm{d}\gamma + 2\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_m^+} u_{m,k}^- u_{m,\ell}^+ \int_S \psi_{m,k}^- \psi_{m,\ell}^+ \,\mathrm{d}\gamma \end{aligned}$$

and in compact form

$$I_{ij}^{S,u} = (u_m^-)^T \,\mathcal{M}_m^- \,u_m^- + (u_m^+)^T \,\mathcal{M}_m^+ \,u_m^+ + 2(u_m^-)^T \,\mathcal{M}_m^\pm \,u_m^+. \tag{1.34}$$

We can now write the whole functional $J(u)$ in matrix form properly assembling the previous matrices in a single one and resorting to vectors h and u. Let $G^h \in \mathbb{R}^{N^F \times N^F}$ and $G^u \in \mathbb{R}^{N^T \times N^T}$ be defined blockwise as follows: for $i \in \mathfrak{I}$, $m \in \mathfrak{M}$ we set

$$G_{ii}^h = \textstyle\sum_{S\in\mathcal{S}_i} M_i^S, \qquad G_{ij}^h = -M_{ij}^S \text{ for } j \in J_i,$$
$$\mathcal{M}_m = \begin{pmatrix} \mathcal{M}_m^- & \mathcal{M}_m^\pm \\ (\mathcal{M}_m^\pm)^T & \mathcal{M}_m^+ \end{pmatrix} \qquad G^u = \mathrm{diag}(\mathcal{M}_1, \dots, \mathcal{M}_{\#\mathfrak{M}}).$$

Since obviously $(M_{ij}^S)^T = M_{ji}^S$, matrix G^h is symmetric. The same property clearly holds true for G^u. With these definitions, the functional $J(u)$ can be rewritten as

$$J(u) = \frac{1}{2} h^T G^h h + \frac{1}{2} u^T G^u u.$$

Now, let us turn our attention to the constraints, writing the algebraic counterparts of operators A_i, B_i in equation (1.14): overloading notations, we let $A_i \in \mathbb{R}^{N_i \times N_i}$ and $B_i \in \mathbb{R}^{N_i \times N_{\mathcal{S}_i}}$ with $N_{\mathcal{S}_i} = \sum_{S \in \mathcal{S}_i} N_{i,S}$ also denote the matrices defining the algebraic operators as follows. We set

$$(A_i)_{k\ell} = \int_{F_i} \nabla\phi_{i,\ell} \nabla\phi_{i,k} \, dF_i, \quad \left(B_i^{S_m}\right)_{k\ell} = \int_{S_m} \phi_{i,k|_{S_m}} \psi_{m,\ell}^{\star} \, \mathrm{d}\gamma, \tag{1.35}$$

where, recalling that $I_{S_m} = \{i, j\}$, we take $\star = -$ if $i < j$ or $\star = +$ otherwise. Matrices $B_i^{S_m}$, $S_m \in \mathcal{S}_i$, are then grouped row-wise to form the matrix B_i, which acts on a column vector u_i containing all the control DOFs corresponding to traces of F_i. Vector u_i is obtained appending the blocks $u_m^{\star}$ in the same order used for assembling B_i, as the action of a suitable operator $R_i : \mathbb{R}^{N^T} \mapsto \mathbb{R}^{N_{\mathcal{S}_i}}$ such that $u_i = R_i u$. Hence, constraints (1.14) lead to the algebraic equations

$$A_i h_i - B_i R_i u = \tilde{q}_i, \qquad i \in \mathfrak{I}, \tag{1.36}$$

where $\tilde{q}_i$ accounts for the term q_i in (1.14) and for the weak discrete imposition of boundary conditions. Letting $w = (h, u) \in \mathbb{R}^{N^F + N^T}$ and defining

$$A = \mathrm{diag}(A_1, \ldots, A_{\#\mathfrak{I}}) \in \mathbb{R}^{N^F \times N^F}, \qquad B = \begin{pmatrix} B_1 R_1 \\ \vdots \\ B_{\#\mathfrak{I}} R_{\#\mathfrak{I}} \end{pmatrix} \in \mathbb{R}^{N^F \times N^T},$$

$$C = (A \quad -B) \in \mathbb{R}^{N^F \times N^F + N^T}, \qquad G = \mathrm{diag}(G^h, \, G^u), \tag{1.37}$$

the overall problem reads

$$\min_{w} \qquad \frac{1}{2} w^T G w, \tag{1.38}$$

$$\text{s.t.} \qquad Cw = \tilde{q}. \tag{1.39}$$

Hence the problem is a Quadratic Programming (QP) problem with equality constraints. First order necessary conditions for a point w^* to be a solution to (1.38)–(1.39) are given by the Karush-Khun-Tucker conditions (see e.g. [21]):

$$\mathcal{A} = \begin{pmatrix} G & C^T \\ C & 0 \end{pmatrix}, \qquad \mathcal{A} \begin{pmatrix} w^* \\ -p^* \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{q} \end{pmatrix} \tag{1.40}$$

being p^* the vector of Lagrange multipliers.

Remark 1.3. The results here presented do not rely on the assumption of homogeneous Dirichlet boundary conditions. If non homogeneous Dirichlet conditions are taken into account, the quadratic functional in (1.38) also contains a linear term, correspondingly the right-hand-side of (1.40) has a nonzero block, and the structure of the problem is therefore the same.

For further discussion, we recall the following classical result concerning solution of equality constrained QPs of the form (1.38)-(1.39), see for example [21]. Referring to problem (1.38)-(1.39), let n and p denote the dimension of w and the number of constraints, respectively, so that $G \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{p\times n}$.

Theorem 1.4. *Let C have full row rank and assume that the matrix Z^TGZ is positive definite, being Z a $n \times (n-p)$ matrix whose columns are a basis of the null space of C. Then the matrix $\mathcal{A}$ defined in* (1.40) *is non singular and the vector w^* satisfying* (1.40) *is the unique global solution of problem* (1.38)–(1.39).

Proof of existence and uniqueness of the solution to the discrete counterpart of problem (1.17) is now a direct application of Theorem 1.4.

Theorem 1.5. *Let us consider the discrete formulation* (1.38)-(1.39) *to the problem of subsurface flow in a DFN, with G and C defined as in* (1.37). *Then, the solution exists and is unique and coincides with the solution to* (1.40).

Proof. First, let us observe that G is symmetric positive semidefinite as for any $w = (h, u)$ we straightforwardly have $w^TGw \geq 0$. Furthermore, since all A_i are nonsingular, due to standard properties of FE discretizations, A is nonsingular as well and C has full row rank. As $\text{rank}(C) = N^F$ we have $\dim(\ker(C)) = N^T$. Let $z_1, \ldots, z_{N^T} \in \mathbb{R}^{N^F+N^T}$ be vectors forming a basis of $\ker(C)$. Then, for all z_k, let us partition $z_k = (z_k^h, z_k^u)$ with $z_k^h \in \mathbb{R}^{N^F}$ and $z_k^u \in \mathbb{R}^{N^T}$. We have $Az_k^h = Bz_k^u$, thus z_k has the form $(A^{-1}Bz_k^u, z_k^u)$. In particular, take $z_k^u = e_k$, where e_k is the k-th vector of the canonical basis of $\mathbb{R}^{N^T}$, hence $z_k = (A^{-1}Be_k, e_k)$. Let us compute $y = Gz_k = (G^hA^{-1}Be_k, G^ue_k)$. Let e_{N^F+s} be a vector of the canonical basis of $\mathbb{R}^{N^F+N^T}$ with $s \geq 1$. We have $y_{N^F+s} = e_{N^F+s}^TGz_k = e_s^TG^ue_k$ with $e_s \in \mathbb{R}^{N^T}$. In particular, taking $s = k$, we have

$$y_{N^F+k} = e_k^TG^ue_k = \int_S {\psi_{i,\ell}^S}^2 \, \mathrm{d}\gamma \tag{1.41}$$

for some $i \in \mathfrak{I}$ and some $1 \leq \ell \leq N_{i,S}$. Since the integral in (1.41) is nonzero, we have at least one component of Gz_k different from zero. Hence we have proved that for any

vector $z \in \ker(C)$, we have $Gz \neq 0$ (unless $z = 0$), hence $z \notin \ker(G)$. This proves that $\ker(G) \cap \ker(C) = \{0\}$. Let now Z be the matrix whose columns are given by the basis vectors z_k previously introduced. Since G is positive semidefinite we have, for any $y \in \mathbb{R}^{NF+NT}$, $y^T G y \geq 0$ and $y^T G y = 0$ if and only if $y \in \ker(G)$ (see e.g. [16]). Let $v \in \mathbb{R}^{NT}$ be an arbitrary vector, $v \neq 0$. Since $Zv \in \ker(C)$ and $\ker(G) \cap \ker(C) = \{0\}$, we have $Zv \notin \ker(G)$ and so $v^T Z^T G Z v > 0$. This proves positive definiteness of $Z^T G Z$. Applying Theorem 1.4 the thesis is proved. □

1.4.1 Computing numerical solutions

Saddle point system (1.40) represents a possible approach for obtaining a numerical solution. For DFN of moderate size, sparse (even direct) solvers can be used efficiently to compute a solution to (1.40). Nevertheless, when the DFN system is composed by a huge number of fractures, even if poor discretizations are introduced on each fracture, solving the linear system may be a quite demanding task and parallel computing has to be taken into account. If this is the case, instead of assembling the linear system and splitting information and operations among processors/cores, a gradient-based method such as the basic one depicted in the sequel can be taken into account. The following numerical method arises from the discretization of the steepest descent method briefly described at the end of Subsection 1.2.2. At step k, given u^k, let us compute h_i^k as the solution to (1.36) and p_i^k as the solution to

$$A_i^T p_i^k = G_{ii}^h h_i^k + \sum_{j \in J_i} G_{ij}^h h_j^k, \qquad \forall i \in \mathfrak{I}. \tag{1.42}$$

Then, we define a vector δu_i^k componentwise as the $\mathrm{L}^2(\mathcal{S}_i)$ projection of the function $p_i^k + \prod_{S_m \in \mathcal{S}_i}((u_m^-)^k + (u_m^+)^k)$ against basis functions (nodal interpolation can be taken, in case of Lagrangian basis functions). Then, we move along direction δu^k with a stepsize

$$\lambda_k = -\frac{\sum_{i \in \mathfrak{I}} (\delta u_i^k)^T \delta u_i^k}{\frac{1}{2} \sum_{i \in \mathfrak{I}} \sum_{S_m \in \mathcal{S}_i} \left(\| \delta h_{i|S_m}^k - \delta h_{j|S_m}^k \|^2_{\mathrm{L}^2(S_m)} + \| (u_m^-)^k + (u_m^+)^k \|^2_{\mathrm{L}^2(S_m)} \right)} \tag{1.43}$$

where δh_i^k is the solution to

$$A_i \delta h_i^k = B_i \delta u_i^k, \qquad \forall i \in \mathfrak{I}. \tag{1.44}$$

The corresponding algorithm is the following.

Algorithm 1.6. 1. Set $k = 0$ and initial guess for control variable u^0;

2. compute $h^0 = h(u^0)$ solving (1.36) on each fracture;

3. Do

   3.1. compute p^k solving on each fracture the dual problem (1.42);

   3.2. compute δu^k and solve (1.44) to get δh;

   3.3. evaluate λ^k according to (1.43) and update $u^{k+1} = u^k + \lambda^k \delta u^k$;

   3.4. compute $h^{k+1} = h^k + \lambda^k \delta h^k$

   3.5. $k = k + 1$.

   while stopping criterion *not satisfied*

Remark 1.7. Algorithm 1.6, which is the discretization of the infinite dimensional steepest descent method, is equivalent to the application of the steepest descent method to the finite dimensional problem (1.38)-(1.39).

Each iteration of Algorithm 1.6 essentially requires the solution of (1.42) and (1.44), whereas it is not necessary to solve the primal equation (1.36) at each iteration, because, thanks to linearity, the new value h^{k+1} for the numerical hydraulic head can be computed as shown in Step 3.4. Nevertheless, in practical computations, it is advisable to periodically replace Step 3.4 with the computation of h^{k+1} via the primal equation, in order to improve numerical stability.

We end this section highlighting that solutions to problems (1.42) and (1.44) can be obtained decoupling the computation among fractures. This point makes the method appealing when parallelization comes into play, as this approach turns out to be highly parallelizable in a very natural way, by distributing fractures among processors and involving a moderate exchange of data. This approach is suitable for massively parallel computers and GPU-based computers.

1.5 Numerical results

In this section we present some preliminary results which aim at showing viability and effectiveness of the method here proposed in circumventing any kind of problem concerning mesh generation on the whole DFN.

Two test problems have been considered here. In Problem 1 the numerical simulations are performed both with standard finite elements on conforming grids aligned to a trace, and with extended finite elements with a trace crossing mesh elements. Numerical results are compared to the known exact solution. In Problem 2 a more complex domain is considered. In both tests, traces entirely crossing a fracture are considered. The application of the method to more complex DFN configurations is shown in [5]. Triangular meshes and first order finite elements are used in all the tests. Let $V_{i,\delta}$ be the discrete enriched finite element space on the fracture F_i, $\forall i \in \mathfrak{I}$, defined according to (1.29). Let $\mathcal{U}_\delta \subset \mathcal{U}$ be the discrete space for the control functions. The space $\mathcal{U}_\delta$ is here defined as the space of the piecewise linear functions on the traces S_m, $m \in \mathfrak{M}$; the nodes of the 1D mesh on each trace are given by the intersections of the 2D mesh on the corresponding fracture with the trace itself. If an edge of the 2D mesh lies on the trace, the endpoints of the edge are taken as nodes of the 1D mesh.

In the presentation of numerical results the following convention is used:

- *FEM*: our optimization approach on standard finite element meshes without enrichments; meshes are aligned along the traces (Figure 1.4, left). For Problem 1 the same mesh is used in all the fractures. This method is used to compare our results with those obtained on a conforming mesh, in which it is ensured that the minimum of J equals 0.

- *XFEM*: extended FE are used and the meshes in all the fractures do not match along the traces (Figure 1.4, right). In this case the minimum of functional J computed with the discrete solutions is in general $\neq 0$.

In all tests we computed the numerical solution both using the gradient method and solving the linear system (1.40). When the gradient method was applied, we started from a null control u^0. Both the overall linear system (1.40) and the smaller dimension systems involved in (1.42) and (1.44) have been solved with MATLAB built-in direct solver.

Depending on the choices of the mesh on each fracture F_i, the minimum of functional $J(u)$ can be different from zero. In Algorithm 1.6 the following stopping criteria have been used:

$$J(u^k) - J(u^{k+1}) < \mathrm{tol}_1, \qquad \text{or} \qquad \frac{J(u^k) - J(u^{k+1})}{J(u^{k+1})} < \mathrm{tol}_2. \tag{1.45}$$

In the results here reported we used $\mathrm{tol}_1 = 10^{-15}$ and $\mathrm{tol}_2 = 10^{-3}$.

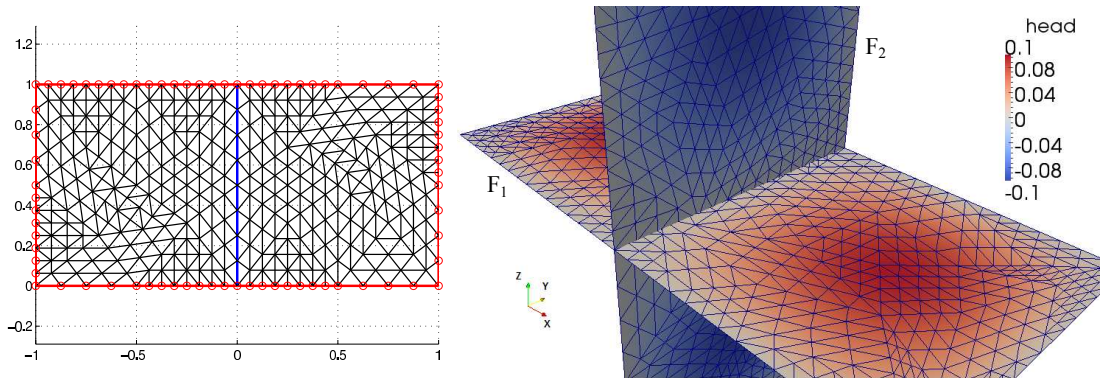


Figure 1.4: Problem 1: Left: standard FEM conforming mesh on each fracture; right: domain description with XFEM meshes and solution h in colorbar

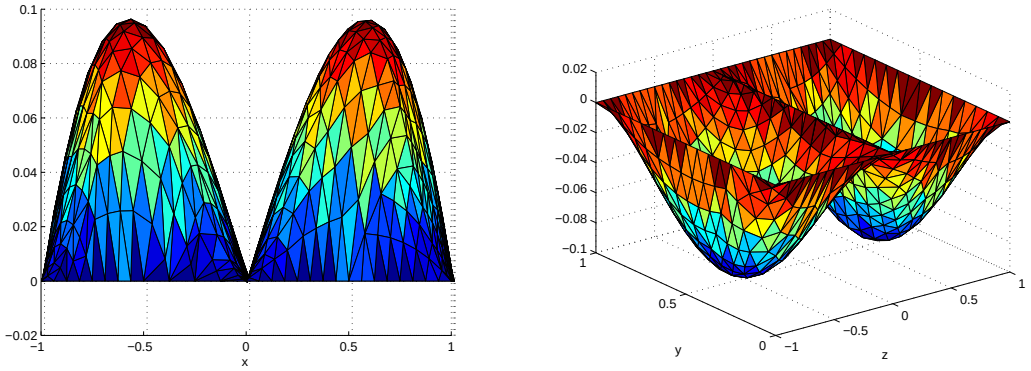


Figure 1.5: Problem 1: Solution with XFEM on fracture F_1 (left) and F_2 (right) for $\delta_{\max} = 0.06$

1.5.1 Problem 1

Let us define $\Omega = F_1 \cup F_2$ with, being $\mathbf{x} = (x, y, z)$, F_1 and F_2 given by

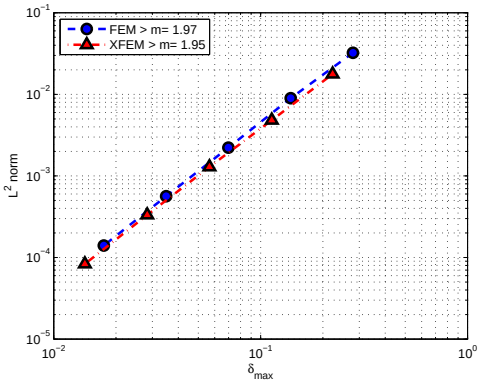
$$F_1 = \left\{\mathbf{x} \in \mathbb{R}^3 : x \in (-1, 1),\ y \in (0, 1),\ z = 0\right\} \quad F_2 = \left\{\mathbf{x} \in \mathbb{R}^3 : x = 0,\ y \in (0, 1),\ z \in (-1, 1)\right\}.$$

Let $S = F_1 \cap F_2$. The problem is set as follows:

$$-\Delta H = q, \qquad \text{in } \Omega \backslash S, \tag{1.46}$$

with homogeneous Dirichlet boundary conditions on all the boundary $\partial\Omega$. The forcing function q is defined as follows:

$$q(\mathbf{x}) = \begin{cases} 6(y - y^2)|x| - 2(|x^3| - |x|) & \text{on } F_1 \\ -6(y - y^2)|z| + 2(|z^3| - |z|) & \text{on } F_2 \end{cases}$$



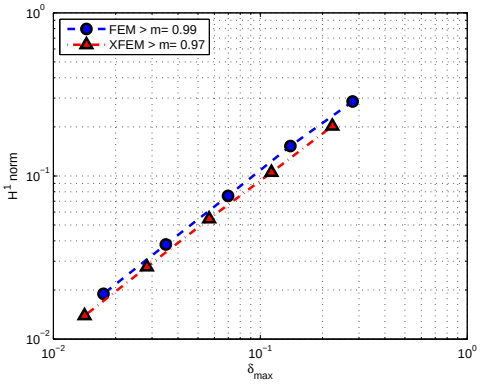


Figure 1.6: Problem 1: L^2 (left) and $\mathrm{H}^1(\Omega)$ (right) error norms under grid refinement

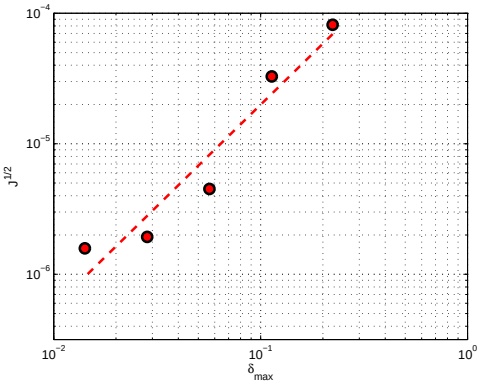


Figure 1.7: Problem 1: Minimum of $\sqrt{J}$ under grid refinement

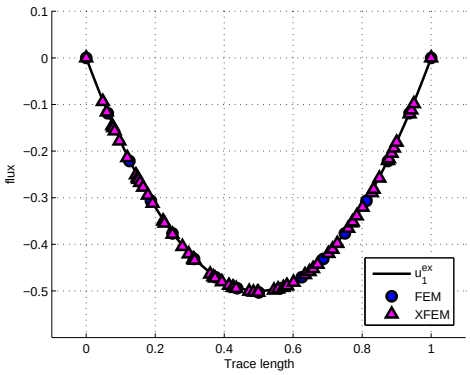


Figure 1.8: Problem 1: Control variable and exact solution (intermediate grid)

and the exact solution is given by

$$H(\mathbf{x}) = \begin{cases} -y(1-y)|x|(x^2-1) & \text{on } F_1 \\ y(1-y)|x|(x^2-1) & \text{on } F_2. \end{cases}$$

Figure 1.4 shows on the left a mesh used for the fractures F_1 and F_2 using standard finite elements, whereas on the right it shows the domain and, on each fracture, the mesh used with the extended finite elements. Note that in the second case the two meshes on F_1 and F_2 are not conforming. Both figures refer to intermediate meshes, corresponding to meshsize $\delta_{\max} = 0.06$, being $\delta_{\max} = 0.25$ and $\delta_{\max} = 0.016$ the meshsizes of the coarsest and finest grids used, respectively.

Figure 1.5 displays the solutions on F_1 and F_2 obtained with XFEM on the intermediate grid (the same solution is represented also in Figure 1.4, right, with a colorbar). Near the trace the numerical solution is plotted on the sub-elements generated by cutting XFEM elements along traces. It can be noted that the correct nonsmooth behaviour of the solution is caught by XFEM enrichments even if element edges do not match the trace. Figure 1.6 shows the behaviour of L^2 and H^1 error norms with respect to the meshsize δ_{max} during a uniform mesh refinement process. The slopes m of the curves, reported in the legend of each figure, agree with the expected values for P^1 elements even in the case of XFEM.

Remark 1.8. For this test prolem we have $H(\mathbf{x}) \notin \mathrm{H}^2(F_i)$, $i = 1, 2$, whereas $H(\mathbf{x}) \in \mathrm{H}^2(f)$, being f any one of the four subfractures in which F_1 and F_2 are divided by the trace. As described in [15, 29], this regularity is enough to provide the convergence orders of Figure 1.6, that are the theoretical ones for $H(\mathbf{x}) \in \mathrm{H}^2(F_i)$.

Figure 1.7 displays the minimum value of $\sqrt{J}$ as a function of the meshsize on non conforming meshes. In the XFEM case the target minimum of the functional is different from zero and, as expected, its value depends on the meshsize, while this is not the case for the standard FEM, since the minimum of the functional can vanish independently of the meshsize.

In Figure 1.8 the exact value of $\left[\!\left[\frac{\partial H_1}{\partial \hat{\nu}_S^1}\right]\!\right]_S$ is compared with the computed values of the control variable u_1 obtained on the intermediate grids, both with FEM and with XFEM. The figure clearly shows a very good agreement between all the values. The norm of the flux mismatch on the trace, i.e. $\|u_1 + u_2\|_{\mathrm{L}^2(S)}$, has been computed with both approaches, obtaining $\|u_1 + u_2\|_{\mathrm{L}^2(S)} \simeq 10^{-16}$ with FEM and $\|u_1 + u_2\|_{\mathrm{L}^2(S)} = 3.1\,10^{-8}$ with XFEM.

Remark 1.9. The vanishing of the minimum value of the functional with standard FEM does not correspond to a significantly better approximation of the global solution, as we can argue comparing the errors in the solution in Figure 1.6, where we can see that the errors corresponding to the same meshsize are comparable in the FEM and XFEM cases, with both L^2 and H^1-norms. As seen in Figure 1.8, also the accuracy of the fluxes on the trace are comparable. The vanishing minimum value of J for FEM is only related to a better satisfiability of the matching conditions between the approximated solutions on the fractures, and the accuracy of the overall solution is comparable for XFEM and FEM.

In Figure 1.9 the behaviour of $\sqrt{J}$ during the minimization process attained by the gradient method is shown. As expected the functional related to XFEM solution reaches a plateau corresponding to a non vanishing value when one of the stopping criteria in (1.45) is satisfied. As shown in Figure 1.9, mesh refinement can reduce the final functional value.

It is to remark that no effort has been spent here for improving convergence properties of the minimization process as our main target here is proving viability of the approach. Many improvements in the optimization process are possible; future work will be devoted to this issue. Nevertheless, despite the number of iterations required by the gradient method, the computational cost of each iteration is small, as it essentially requires the solution of the state equations on each fracture. This aspect itself makes the method appealing when parallelization comes into play.

1.5.2 Problem 2

In the second test problem the proposed method is applied to a DFN composed by seven rectangular fractures. In Figure 1.10 the intersections of the fractures with the plane $z = 0$ is drawn. All the fractures have z ranging from 0 to 1. In Figure 1.10, P_n, $n = 1, .., 14$ denotes the starting and ending points of the intersections; F_i, $i = 1, .., 7$ the intersection of the fractures with $z = 0$ and T_m, $m = 1, .., 11$ the intersections of the traces S_m with $z = 0$. The 3D DFN configuration is shown in Figure 1.11.

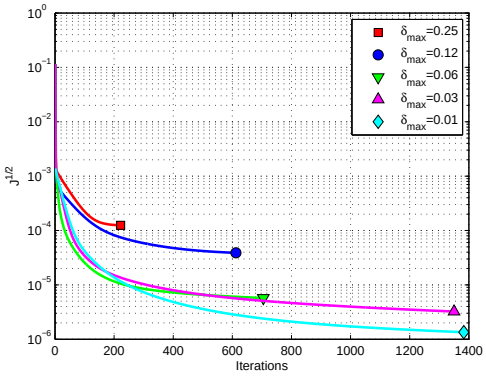


Figure 1.9: Problem 1: Functional trend against iterations with XFEM (five grids)

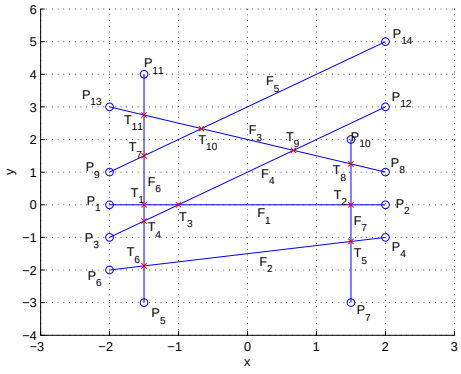


Figure 1.10: Problem 2: Fractures configuration, projection on $x-y$ plane

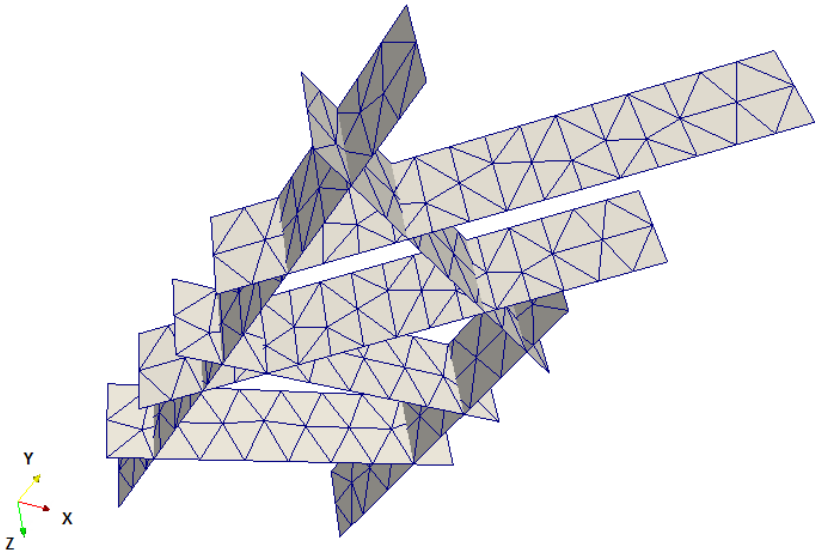
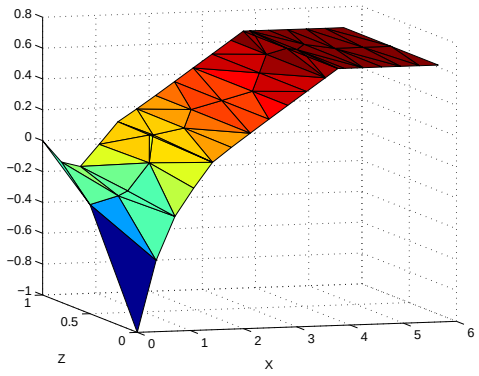


Figure 1.11: Problem 2: Fractures configuration and meshes



	flux mismatch	total flux
F_1	2.43e-6	-0.39
F_2	2.01e-6	0.64
F_3	3.27e-6	-1.14
F_4	4.27e-6	0.17
F_5	-5.36e-7	0.04
F_6	5.32e-6	0.29
F_7	4.95e-7	0.38

Figure 1.12: Problem 2: Solution on Fracture F_4 (left) and flux mismatch on the fractures (right).

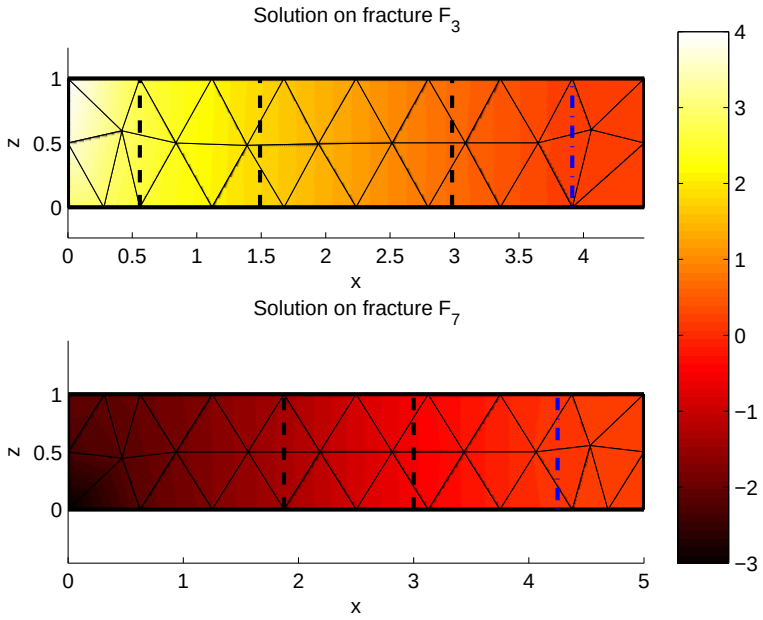


Figure 1.13: Problem 2: Solution on F_3 (top) and F_7 (bottom)

The problem is set as follows:

$$
\begin{aligned}
-\Delta H &= 0, && \text{in } \Omega \setminus \mathcal{S}, && (1.47)\\
H_{|\Gamma_D} &= y + \sqrt{z}, && on\ \Gamma_D, && (1.48)\\
\frac{\partial H}{\partial \hat{\nu}} &= 0, && on\ \Gamma_N. && (1.49)
\end{aligned}
$$

where $\mathcal{S} = \bigcup_{m=1,...,11} S_m$, Γ_D is the set of the edges along the z direction intersecting $z = 0$ in the points P_{13}, P_9, P_1, P_3, P_6, P_5 and P_7, whereas Γ_N is the set of all the other boundaries of the fractures. The computing mesh used is depicted in Figure 1.11. We remark that the meshes on the fractures are independently generated with meshsize $\delta_{\max} = 0.39$, without requiring any conformity constraint along the traces.

The solution is shown on some selected fractures. In Figure 1.12 the solution on fracture F_4 is shown. Here, in order to better display the enriched numerical solution, it is plotted, rather than on the actual computing elements, on sub-elements generated by splitting the computing elements along traces.

Figure 1.13 shows, using a colormap, solutions on Fractures F_3 and F_7. Here, the mesh depicted is the actual computing mesh. The vertical dashed lines correspond to traces. The rightmost dash-dot vertical line is a common trace between the two fractures. Nonconformity of meshes is clearly shown in the Figure. Finally, in the Table on the right of Figure 1.12 we report, for each fracture F_i, $i = 1, ..., 7$, the flux mismatch and total flux, computed as $\sum_{S \in \mathcal{S}_i} \int_S u_i^S + u_j^S \, \mathrm{d}\gamma$ and $\sum_{S \in \mathcal{S}_i} \int_S u_i^S \, \mathrm{d}\gamma$, respectively. The overall flux mismatch on the whole DFN is 8.14e-6.

1.6 Conclusions

In this paper we propose a new approach to the Discrete Fracture Network simulation, which does not need any kind of conformity along the traces for the meshes introduced on the fractures. The method proposed thus circumvents all the difficulties typically related to mesh generation processes of partially or totally conforming grids. This novel approach is based on a PDE-constrained optimization problem and is developed in order to be easily parallelized on massively parallel or GPU-based or hybrid parallel computers. The key points which make the method suitable for a parallel approach are the following: the global solution is obtained through the resolution of many small local problems, that require a moderate exchange of data among fractures.

Some preliminary numerical simulations prove the viability of the approach. A detailed analysis of the performance of the method on more complex fracture configurations is proposed in [5].

Acknowledgments

The authors are very grateful to Professor Fidelibus for many helpful discussions.

This work was financed by Italian funds from Miur-PRIN-2008 "Interazione tra modelli differenziali e tra metodi di discretizzazione diversi per il loro trattamento numerico" (200834WK7H_004), coordinated by Professor Franco Brezzi.

Bibliography

[1] P. M. Adler, *Fractures and Fracture Networks*, Kluwer Academic, Dordrecht, 1999.

[2] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[3] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, Internat. J. Numer. Methods Engrg., 45 (1999), pp. 601–620.

[4] T. Belytschko, N. Möes, S. Usui, and C. Parimi, *Arbitrary discontinuities in finite elements*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 993–1013.

[5] S. Berrone, S. Pieraccini, and S. Scialó, *A quadratic programming approach to extended finite element method for simulating discrete fracture network flows.* Submitted, SISC manuscript # 88288, 2012.

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

[7] J. Chessa, H. Wang, and T. Belytschko, *On the construction of blending elements for local partition of unity enriched finite elements*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 1015–1038.

[8] C. Daux, N. Möes, J. Dolbow, N. Sukumar, and T. Belytschko, *Arbitrary branched and intersecting cracks with the extended finite element method*, Internat. J. Numer. Methods Engrg., 48 (2000), pp. 1741–1760.

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

[10] W. S. Dershowitz and C. Fidelibus, *Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method*, Water Resource Res., 35 (1999), pp. 2685–2691.

[11] J. Erhel, J.-R. de Dreuzy, and B. Poirriez, *Flow simulation in three-dimensional discrete fracture networks*, SIAM J. Sci. Comput., 31 (2009), pp. 2688–2705.

[12] C. Fidelibus, *The 2d hydro-mechanically coupled response of a rock mass with fractures via a mixed bem-fem technique*, International Journal for Numerical and Analytical Methods in Geomechanics, 31 (2007), pp. 1329–1348.

[13] T.-P. Fries, *A corrected xfem approximation without problems in blending elements*, Internat. J. Numer. Methods Engrg., 75 (2008), pp. 503–532.

[14] T.-P. Fries and T. Belytschko, *The extended/generalized finite element method: an overview of the method and its applications*, Internat. J. Numer. Methods Engrg., 84 (2010), pp. 253–304.

[15] V. Girault and R. Glowinski, *Error analysis of a fictitious domain method applied to a Dirichlet problem*, Japan J. Indust. Appl. Math., 12 (1995), pp. 487–514.

[16] R. A. Horn and C. A. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[17] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[18] J. L. Lions, *Optimal Control of Systems Governed by Partial Differential Equations*, Grundelhren Der Mathematishen Wissenschaften : Vol 170, Springer-Verlag, 1971.

[19] I. Miller, G. Lee, and W. Dershowitz, *MAFIC Matrix/Fracture Interaction Code With Heat and Solute Transport User Documentation, Version 1.6*, Golder Associates Inc., Redmond, Washington, 1997.

[20] H. Mustapha and K. Mustapha, *A new approach to simulating flow in discrete fracture networks with an optimized mesh*, SIAM J. Sci. Comput., 29 (2007), pp. 1439–1459.

[21] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[22] A. W. Nordqvist, Y. W. Tsang, C. F. Tsang, B. Dverstop, and J. Andersson, *A variable aperture fracture network model for flow and transport in fractured rocks*, Water Resource Res., 28 (1992), pp. 1703–1713.

[23] G. Pichot, J. Erhel, and J.-R. De Dreuzy, *A mixed hybrid Mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629–1643.

[24] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[25] T. Strouboulis, I. Babuška, and K. Copps, *The design and analysis of the generalized finite element method*, Comput. Methods Appl. Mech. Engrg., 181 (2000), pp. 43–69.

[26] T. Strouboulis, K. Copps, and I. Babuška, *The generalized finite element method: an example of its implementation and illustration of its performance*, Internat. J. Numer. Methods Engrg., 47 (2000), pp. 1401–1417.

[27] N. Sukumar, D. Chopp, N. Möes, and T. Belytschko, *Modeling holes and inclusions by level sets in the extended finite element method*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 1209–1233.

[28] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.

[29] L. B. Wahlbin, *Local behavior in finite element methods*, in Handbook of numerical analysis, Vol. II, Handb. Numer. Anal., II, North-Holland, Amsterdam, 1991, pp. 353–522.

Chapter 2

On simulations of discrete fracture network flows with an optimization-based extended finite element method

Abstract Following the approach introduced in [7], we consider the formulation of the problem of fluid flow in a system of fractures as a PDE constrained optimization problem, with discretization performed using suitable extended finite elements; the method allows independent meshes on each fracture, thus completely circumventing meshing problems usually related to the DFN approach. The application of the method to discrete fracture networks of medium complexity is fully analyzed here, accounting for several issues related to viable and reliable implementations of the method in complex problems.

2.1 Introduction

In many applications, such as water resources monitoring, contaminant transport, oil/gas recovery, efficient numerical simulations of subsurface fluid flow in fractured porous rocks are of increasing interest. The description of the phenomena has to correctly account for the intrinsic heterogeneity and directionality of the rock medium and the multiscale nature of the flow. In dense fracture networks the flow can be well modelled as the flow in a continuous porous medium where fractures influence the distri-

bution of an equivalent permeability tensor. On the contrary, in sparse fracture networks flow properties are mainly determined by the larger fractures, thus Discrete Fracture Network (DFN) models are preferred to more conventional continuum models as basis for the simulations.

A DFN is an assemblage of resembling-fractures planar ellipses or polygons, stochastically generated given probabilistic data on distribution of density, aspect ratio, orientation, size, aperture and hydrological properties of the medium [13]. The fluid regime in a DFN can be conditioned even by the smallest elements, therefore neglecting fractures below a specified threshold is not recommended. As a consequence the number of generated fractures is frequently high even for a limited size of the domain of interest. Discretization thus often leads to poor meshes with a huge number of nodes. At the same time, a stochastic approach to the uncertainty of the parameters requires large numbers of simulations so that efficiency of numerical methods is of paramount importance for the applicability of DFN-based numerical solutions.

A DFN is a complex 3D structure. The first numerical challenge is to provide good-quality conforming meshes where the discretization of fracture intersections (*traces*) is the same on all the fractures involved. This is usually achieved by the introduction of a huge number of elements, independently of the required accuracy of the numerical solution.

In order to reduce computational cost, a possible approach consists in reducing the DFNs into systems of 1D pipes that are aligned along the fractures and mutually connect the centres of the traces with the surrounding fractures. This approach eases mesh generation problems and the resulting mesh of pipes still reflects the topological properties of the fracture network [8, 23]. An accurate definition of pipe properties is obtained with a boundary element method in [14].

Without resorting to dimensionality reduction, in [30] a mixed non-conforming finite element method on a conforming mesh is proposed. In [21], an adaptive approach to the conforming mesh generation requiring adjustments of the trace spatial collocations is proposed. Local modifications of the mesh or of the fracture network in order to preserve conformity of the meshes or alignment of meshes along the traces are considered in several works (see e.g. [18, 30]). In [15], a method to generate a good-quality conforming mesh on the network system is proposed based on the projection of the discrete 3D network on the 2D planar fractures in order to remove those connections among fractures which are difficult to be meshed. In [25, 26], a mixed hybrid mortar method is proposed

allowing nonconformities of the meshes on the fractures, but requiring that the traces are contained in the set of the edges of each fracture triangulation. Resorting to mortar methods the discretization of each fracture can lead to a different discretization of the traces. Interesting very complex DFN configurations are tested in [12].

In the recent work [7] the authors have proposed a different approach for the description of steady-state flows in a given DFN, which consists in the reformulation of the problem as a PDE constrained optimization problem. Following this approach, it is shown that the meshes introduced on each fracture are allowed to be independent of the meshes on other fractures, and independent of trace number and disposition, thus actually eliminating any kind of meshing problems related to DFN. The discrete problem is formulated as an equality constrained quadratic programming problem. Discretization on each fracture is performed with the extended finite element method for approximating the non smooth behaviour of the solution, which may present discontinuities in the fluxes. Here, we further analyze viability of the method proposed in [7] by discussing several issues arising when the method is applied to complex DFNs. In particular, we fully account for the extended finite element discretization with the so-called *open interfaces*, i.e. traces not ending on fracture edges. We also discuss preconditioning issues related to the numerical solution of the problem. Several numerical results are proposed, showing the capability of the method in dealing with complex situations, such as for example critical traces intersections.

The paper is organized as follows. In Section 2.2 we briefly recall the physical model and the continuous optimization problem, and in Section 2.3 the discrete formulation of the problem is given. In Section 2.4 we describe the basics of extended finite elements considered herein, with special attention to the treatment of open interfaces. In Section 2.5 numerical results are discussed in order to prove viability and reliability of the method.

2.2 Problem description

The quantity of interest of the problem we are dealing with is the hydraulic head, given by $H = \mathcal{P} + \zeta$, 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, ζ is the elevation. The computation of the hydraulic head in a Discrete Fracture Network requires the solution of differential equations on a system of planar polygonal open sets called fractures,

denoted by F_i with $i \in \mathfrak{I}$. Let us introduce on each F_i a local tangential coordinate system $\hat{x}_i$. Despite being planar, their orientations typically differ so that their union is a 3D set. Let us denote by Ω the union of the fractures and let $\partial\Omega$ be its boundary. The intersection of the closure of each couple of fractures is either an empty set or a set of non vanishing segments called *traces*, denoted by S_m, $m \in \mathfrak{M}$. Let $\mathcal{S}$ denote the set of all these traces. Furthermore, let each fracture of the system be endowed with a hydraulic transmissivity tensor $\mathbf{K}_i(\hat{x}_i)$.

In the paper the following assumptions are made on the DFN: 1) $\bar{\Omega}$ is a connected set; 2) each trace S_m, $m \in \mathfrak{M}$, is shared by exactly two polygonal fractures F_i and F_j, $i \neq j$: $S_m \subseteq \bar{F}_i \cap \bar{F}_j$; 3) on each fracture, the transmissivity tensor $\mathbf{K}_i(\hat{x}_i)$ is symmetric and uniformly positive definite.

Given a trace S_m, let F_i and F_j be the fractures sharing the trace: the set of indices i and j is denoted by $I_{S_m} = \{i, j\}$. For each fracture F_i let us denote by $\mathcal{S}_i$ the set of traces shared by F_i with other fractures, and by $J_i \subset \mathfrak{I}$ the set of indices of fractures sharing one trace with F_i.

While referring the reader to [7] for all the details, we sketch here a brief description of the approach. Let us split the boundary $\partial\Omega$ into two sets $\Gamma_D \neq \emptyset$ and Γ_N, with $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$, on which Dirichlet boundary conditions H_D and Neumann boundary conditions G_N are respectively imposed. Let H_{iD} and G_{iN} be the restriction of H_D and G_N to $\Gamma_{iD} = \Gamma_D \cap \partial F_i$ and $\Gamma_{iN} = \Gamma_N \cap \partial F_i$, respectively. Let us define $\forall i \in \mathfrak{I}$

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

and let V_i' be the dual space of V_i.

The global hydraulic head H in the whole connected system Ω is provided by the solution of the following problems: $\forall i \in \mathfrak{I}$ find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\int_{F_i} \mathbf{K}_i \nabla H \nabla v d\Omega = \int_{F_i} q_i v d\Omega + \int_{\Gamma_N \cap \partial F_i} G_{i,N} v_{|S} d\Gamma + \sum_{S \in \mathcal{S}_i} \int_S \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S v_{|S} d\Gamma, \quad (2.1)$$

where $\frac{\partial H_i}{\partial \hat{\nu}_S^i} = (\hat{n}_S^i)^T \, \mathbf{K} \nabla H$ is the outward co-normal derivative of the hydraulic head, being $\hat{n}_S^i$ the unique normal fixed for the trace S on the fracture F_i, and the symbol $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S$ denotes the jump of the co-normal derivative along $\hat{n}_S^i$. This jump is independent of the orientation of $\hat{n}_S^i$.

In equation (2.1) the left hand side models the diffusion of hydraulic head on each fracture, the first term of the right hand side is the external load in each facture, the

second is the term due to the Neumann boundary conditions, whereas the last term describes the net flow of hydraulic head entering the fracture at each trace.

In order to set up a well defined problem, the following matching conditions have to be added to (2.1):

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i,j \in I_{S_m}, \tag{2.2}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_{S_m}} \right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_{S_m}} \right]\!\right]_{S_m} = 0, \qquad \text{for } i,j \in I_{S_m}. \tag{2.3}$$

These two additional conditions correspond to the physical requirement of continuity of the hydraulic head and conservation of hydraulic fluxes across each trace S_m, $m \in \mathfrak{M}$. Condition (2.2) implies that the hydraulic head H on the whole domain Ω belongs to the space

$$V^D = \mathrm{H}^1_{\mathrm{D}}(\Omega) = \left\{ v \in \prod_{i\in\mathfrak{I}} V_i^D : (v_{|_{F_i}})_{|_{S_m}} = (v_{|_{F_j}})_{|_{S_m}},\ i,j \in I_{S_m},\ \forall m \in \mathfrak{M} \right\}. \tag{2.4}$$

For simplicity of notation and exposition in the following of this section we assume that the traces $S \in \mathcal{S}$ are disjoint. This assumption can be removed at the cost of a more complex and heavy notation. Let us define for each trace $S \in \mathcal{S}$ a suitable space $\mathcal{U}^S$ and its dual that we denote by $\left(\mathcal{U}^S\right)'$. We define similar spaces on all the traces of fracture F_i, $\forall i \in \mathfrak{I}$ and on the full set of traces $\mathcal{S}$:

$$\mathcal{U}^{\mathcal{S}_i} = \prod_{S\in\mathcal{S}_i} \mathcal{U}^S, \qquad \mathcal{U} = \prod_{i\in\mathfrak{I}} \mathcal{U}^{\mathcal{S}_i}.$$

For each trace S common to F_i and F_j we introduce suitable variables $U_i^S \in \mathcal{U}^S$ and $U_j^S \in \mathcal{U}^S$ representing the unknown quantities $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_S} \right]\!\right]_S$ and $\left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_S} \right]\!\right]_S$, respectively. Moreover, for each fracture F_i let us denote by

$$U_i = \prod_{S\in\mathcal{S}_i} U_i^S \in \mathcal{U}^{\mathcal{S}_i}$$

the tuple of functions U_i^S with $S \in \mathcal{S}_i$, and by $U = \prod_{i\in\mathfrak{I}} U_i \in \mathcal{U}$ the tuple of all functions U_i^S with $S \in \mathcal{S}_i$ and $i \in \mathfrak{I}$, i.e. the $2(\#\mathfrak{M})$-tuple of functions on all traces in $\bar{\Omega}$. Let us introduce the following linear bounded operators:

$$\begin{aligned}
A_i \in \mathcal{L}(V_i, V_i'), &\qquad \langle A_i H_i^0, v\rangle_{V_i',V_i} = \left(\mathbf{K}\nabla H_i^0, \nabla v\right),\ H_i^0 \in V_i,\\
A_i^D \in \mathcal{L}(V_i^D, V_i'), &\qquad \langle A_i^D H_i^D, v\rangle_{V_i',V_i} = (\mathbf{K}\nabla H_i^D, \nabla v),\ H_i^D \in V_i^D,\\
B_i \in \mathcal{L}(\mathcal{U}^{\mathcal{S}_i}, V_i'), &\qquad \langle B_i U_i, v\rangle_{V_i',V_i} = \langle U_i, v_{|_{\mathcal{S}_i}}\rangle_{\mathcal{U}^{\mathcal{S}_i},\mathcal{U}^{\mathcal{S}_i\prime}},\\
B_{\Gamma_{iN}} \in \mathcal{L}(\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), V_i'), &\qquad \langle B_{\Gamma_{iN}} G_{iN}, v\rangle_{V_i',V_i} = \langle G_{iN}, v_{|_{\Gamma_{iN}}}\rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}),\mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})}
\end{aligned}$$

the definitions holding $\forall v \in V_i$. Further, we introduce the dual operators $A_i^* \in \mathcal{L}(V_i, V_i')$, $B_i{}^* \in \mathcal{L}(V_i, \mathcal{U}^{\mathcal{S}_i\prime})$ and the Riesz isomorphism $\Lambda_{\mathcal{U}^{\mathcal{S}_i}} : \mathcal{U}^{\mathcal{S}_i} \to \mathcal{U}^{\mathcal{S}_i\prime}$. Finally, let $\mathcal{R}_i\, H_{iD} \in V_i^D$ be a lifting of Dirichlet boundary condition H_{iD}. The problem is then clearly stated as follows: $\forall i \in \mathfrak{I}$ *find* $H_i = H_i^0 + \mathcal{R}_i\, H_{iD}$, *with* $H_i^0 \in V_i$ *such that:*

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_{iN} - A_i^D\, \mathcal{R}_i\, H_{iD}. \tag{2.5}$$

2.2.1 Formulation as an optimization problem

The novel approach introduced in [7] consists in replacing the differential problems on the fractures (2.5) $\forall i \in \mathfrak{I}$, coupled with the matching conditions (2.2), (2.3), with a PDE constrained optimal control problem, in which the variable U acts as a control variable; equations (2.5) $\forall i \in \mathfrak{I}$ are the constraints, and the matching conditions are replaced by the task of minimizing a nonnegative functional. Let us define the spaces

$$\mathcal{H}^{\mathcal{S}_i} = \prod_{S \in \mathcal{S}_i} \mathcal{H}^S, \qquad \mathcal{H} = \prod_{i \in \mathfrak{I}} \mathcal{H}^{\mathcal{S}_i},$$

and the Riesz isomorphism $\Lambda_{\mathcal{H}^{\mathcal{S}_i}} : \mathcal{H}^{\mathcal{S}_i} \to \mathcal{H}^{\mathcal{S}_i\prime}$. We introduce the following linear bounded observation operators C_i^S and C_i and the dual $C_i{}^*$:

$$C_i^S \in \mathcal{L}(V_i, \mathcal{H}^S), \qquad C_i \in \mathcal{L}(V_i, \mathcal{H}^{\mathcal{S}_i}) = \prod_{S \in \mathcal{S}_i} C_i^S, \qquad C_i{}^* \in \mathcal{L}(\mathcal{H}^{\mathcal{S}_i\prime}, V_i').$$

For all $i \in \mathfrak{I}$, let us denote by $H_i(U_i)$ the solution to (2.5) corresponding to the value U_i for the control variable. Furthermore, fixed a fracture F_i, we denote by $\prod_{S \in \mathcal{S}_i} U_j^S$ the tuple of control functions defined on the fractures F_j intersecting F_i in the traces $S \in \mathcal{S}_i$.

Let us now introduce the following differentiable functional $J : \mathcal{U} \to \mathbb{R}$:

$$\begin{aligned} J(U) &= \sum_{S \in \mathcal{S}} J^S(U) = \sum_{S \in \mathcal{S}} \left(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{H}^S} + ||U_i^S + U_j^S||^2_{\mathcal{U}^S} \right) \\ &= \frac{1}{2} \sum_{i \in \mathfrak{I}} \left(|| \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right) ||^2_{\mathcal{H}^{\mathcal{S}_i}} + ||U_i + \prod_{S \in \mathcal{S}_i} U_j^S||^2_{\mathcal{U}^{\mathcal{S}_i}} \right). \end{aligned} \tag{2.6}$$

The problem of finding the hydraulic head in the whole domain is restated as the following optimization problem: find $U \in \mathcal{U}$ solving the problem

$$\min J(U) \quad \text{subject to (2.5)}, \ \forall i \in \mathfrak{I}. \tag{2.7}$$

In [7] it is shown that, if $\mathcal{U}^S = \mathrm{H}^{-\frac{1}{2}}(S)$ and $\mathcal{H}^S = \mathrm{H}^{\frac{1}{2}}(S)$, there exists a unique control variable U vanishing the functional $J(U)$ and correspondingly the unique solution H satisfying (2.5) $\forall i \in \mathfrak{I}$ is the solution to (2.1)-(2.3), as the vanishing of the two terms of the functional J corresponds to the imposition of the matching conditions (2.2), (2.3) $\forall m \in \mathfrak{M}$. It is further shown that the optimal control $U \in \mathcal{U}$ providing the minimum of the functional $J(U)$ is characterized by the following conditions:

$$(\Lambda_{\mathcal{U}^{\mathcal{S}_i}})^{-1} B_i{}^* P_i + U_i + \prod_{S \in \mathcal{S}_i} U_j^S = 0, \tag{2.8}$$

$\forall i \in \mathfrak{I}$, where the functions $P_i \in V_i$ are the solution of

$$A_i^* P_i \;=\; C_i{}^* \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right), \quad \text{in } F_i. \tag{2.9}$$

The computation of the solution to the problem of interest on the whole DFN may either be approached solving problems (2.5) coupled with equations (2.8) and (2.9) $\forall i \in \mathfrak{I}$, or setting up an iterative process for solving the optimization problem (2.7). In the next Section we will give details concerning computation of a numerical solution with these approaches.

Remark 2.1. The assumption of each trace being shared by exactly two fractures can be circumvented by redefining the functional as follows. With straightforward extension to more general cases, we allow three fractures F_i, F_j, F_k to share the same trace S. Then the corresponding $J^S(U)$ term in the definition of $J(U)$ is

$$\begin{aligned} J^S(U) \;=\; & ||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{H}^S} + ||C_i^S H_i(U_i) - C_k^S H_k(U_k)||^2_{\mathcal{H}^S} \\ & + ||U_i^S + U_j^S + U_k^S||^2_{\mathcal{U}^S}. \end{aligned}$$

2.3 Discretization of the constrained optimization problem

In this section, we account for the numerical solution of the problem, and we start briefly sketching the derivation of the finite dimensional counterpart of problem (2.7). For the sake of simplicity, in this section we assume homogeneous Dirichlet boundary conditions, i.e. $H_D = 0$. All the results can be extended to the general case $H_D \neq 0$. We describe our numerical method for the approximation of the solution assuming $\mathcal{U}^S = \mathrm{L}^2(S)$, $\mathcal{H}^S = \mathrm{L}^2(S)$, $\forall S \in \mathcal{S}$. We remark that with these choices the assumption of disconnected traces can be removed [7].

Let us introduce an independent conforming triangulation $\mathcal{T}_{\delta,i}$ on each fracture F_i $\forall i \in \mathfrak{I}$. Let $\mathrm{V}_{\delta,i}$ be the finite dimensional trial and test spaces defined on the elements of $\mathcal{T}_{\delta,i}$ and spanned by Lagrangian basis functions $\phi_{i,k}$, $k = 1, ..., N_i$. The discrete approximation of H_i on each fracture is defined as $h_i = \sum_{k=1}^{N_i} h_{i,k}\phi_{i,k}$, $\forall i \in \mathfrak{I}$.

Let us 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. Let us fix a finite dimensional subspace of $\mathcal{U}^S$ for the discrete approximation $u_m^\star$ of the control variable $U_m^\star$, $\star = -, +$ and let us introduce basis functions $\psi_{m,k}^-$, $k = 1, ..., N_m^-$ and $\psi_{m,k}^+$, $k = 1, ..., N_m^+$. Then we have, for $m \in \mathfrak{M}$, $\star = -, +$, $u_m^\star = \sum_{k=1}^{N_m^\star} u_{m,k}^\star \psi_{m,k}^\star$.

With these notations, using L^2-norms in (2.6) and $C_i^S h_i = h_{i|_S}$, we obtain the following finite dimensional form of the functional $J(u)$:

$$\begin{aligned} J(u) \;=\; & \frac{1}{2}\sum_{i\in\mathfrak{I}}\sum_{S\in\mathcal{S}_i}\int_S (\sum_{k=1}^{N_i} h_{i,k}\phi_{i,k_{|S}} - \sum_{k=1}^{N_j} h_{j,k}\phi_{j,k_{|S}})^2 \,\mathrm{d}\gamma + \\ & \frac{1}{2}\sum_{m\in\mathfrak{M}}\int_S (\sum_{k=1}^{N_m^-} u_{m,k}^-\psi_{m,k}^- + \sum_{k=1}^{N_m^+} u_{m,k}^+\psi_{m,k}^+)^2 \,\mathrm{d}\gamma. \end{aligned} \tag{2.10}$$

In view of deriving a compact form for (2.10), let us introduce vectors $h_i \in \mathbb{R}^{N_i}$, $h_i = (h_{i,1}, \ldots, h_{i,N_i})^T$, $i \in \mathfrak{I}$ and setting $N^F = \sum_{i\in\mathfrak{I}} N_i$, let $h \in \mathbb{R}^{N^F}$ be obtained concatenating, for $i \in \mathfrak{I}$, vectors h_i. Hence from now on, besides denoting the discrete solution, h_i will also denote the vector of degrees of freedom. Similarly, let us introduce the vectors $u_m^\star \in \mathbb{R}^{N_m^\star}$, $u_m^\star = (u_{m,1}^\star, \ldots, u_{m,N_m^\star}^\star)^T$, $m \in \mathfrak{M}$, $\star = -, +$, and setting $N^T = \sum_{m\in\mathfrak{M}}(N_m^- + N_m^+)$ we define $u \in \mathbb{R}^{N^T}$ concatenating $u_1^-, u_1^+, \ldots, u_{\#\mathfrak{M}}^-, u_{\#\mathfrak{M}}^+$.

For all $i \in \mathfrak{I}$, $S \in \mathcal{S}_i$, let us define matrices $M_i^S \in \mathbb{R}^{N_i\times N_i}$ and (for $j \in J_i$) $M_{ij}^S \in \mathbb{R}^{N_i\times N_j}$ as:

$$(M_i^S)_{k\ell} = \int_S \phi_{i,k_{|S}}\phi_{i,\ell_{|S}}\,\mathrm{d}\gamma, \qquad (M_{ij}^S)_{k\ell} = \int_S \phi_{i,k_{|S}}\phi_{j,\ell_{|S}}\,\mathrm{d}\gamma$$

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

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

Then, let $G^h \in \mathbb{R}^{N^F\times N^F}$ and $G^u \in \mathbb{R}^{N^T\times N^T}$ be defined blockwise as follows:

$$G_{ii}^h = \sum_{S\in\mathcal{S}_i} M_i^S,\ i \in \mathfrak{I} \qquad G_{ij}^h = -M_{ij}^S,\ i \in \mathfrak{I},\ j \in J_i \qquad G^u = \mathrm{diag}(\mathcal{M}_1, \ldots, \mathcal{M}_{\#\mathfrak{M}}).$$

With these definitions at hand, the functional $J(u)$ in matrix form reads

$$J(u) = \frac{1}{2} h^T G^h h + \frac{1}{2} u^T G^u u.$$

Matrices G^h and G^u are clearly symmetric and semi-definite.

Now, let us turn our attention to the algebraic counterparts of operators A_i, B_i in (2.5): overloading notations, we let A_i and B_i also denote the matrices defining the algebraic operators. We set $A_i \in \mathbb{R}^{N_i \times N_i}$ and $B_i^{S_m} \in \mathbb{R}^{N_i \times N_m^\star}$ as

$$(A_i)_{k\ell} = \int_{F_i} \nabla\phi_{i,\ell} \nabla\phi_{i,k} \, dF_i, \quad \left(B_i^{S_m}\right)_{k\ell} = \int_{S_m} \phi_{i,k_{|S_m}} \psi^\star_{m,\ell} \, \mathrm{d}\gamma, \tag{2.11}$$

where, being $S_m \subseteq \bar{F}_i \cap \bar{F}_j$, we take $\star = -$ if $i < j$ or $\star = +$ otherwise. Matrices $B_i^{S_m}$, $S_m \in \mathcal{S}_i$, are then grouped row-wise to form the matrix $B_i \in \mathbb{R}^{N_i \times N_{\mathcal{S}_i}}$, with $N_{\mathcal{S}_i} = \sum_{S_m \in \mathcal{S}_i} N_m^\star$ and $\star$ chosen as before, which acts on a column vector u_i obtained appending the blocks $u_m^\star$ in the same order used for $B_i^{S_m}$, as the action of a suitable operator $R_i : \mathbb{R}^{N^T} \mapsto \mathbb{R}^{N_{\mathcal{S}_i}}$ such that $u_i = R_i u$. According to these definitions, the constraints (2.5) lead to the algebraic equations

$$A_i h_i - B_i R_i u = \tilde{q}_i, \qquad i \in \mathfrak{I} \tag{2.12}$$

where $\tilde{q}_i$ accounts for the term q_i in (2.5) and the boundary conditions. Denoting $w = (h^T, u^T)^T \in \mathbb{R}^{N^F + N^T}$ and defining

$$A = \mathrm{diag}(A_1, \ldots, A_{\#\mathfrak{I}}) \in \mathbb{R}^{N^F \times N^F}, \qquad B = \begin{pmatrix} B_1 R_1 \\ \vdots \\ B_{\#\mathfrak{I}} R_{\#\mathfrak{I}} \end{pmatrix} \in \mathbb{R}^{N^F \times N^T},$$
$$C = \begin{pmatrix} A & -B \end{pmatrix} \in \mathbb{R}^{N^F \times N^F + N^T}, \qquad G = \mathrm{diag}(G^h, \, G^u), \tag{2.13}$$

the overall problem reads as the following equality constrained Quadratic Programming problem:

$$\min_w \ \frac{1}{2} w^T G w, \qquad \text{s.t. } Cw = \tilde{q}. \tag{2.14}$$

Classical results (see e.g. [22, Theorem 16.2]) show that, under proper assumptions on C and G, w^* is the unique global solution to (2.14) if and only if it is the unique solution to the following saddle point system:

$$\mathcal{A} = \begin{pmatrix} G & C^T \\ C & 0 \end{pmatrix}, \qquad \mathcal{A} \begin{pmatrix} w^* \\ -p^* \end{pmatrix} = \begin{pmatrix} 0 \\ \tilde{q} \end{pmatrix} \tag{2.15}$$

being p^* the vector of Lagrange multipliers. In [7] the following result, concerning existence and uniqueness of the solution to the discrete counterpart of problem (2.7), is proven.

Theorem 2.2. *Let us consider the discrete formulation* (2.14) *to the problem of subsurface flow in a DFN, with* G *and* C *defined as in* (2.13)*. Then, the solution exists and is unique and coincides with the solution to* (7.22)*.*

The numerical approximation of the hydraulic head can be obtained in a twofold manner. A possible method consists in solving the saddle point linear system (7.22). This approach is viable for DFNs of moderate size: in this case sparse solvers can efficiently compute a solution to (7.22). When very large DFN systems come into play, solving the linear system may be a quite demanding task even if very coarse meshes are used on each fracture, and parallel computing may become preferable. In these cases, as depicted in [7], a worthwhile approach consists in using a gradient-based method for the minimization of (2.14). Indeed, as shown in [7], this method allows for the decoupled solution of local problems on the fractures, with a moderate exchange of information among them. This point makes the method appealing for parallelization on massively parallel computers and GPU-based computers, in which the local problems on fractures can be distributed among processors.

2.4 XFEM Discretization

2.4.1 XFEM description

The Extended Finite Element Method (XFEM) [3, 20, 11, 4] is a finite element-based numerical method to approach partial differential equations in variational form with non smooth or discontinuous solutions. XFEM in the context of poro-fractured media are also used in [10]. The non smooth behaviour of the solution is added to the standard Finite Element (FE) approximation space through customized enrichment functions in order to extend approximation capabilities. By means of the Partition of Unity Method (PUM) [1] the influence of the enrichments is localized in a neighbourhood of irregularity interfaces. In this way the XFEM allows to reproduce irregularities regardless of the underlying triangulation.

Let us consider a problem set on a domain $\omega \subset \mathbb{R}^d$, with a weak discontinuity (i.e. a discontinuity in derivatives) along the manifold $S \subset \omega, S \subset \mathbb{R}^{d-1}$, and let $\mathcal{T}_\delta$ be a

conforming triangulation on ω, with N^{el} elements $\tau_e \subset \mathbb{R}^d$, $\bar{\omega} = \bigcup_{1 \le e \le N^{el}} \tau_e$. Let V_δ^{fem} be the standard finite dimensional trial and test space defined on the elements of $\mathcal{T}_\delta$ and spanned by Lagrangian basis functions ϕ_k, $k \in \mathcal{I}$. Each basis function ϕ_k has compact support denoted by Δ_k.

If the nonsmooth character of the solution is *a priori* known, it is possible to introduce it in the FEM discrete space. Let us assume Φ is a continuous bounded function on ω, $\Phi \in \mathrm{H}^1(\omega) \cap C^0(\bar{\omega})$ that well approximates the behaviour of a function h in a neighbourhood Δ_S of S given by the union of some mesh elements τ_e. It is possible to build a partition of unity on Δ_S based on the standard FE shape functions to define new *enriching* basis functions starting from Φ that can be introduced into the FEM space, thus giving the enriched functional space:

$$\mathrm{V}_\delta^{xfem} = span\left(\{\phi_k\}_{k\in\mathcal{I}}, \{\phi_k \Phi\}_{k\in\mathcal{J}}\right), \tag{2.16}$$

where $\mathcal{J} \subset \mathcal{I}$ is the subset of indices of functions ϕ_k used to define the partition of Δ_S. DOFs in $\mathcal{J}$ are called enriched DOFs (and the corresponding nodes enriched nodes). The selection of the domain Δ_S can vary with the specific application of the method, but is usually given by the union of mesh elements intersected by the interface S. The approximate solution h^{xfem} of the problem with the XFEM will be in general:

$$h_\delta^{xfem}(\hat{x}) = \sum_{k\in\mathcal{I}} h_k^{xfem} \phi_k(\hat{x}) + \sum_{k\in\mathcal{J}} a_k^{xfem} \phi_k(\hat{x}) \Phi(\hat{x}), \tag{2.17}$$

where h_k^{xfem} and a_k^{xfem} are the unknowns related to the standard and enriching basis functions, respectively. The nonsmoothness of the exact solution is now present in the discrete solution and is reproduced independently of the position of mesh elements. Since only a subset of total degrees of freedom is enriched, elements in $\mathcal{T}_\delta$ may have a variable number of enriched nodes. In particular, according to the classification given in [17] we have *standard* elements when no nodes are enriched, *reproducing* elements if all nodes are enriched, and *blending* elements if only some nodes are enriched.

The enrichment function Φ can be correctly reproduced only in reproducing elements where the partition of unity is complete. On the contrary, in the blending elements partition of unity is partially established and unwanted terms are introduced in the approximation, affecting the convergence rate of the standard FE [9, 29, 16]. Moreover the basis of V_δ^{xfem} is no longer a Lagrangian basis. For these reasons we will actually implement the modified version of XFEM with shifted basis functions, as suggested in

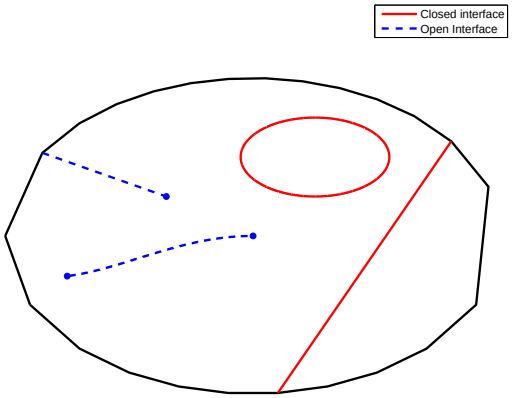


Figure 2.1: Classification of discontinuity interfaces

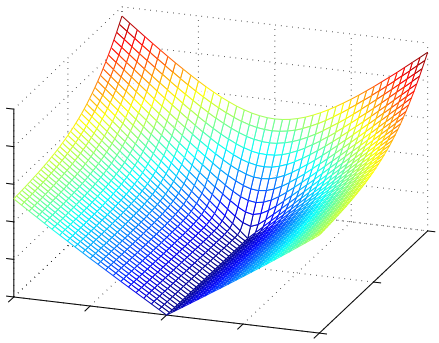


Figure 2.2: Example of function behaviour for near-tip enrichments

[16]. The enrichment basis function $\phi_k\Phi$ is replaced by

$$\phi_k(\hat{x})\widetilde{\Phi}(\hat{x}) = \phi_k(\hat{x})R(\hat{x})\left(\Phi(\hat{x}) - \Phi(\hat{x}_k)\right),$$

where $R(\hat{x}) = \sum_{j\in\mathcal{J}}\phi_j(\hat{x})$ and $\hat{x}_k$ are the coordinates of the k-th node. The enriched domain is extended including blending elements through a redefinition of the set $\mathcal{J}$ as $\widetilde{\mathcal{J}} = \left\{k \in \mathcal{I} : \Delta_k \cap \mathring{\Delta}_\Phi \neq \emptyset\right\}$, where $\Delta_\Phi = \bigcup_{k\in\mathcal{J}}\Delta_k$. In this way the approximation capability of the enriched space is unaffected in reproducing elements, where $R(\hat{x}) = 1$, and depends on the choice of the enrichment function Φ, while the standard FE polynomial representation of solution can now be obtained in blending elements, restoring optimal convergence rates. The shift restores Lagrangian property of the basis functions making easier the imposition of Dirichlet boundary conditions and graphical representation of the results.

The generalization to multiple enrichments is straightforward. In particular we remark that XFEM enjoys and additivity property with respect to the interfaces: independently of traces disposition, the set of enriching functions with multiple interfaces is the union of the enrichments introduced by each interface. A comprehensive review of the XFEM method, including implementation details, can be found in [17].

2.4.2 Enrichment functions selection

We now focus on the definition of the enrichments used in the application of the XFEM to DFNs. Recalling definitions introduced in Section 2.2, on each fracture F_i

the exact solutions H_i to (2.5) may have a jump of fluxes across the traces in $\mathcal{S}_i$. The XFEM approach allows the triangulation to be set on each fracture independently of the disposition and number of the traces, thus actually eliminating meshing problems related to DFNs. Let us fix a fracture $F \subset \mathbb{R}^2$ and let $\mathfrak{M}_F \subset \mathfrak{M}$ be the subset of indices corresponding to traces on F.

The selection of the enrichment functions is related to the irregularity to be reproduced and to the type of interfaces. Here we deal with solutions with discontinuous gradient (*weak* discontinuities) and different enrichment functions need to be employed according to the location of the traces (interfaces) in the domain, with a distinction between *closed* and *open* interfaces (see Figure 2.1). In order to describe the enrichment functions, let us introduce, for $m \in \mathfrak{M}_F$, the function $d_m(\mathbf{x})$ given by the signed distance from S_m [29, 4]: for $\hat{x} \in F$, $d_m(\hat{x}) = \|\bar{x} - \hat{x}\| \text{sign}(\hat{n}_{S_m} \cdot (\bar{x} - \hat{x}))$, where $\bar{x}$ is the projection of $\hat{x}$ on S_m and $\hat{n}_{S_m}$ is the fixed unit normal vector to S_m.

For a *closed* interface we use the enrichment function Ψ^m defined as $\Psi^m(\hat{x}) = |d_m(\hat{x})|$, [4], that is a continuous function with discontinuous first order derivatives across S_m. This introduces the required nonsmooth behaviour in the approximation. The enrichment is localized in a neighbourhood of S_m defined by the set of DOF $\mathcal{J}^m_\Psi = \{k \in \mathcal{I} : \Delta_k \cap S_m \neq \emptyset\}$.

On the contrary, if S_m is an *open* interface, different enrichment functions are needed to reproduce the behaviour of the solution close to the extrema of the interface and away from the extrema $\left\{s^1, s^2\right\} = \sigma_m$. Away from the extrema, the nonsmooth behaviour of the solution is similar to the case of closed interfaces and the same function Ψ^m is used, being the set $\mathcal{J}^m_\Psi$ defined as $\left\{k \in \mathcal{I} : \Delta_k \cap S_m \neq \emptyset,\, \Delta_k \cap s^\ell = \emptyset\right\},\ \forall s^\ell \in \sigma_m$. Other enrichment functions are introduced to describe near-tip behaviour of the solution; we adopt here the functions suggested in [4] and defined as follows. Let r be the signed distance between the current point and trace tip; furthermore, let us consider for each tip a reference system centered into trace tip, with the x-axis aligned to the trace and oriented in such a way that the trace lies on the negative side, and let $\theta \in (-\pi, \pi)$ be the polar angle of $\hat{x}$ in this system. Then, the enriching functions are

$$\Theta^m_{s^\ell}(\hat{x}) \in \left\{ r\cos\frac{\theta}{2}, r^2\cos\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2} \right\}, \qquad s^\ell \in \sigma_m.$$

Functions $\Theta^m_{s^\ell j}(\hat{x})$ are continuous and cusp-like on S_m, and their behaviour around trace tips is a combination of $\left\{\sqrt{r}, r, r^2\right\}$, as shown for example in Figure 2.2, in which we plot the function $r\cos\theta/2$. The set of DOFs subject to tip enrichments is given by

$\mathcal{J}^m_{\Theta s^\ell} = \{k \in \mathcal{I} : \Delta_k \cap s^\ell \neq \emptyset\}$, $\forall s^\ell \in \sigma_m$. In order to prevent blending elements related problems, the enrichment functions here described are used as basis for the modified XFEM version [16] mentioned in the previous subsection.

With all the enrichments here described, the number of DOFs on each fracture F_i is $N_i = \#\mathcal{I} + \sum_{m\in\mathfrak{M}} \#\widetilde{\mathcal{J}}^m + 3\sum_{m\in\mathfrak{M}}\sum_{s^\ell\in\sigma_m} \#\widetilde{\mathcal{J}}^m_{\Theta s^\ell}$, where $\widetilde{\mathcal{J}}^m$ and $\widetilde{\mathcal{J}}^m_{\Theta s^\ell}$ denote the sets of DOFs for the modified version.

The numerical integration of singular functions was performed on sub-domains not crossing the traces [20, 4]. A Gauss quadrature rule was used with special care for the integration of gradients of near-tip enrichment functions, where a concentration of integration nodes around trace tip is recommended to correctly evaluate the singularities [19].

2.5 Numerical results

The numerical simulations reported in this Section aim at showing the viability of the approach proposed in [7] in solving problems on complex networks. In Subsection 2.5.1 a problem with open interfaces is considered, and numerical convergence of the method is analyzed. In Subsection 2.5.2 a critical situation is introduced, in which three traces are very close each other, almost parallel and intersecting each other. The great deal of flexibility in mesh generation allowed by our approach is shown. In Subsection 2.5.3 some more complex DFNs are considered. In Subsection 2.5.4 preconditioning issues for system (7.22) are analyzed. Finally, in 2.5.5 we show how the method can deal with broadly ranging transmissivity values.

All the simulations are performed with triangular meshes and first order finite elements. The problems have been solved through the optimization approach introduced in [7], in conjunction with extended finite elements, and mesh elements arbitrarily placed with respect to the traces. We highlight that since the triangulations on a couple of intersecting fractures induce different discretizations on the common trace, the minimum of the discrete functional (2.10) is different from zero, that is the theoretical minimum of the functional in the continuous case.

The problems have been solved in a twofold manner: either solving the whole system (7.22) via an iterative method, or applying the steepest descent method to problem (2.14) (Algorithm 4.5 in [7]). Concerning the first case, the matrix $\mathcal{A}$ in (7.22) is

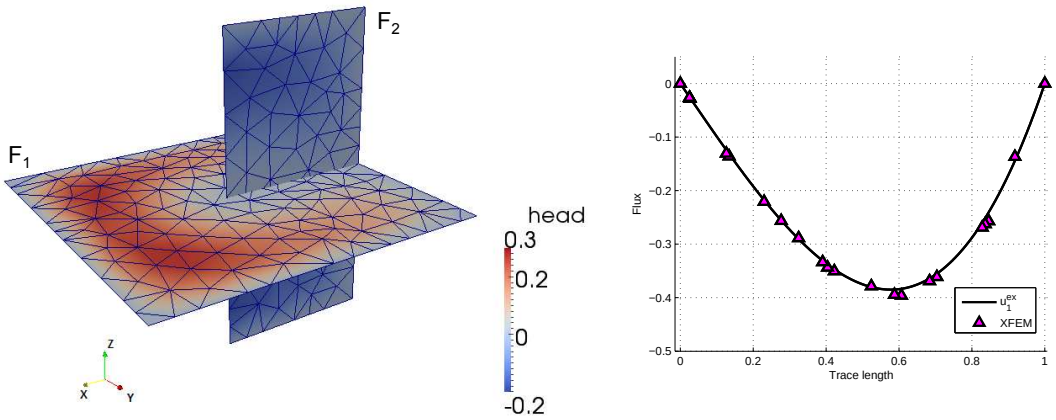


Figure 2.3: Problem 1: Domain description with mesh and solution h (left) and control variable along trace (right)

symmetric but indefinite, as shown in classic literature on saddle point problems (see e.g. [5]). Furthermore, in real applications $\mathcal{A}$ is of huge dimensions but highly sparse, hence an iterative method with matrix free approach appears to be a suitable choice. Among iterative methods for solving linear systems, SYMMLQ [24] is recommended for symmetric indefinite systems, and requires a symmetric positive definite preconditioner. This is the choice we adopted here, using the MATLAB built-in SYMMLQ function. The issue of preconditioning SYMMLQ on DFN applications is addressed in Subsection 2.5.4.

Nevertheless, when large DFNs are considered, even assembling and storing the system (7.22) may be a quite demanding task. The steepest descent method suggested in [7] may help in this respect as only the decoupled solution of local problems on fractures are required at each step, and with this approach a large problem can be dealt with also on a simple PC without requiring excessive memory resources. When this algorithm is used, the local problems (2.12) are typically of small dimension, so that a direct solver can be effectively used to compute these solutions. We used in our experiments the MATLAB built in direct solver. Computations are always started from $u^0 = 0$.

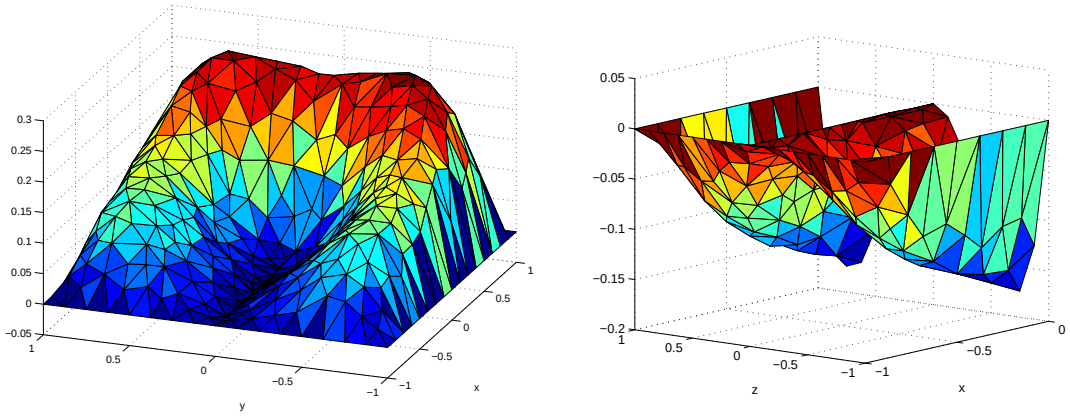


Figure 2.4: Problem 1: Numerical solution on F_1 (left) and on F_2 (right)

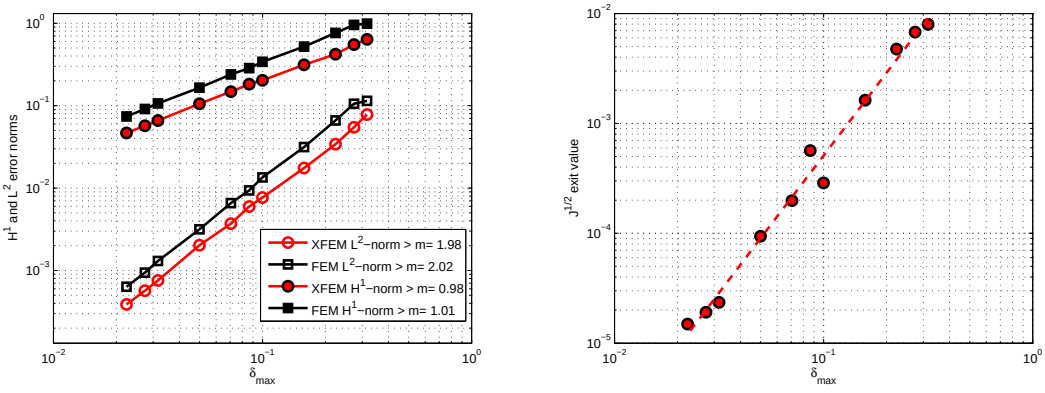


Figure 2.5: Problem 1: $\mathrm{H}^1(\Omega)$ and $L^2(\Omega)$ error norms (left) and functional minima (right) under refinement

2.5.1 Behaviour of the method with open interfaces

The first problem proposed is designed in order to test the behaviour of the method with near tip-enrichments. Let us define the domain $\Omega = F_1 \cup F_2$, with

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

The trace S ends in the interior of F_1 and is an open interface. Let us define $H^{ex}(x,y,z)$ in Ω as:

$$H^{ex}(x,y,z) = \begin{cases} (x^2-1)(y^2-1)(x^2+y^2)\cos\left(\frac{1}{2}\arctan2(x,y)\right) & \text{on } F_1, \\ -(z^2-1)(x^2-1)(z^2+x^2)\cos\left(\frac{1}{2}\arctan2(z,x)\right) & \text{on } F_2, \end{cases}$$

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 $\arctan(\cdot)$ for placing the angle in the correct quadrant. The function H is the solution of the system:

$$\begin{aligned} -\Delta H &= -\Delta H^{ex}, && \text{in } \Omega \setminus S, \\ H &= 0, && \text{on } \partial F_1 \cup \partial F_2 \setminus \Gamma, \\ H &= \frac{\sqrt{2}}{2}(z^2 - z^4), && \text{on } \Gamma, \end{aligned}$$

where Γ is the boundary of F_2 parallel to the z-axis and intersecting the x-axis in $x = -1$. In Figure 2.3 we report on the left the geometry of the problem and the non conforming mesh used with XFEM ($\delta_{\max} = 0.1$). On the right, we report the control variable u_1 computed, compared with the exact function. The flux mismatch computed along the trace is $\|u_1 + u_2\|_{\mathrm{L}^2(S)} = 2.8\,10^{-4}$. The results obtained with XFEM are shown in Figure 2.4. The problem has also been solved with standard finite elements on meshes conforming to the trace. The rates of convergence in both cases, reported in Figure 2.5 (left), are optimal. As expected, the curves relative to the solution obtained with the XFEM lie below the curves corresponding to standard finite elements. In fact, the basis function $r^2\cos\theta/2$ introduced for trace tip behaves essentially as H^{ex} close to the center of F_1, where tip is located, thus locally reducing the error with respect to standard FE. Minima of $\sqrt{J}$ are reported on the right plot of Figure 2.5, showing that grid refinement pushes these minima towards zero.

Table 2.1: Number of DOFs for fracture F_1 for different solution strategies

$A_{\max}$	XFEM	non-fitting FEM	fitting FEM
0.05	48	12	655
0.0225	85	34	672
0.01	135	71	715
0.0025	398	311	910
0.0004	486	396	1017

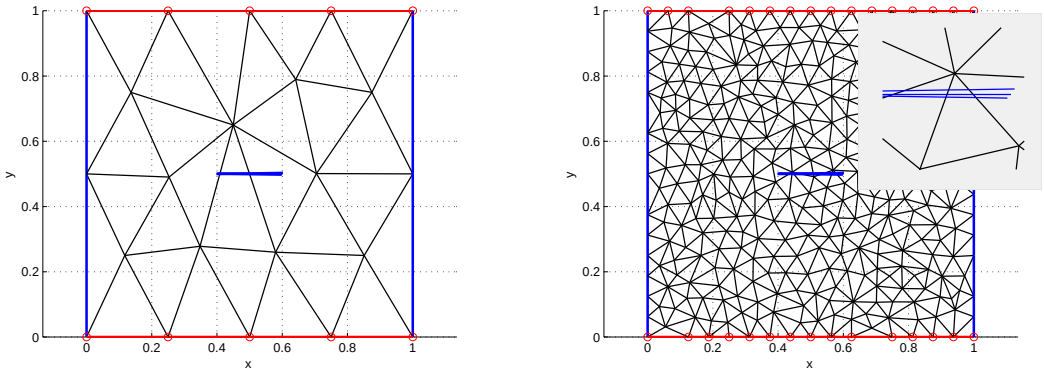


Figure 2.6: Problem 2: meshes on F_1. Left: coarse grid; right: fine grid.

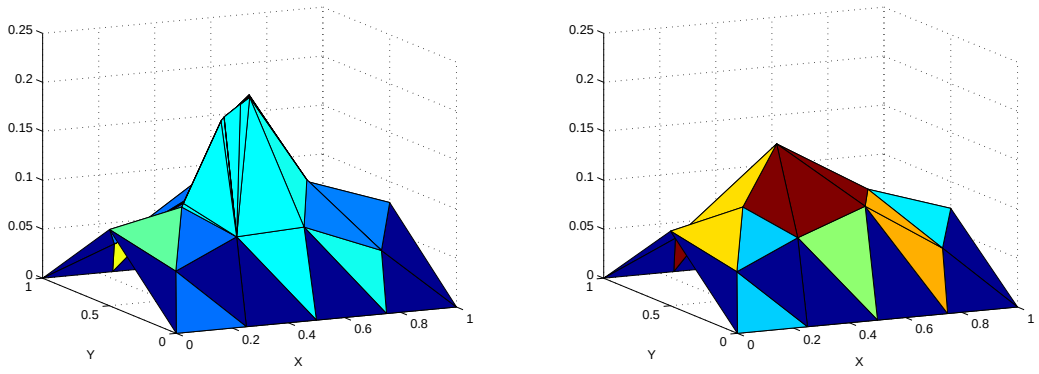


Figure 2.7: Problem 2: Solution on coarse grid. Left: XFEM; right: FEM

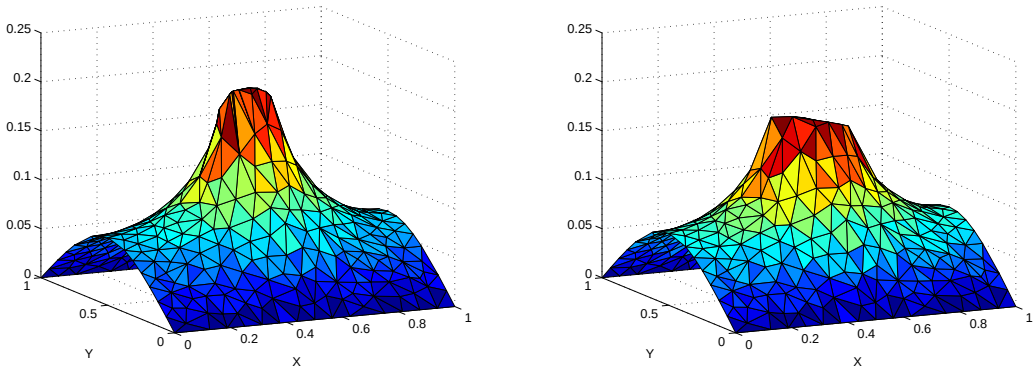


Figure 2.8: Problem 2: Solution on fine grid. Left: XFEM; right: FEM

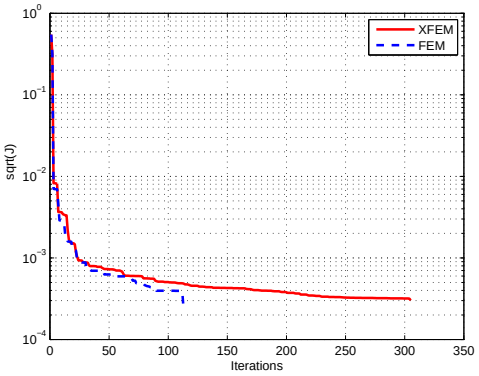


Figure 2.9: Problem 2: values of $\sqrt{J}$ against number of iterations

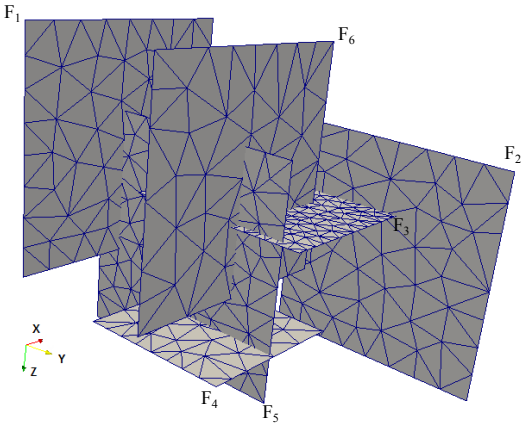


Figure 2.10: Problem 6F: Geometry with a coarse mesh

2.5.2 Critical traces disposition and DOFs investigation

In this subsection we consider a problem with critical traces disposition. We consider four fractures: F_1, located on the $x-y$ plane of a 3D reference system; the other three fractures are orthogonal to the $x-y$ plane and generate with F_1 three traces very close to each other and "almost" parallel, i.e. the angles between traces are very small, ranging from 0.8 (sexagesimal) degrees up to 1.8 degrees. The three traces are open interfaces. The fracture F_1 and the three traces are represented in Figure 2.6, along with examples of mesh used on F_1. On the right plot, also a detail of the right extremities of the traces is reported. The coordinates of traces extremities are $(x_1^b, y_1^b) = (0.4, 0.5)$, $(x_1^e, y_1^e) = (0.6, 0.5)$, $(x_2^b, y_2^b) = (0.398, 0.5)$, $(x_2^e, y_2^e) = (0.602, 0.503)$, $(x_3^b, y_3^b) = (0.402, 0.501)$, $(x_3^e, y_3^e) = (0.598, 0.498)$.

In Table 2.1 we report, for fracture F_1, the number of degrees of freedom obtained meshing the fracture for the following approaches: our optimization approach in conjunction with XFEM, hence without fitting the mesh to the traces; the same optimization approach, on the same mesh, with standard FEM basis functions (hence without enriching basis functions); standard FEM on a mesh fitting the traces. We remark that in this latter case the mesh has been generated only on F_1 and is only constrained to fit trace disposition; if also the mesh on the other three fractures were generated, and conformity on all the DFN were required, the number of degrees of freedom might be possibly even larger. In all three cases the meshes have been obtained with the software `Triangle` [28], requiring a good quality mesh ($-q$ option in `Triangle`) and imposing a given maximum element area $A_{\max}$, reported in Table 2.1. Comparing first and second column of the table, it is clear that, when the same mesh is considered, XFEM requires a larger number of DOFs than FEM, with a more significant percentage on the coarser meshes, since a larger fraction of elements are subject to enrichment. Under grid refinement, the number of elements enriched increases, but the percentage decreases, and the relative difference in DOFs between the two approaches becomes smaller. As shown by the last column, the number of DOFs introduced with a regular, fitting mesh, is in this case much higher then the previous ones, thus showing how effective is our approach in reducing the number of DOFs with respect to a conforming approach. Besides, we stress that non fitting meshes are produced without any kind of knowledge about traces disposition, thus easily obtained.

A problem has been introduced on this DFN as follows: $-\Delta H = 0$ in $\Omega \backslash S$; on F_1 we

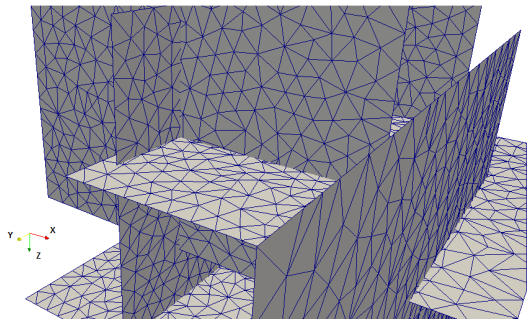


Figure 2.11: Problem 6F: detail of fine mesh

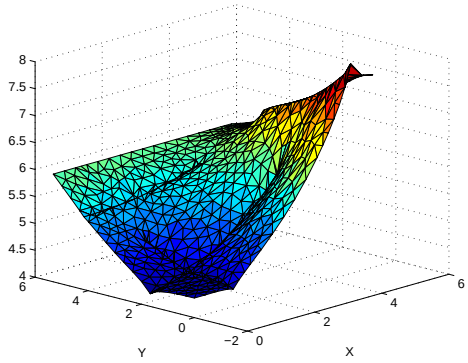


Figure 2.12: Problem 6F: solution on F_2 with fine mesh

set homogeneous Dirichlet conditions on fracture edges (almost) parallel to the traces, and homogeneous Neumann condition on the other sides; on fractures F_i, $i = 2, 3, 4$, we set $H = 1$ on the top edge, and homogeneous Neumann conditions on the other sides. The problem has been solved with the first two approaches mentioned before (XFEM and FEM on the same mesh, with our optimization approach). A coarse ($A_{\max} = 0.05$) and a fine ($A_{\max} = 0.0025$) mesh have been used, and are depicted in Figure 2.6. The numerical results obtained on the coarse and fine meshes are reported in Figures 2.7 and 2.8, respectively. The XFEM solutions are plotted on sub-elements generated by cutting XFEM elements along traces. Finally, in Figure 2.9 we report the values of $\sqrt{J}$ versus the number of iterations of the steepest descent method using both FEM and XFEM on the coarse mesh. It can be seen that the larger number of DOFs introduced by enrichments, and the larger number of iterations required by XFEM, are counterbalanced by the higher quality of the solution.

2.5.3 DFN systems simulations

In this subsection we consider systems of fractures of increasing complexity. Fracture transmissivities K_i are assumed constant on each fracture but different from fracture to fracture.

First, we consider the DFN configuration depicted in Figure 2.10: the system is composed by six fractures. Some of the traces generated do intersect each other. A detail of the mesh, presented in Figure 2.11, highlights non conformity of the mesh.

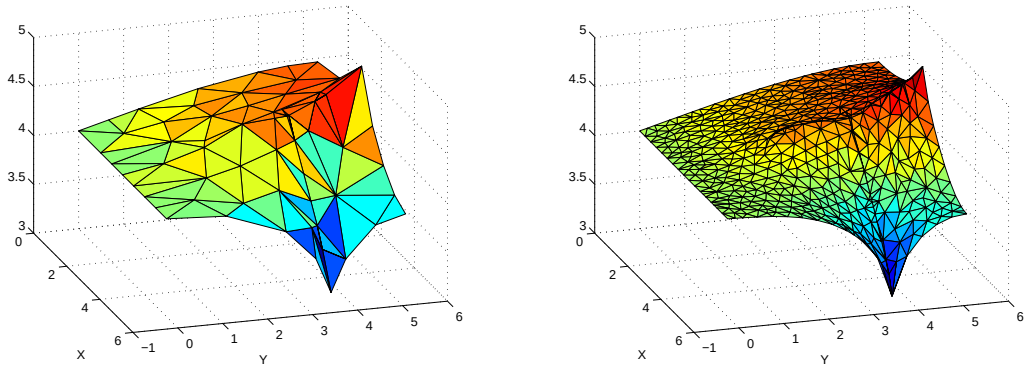


Figure 2.13: Problem 6F: solution on F_6 with coarse (left) and fine (right) mesh

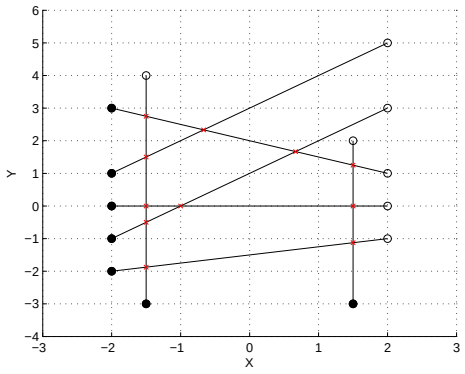


Figure 2.14: Problem 7F: Domain description

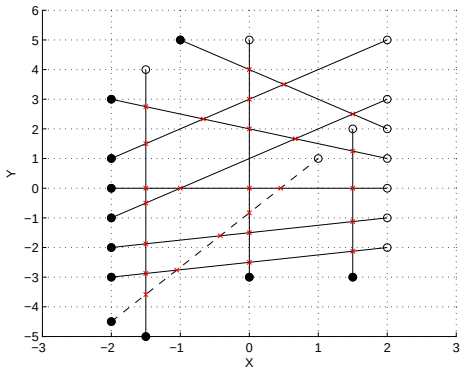


Figure 2.15: Problem 11F: Domain description

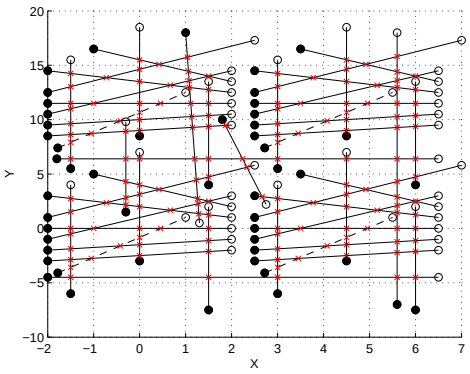


Figure 2.16: Problem 50F: Domain description

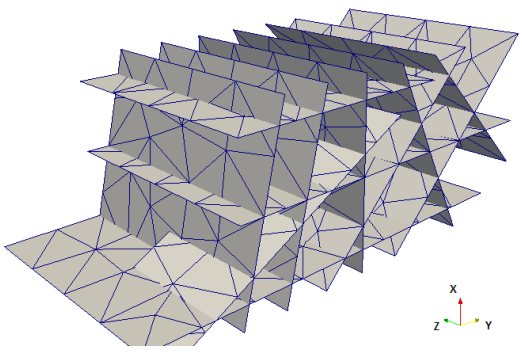


Figure 2.17: Problem 11F: geometry and a viable coarse mesh ($\delta_{\max} = 1$)

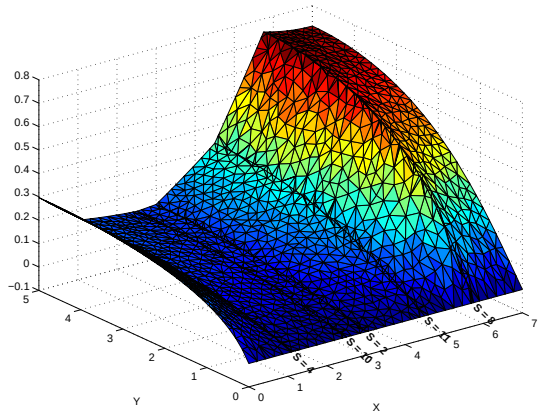


Figure 2.18: Problem 7F: Solution on fracture F_6 (traces numbering is global)

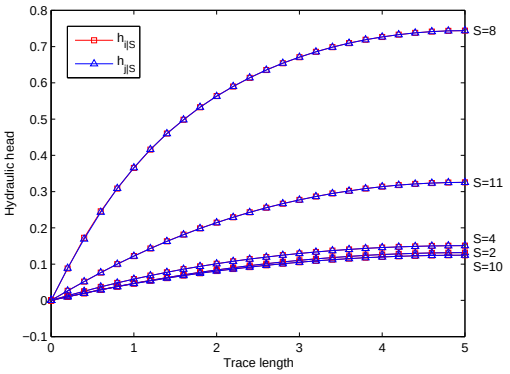


Figure 2.19: Problem 7F: Solution h_i on the traces of fracture F_6 and solutions $\{h_j\}$ on the fractures intersecting F_6 in its traces

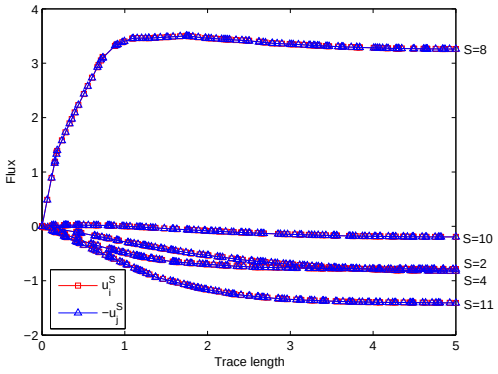


Figure 2.20: Problem 7F: Solution u_i on the traces of fracture F_6 and solutions $\{-u_j\}$ on the fractures intersecting F_6 in its traces

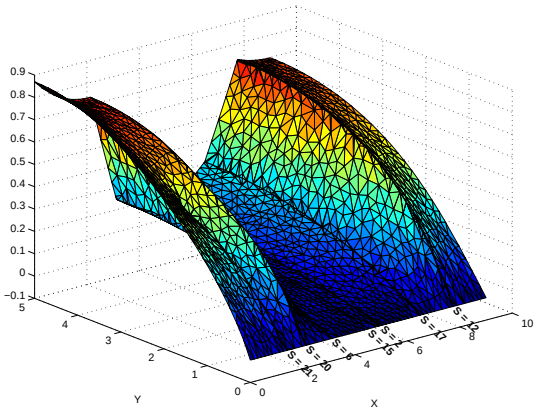


Figure 2.21: Problem 11F: Solution on fracture F_6 (traces numbering is global)

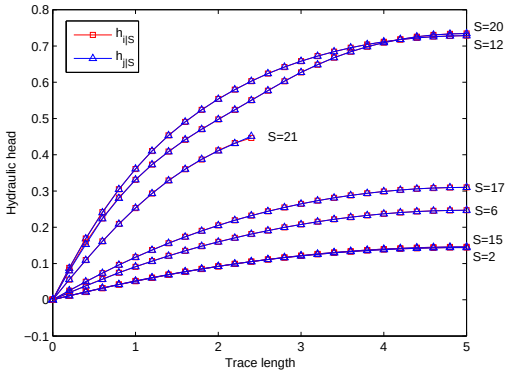


Figure 2.22: Problem 11F: Solution h_i on the traces of fracture F_6 and solutions $\{h_j\}$ on the fractures intersecting F_6 in its traces

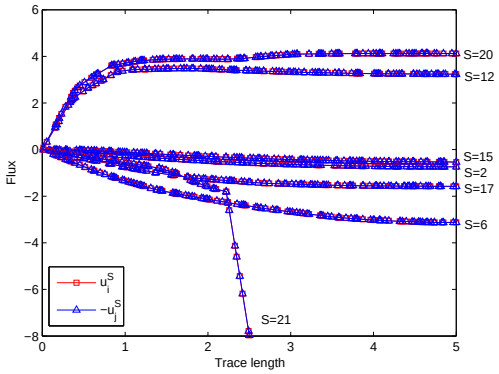


Figure 2.23: Problem 11F: Solution u_i on the traces of fracture F_6 and solutions $\{-u_j\}$ on the fractures intersecting F_6 in its traces

Table 2.2: Problem 11F: Fracture flux unbalance and total fluxes ($\delta_{\max} = 0.16$)

	flux unbalance	total flux		flux unbalance	total flux
F_1	-9.69e-7	1.44	F_7	-1.38e-6	0.50
F_2	-1.98e-6	4.72	F_8	-1.98e-6	-14.41
F_3	2.02e-7	-17.10	F_9	2.19e-6	9.06
F_4	-1.07e-6	2.99	F_{10}	3.61e-6	-4.17
F_5	-9.81e-7	7.20	F_{11}	3.87e-6	2.88
F_6	-2.51e-6	6.87			

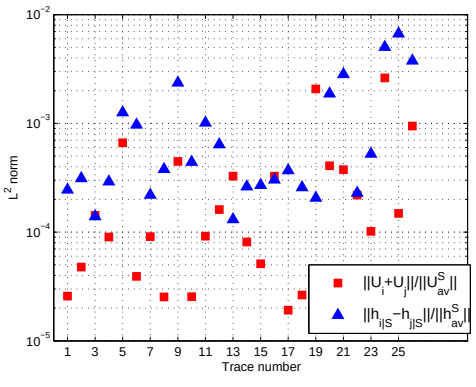


Figure 2.24: Problem 11F: relative continuity mismatch and flux unbalance

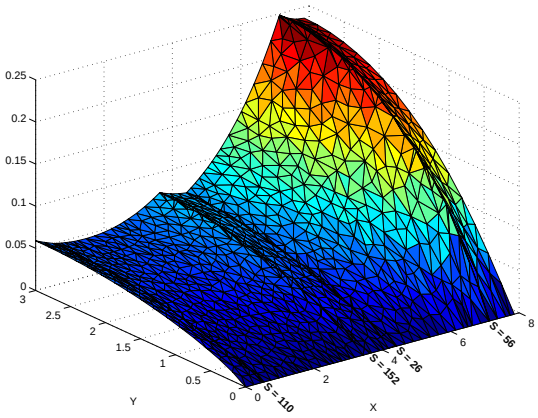


Figure 2.25: Problem 50F: Solution on fracture F_{50} (traces numbering is global)

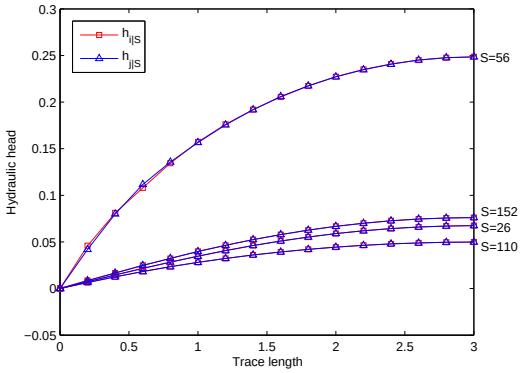


Figure 2.26: Problem 50F: Solution h_i on the traces of fracture F_{50} and solutions $\{h_j\}$ on the fractures intersecting F_{50} in its traces

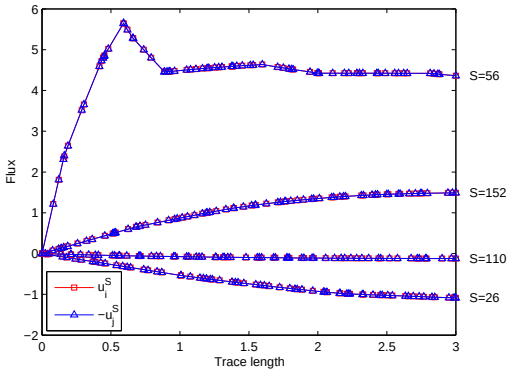


Figure 2.27: Problem 50F: Solution u_i on the traces of fracture F_{50} and solutions $\{-u_j\}$ on the fractures intersecting F_{50} in its traces

The numerical solution computed on fracture F_2 is reported in Figure 2.12, and is represented with respect to a local tangential reference system (X, Y). This convention also applies from now on to similar plots of the solutions. The figure shows that intersecting traces are easily handled by our approach. In particular, we see in Figure 2.12 that the discontinuities in the flux along the traces are clearly shown. In Figure 2.13 we report the solution computed on fracture F_6 with a coarse and a fine mesh ($\delta_{\max} = 0.77$ and $\delta_{\max} = 0.22$, respectively), showing the behaviour of the solution close to intersecting traces. The solutions are plotted on sub-elements obtained splitting XFEM elements along traces.

Then, the following configurations are considered. In these problems the reference system for $\mathbb{R}^3$ is a right-handed orthogonal system oriented such that the $x - y$ plane lies on the page plane, and fractures are parallel to z axis.

7F: The domain is composed of 7 fractures and 11 traces, as shown in Figure 2.14. Fractures range from $z = 0$ to $z = 5$. All the traces completely cross each fracture, thus tip-enrichments are not used.

11F: The domain is composed of 11 fractures and 26 traces, as shown in Figure 2.15. The fracture in dashed line ranges from $z = 0$ to $z = 2.5$, while all other fractures range from $z = 0$ to $z = 5$, thus in this case tip-enrichment functions are employed, since some traces end inside the domain.

50F: In this last case the domain is composed of 50 fractures and 153 traces as sketched in Figure 2.16. All fractures in continuous lines range between $z = 0$ and $z = 3$, while fractures drawn with dashed lines range from $z = 0$ to $z = 1.5$. Also in this case tip-enrichment functions are employed.

Boundary conditions are set in a similar fashion in all cases. Homogeneous Dirichlet boundary conditions are set on $\Gamma_D = \partial\Omega \cap \{z = 0\}$, while $\Gamma_N = \partial\Omega \setminus \Gamma_D$. A different constant-value of Neumann boundary condition is imposed on fracture edges belonging to Γ_N and marked with a plain black dot in the figures showing domain configurations. Homogeneous Neumann boundary conditions are placed on the other fracture edges in Γ_N. In all cases different (constant) values of K are randomly taken on each fracture, approximately ranging from 10^{-1} to 10^2. The geometry of the DFN and a mesh example are reported in Figure 2.17 for the case 11F. In Figures 2.18-2.23 and 2.25-2.27 we report for each system considered and for a selected fracture F_i: i) the solution h_i on the fracture; ii) the restriction on the traces of h_i and of the solution h_j obtained on the fracture F_j which generates the trace through its intersection with F_i; iii) the control variables u_i and $-u_j$. All the results here reported are obtained with a grid parameter $\delta_{\max} = 0.16$. As shown in particular in the 2D plots, the computed numerical solution well approximates continuity and flux conservation (2.2)-(2.3). Focusing on the intermediate 11F case, in Figure 2.24 we plot, for each trace, the L^2-norm of the difference of the hydraulic head on intersecting fractures, $||h_{i|S} - h_{j|S}||$, relative to the average L^2-norm of h on the trace, $h_{av} = 1/2\left(||h_{i|S}|| + ||h_{j|S}||\right)$ (triangular markers), and in square markers flux unbalance at traces, $||u_i + u_j||$, relative to the average flux $u_{av} = 1/2\,(||u_i|| + ||u_j||)$. It can be seen that the relative mismatches in flux conservation and head continuity are small and roughly of the same order. Furthermore, in Table 2.2 we report, again for problem 11F, the flux unbalance and the total flux on each fracture, which are computed on F_i, $i = 1, ..., 11$, as $\sum_{S\in\mathcal{S}_i} \int_S u_i^S + u_j^S \, d\gamma$ and $\sum_{S\in\mathcal{S}_i} \int_S u_i^S \, d\gamma$, respectively. The sum of the flux unbalances on all the DFN is -5.0114e-7, and, clearly, the sum of the total fluxes on the fractures exactly match this value. It can be seen from the table that flux unbalance on the fractures is quite small, being six orders of magnitude below the respective total flux.

Table 2.3: System matrices data. *Dim*: matrix dimension, *NCond*: matrix condition number, *Iter*: iterative solver number of iterations, *Relres*: solution relative residual

Problem	Dim	NCond	SYMMLQ Iter	Relres	Grid Prameter
S1	8324	$1.9 \cdot 10^6$	3000	$1.75 \cdot 10^{-1}$	0.1
S2	15067	$9.0 \cdot 10^9$	3000	$1.25 \cdot 10^{-1}$	0.1
7F	18261	$1.3 \cdot 10^9$	3000	1	0.16
11F	32888	$1.7 \cdot 10^{10}$	3000	1	0.16
50F	69476	$9.3 \cdot 10^9$	3000	1	0.22

2.5.4 Preconditioning

The choice of a good preconditioner for SYMMLQ is a crucial task as the linear systems arising from the discrete DFN-like problems are ill-conditioned even for the smaller problems considered, and conditioning worsens both if grid parameter is reduced and if the number of fractures increases. In Table 2.3 we report the data related to the conditioning of the system for various problems considered, along with the results obtained while attempting to solve the non preconditioned linear system with SYMMLQ. Problems 7F, 11F, 50F refer to the examples shown in Subsection 2.5.3 while Problems S1 and S2 are a modified version of Problems 7F, 11F respectively. With reference to Figure 2.14 and Figure 2.15 z-quotes are reduced in Problems S1 and S2 to $z = 1$ for the fractures represented with solid lines and to $z = 0.5$ for the fracture in dashed line. Different Dirichlet boundary conditions are set on fracture edges in the z-direction marked with a black dot, while homogeneous Dirichlet boundary conditions are placed on the remaining edges. Finally a constant value $K = 1$ is prescribed to all the fractures. These modified problems yield smaller linear systems. The data in Table 2.3 show that the iterative solver never succeeded in reaching the required exit tolerance $tol = 10^{-6}$ within the maximum number of iterations allowed ($\max_{it} = 3000$).

In order to precondition the system, we follow here the approach described in [27], in which a block triangular preconditioner is suggested for linear systems of saddle point type arising from general QP problems. In detail, for a saddle point problem of the form (7.22), the following preconditioner is suggested:

$$\mathcal{P} = \begin{pmatrix} G + C^T W^{-1} C & kC^T \\ 0 & W \end{pmatrix} \tag{2.18}$$

where k is a scalar and W is a $N^F \times N^F$ symmetric positive definite weight matrix. A suitable choice for k and W suggested in [27] is $k = 0$ and $W = \gamma I$ where I is the identity matrix and $\gamma > 0$ is a given constant which should provide an augmenting term $C^T W^{-1} C$ not too small in comparison with G. We remark that the choice $k = 0$ yields a block diagonal symmetric preconditioner, hence suitable for the use along with SYMMLQ solver.

The preconditioner (2.18) is introduced in [27] in the context of interior point methods for optimization problems, which expecially in the case of inexact methods [2] heavily rely on iterative methods and hence on good preconditioners. In the case of interior point methods, at each outer iteration a linear system with a structure similar to (7.22) has to be solved, with the block G being typically more and more ill-conditioned as the solution is approached. In [27], an adaptive choice of γ along outer iterations appears to be an effective choice: when used in conjunction with MINRES solver, an effective choice is $\gamma = 1/\max(G)$ for linear programming problems, and for quadratic programming problems the choice suggested is given by $\gamma = \|C\|^2/\|G\|$.

Since here we deal with a different context and the block G is not necessarily the major source of ill-conditioning, a preliminary investigation has been performed on Problems S1, S2, 7F, 11F, 50F in order to study effectiveness of the preconditioner in our applications, and, possibly, identify a suitable value for the parameter γ. A broad range of values for γ has been considered, ranging between 10^{-9} and 300, which roughly corresponds to the optimal value $\|C\|^2/\|G\|$ suggested in [27] applied to problems S1 and S2 (for problems 7F,11F,50F this value corresponds to $\approx 7 \cdot 10^5$). Exit tolerance for iterative solver is now set to $tol = 10^{-12}$ and the maximum number of iterations is set to $\max_{it} = 3000$. We point out that the implementation of SYMMLQ that we used for solving the system $\mathcal{A}x = q$, performs the check on the exit tolerance on the unpreconditioned relative residual $\|q - \mathcal{A}x\|/\|q\|$ even if the linear system is preconditioned. Results of this preliminary investigation are reported in Figures 2.28 and 2.29. In particular, in Figure 2.28 we report the number of iterations required by SYMMLQ for several values of γ. As shown in the Figure, in all problems considered for γ small enough the iterative solver succeeded in satisfying the stopping criterion within a very moderate number of iterations. The value $\gamma = 10^{-7}$ appears to ensure the best performance in the preconditioner, for all the considered problems, independently of the number of fractures, of the number of unknowns, and of the boundary conditions. Indeed, Figure 2.29 shows that for optimal γ-values the condition number of the preconditioned linear system reaches

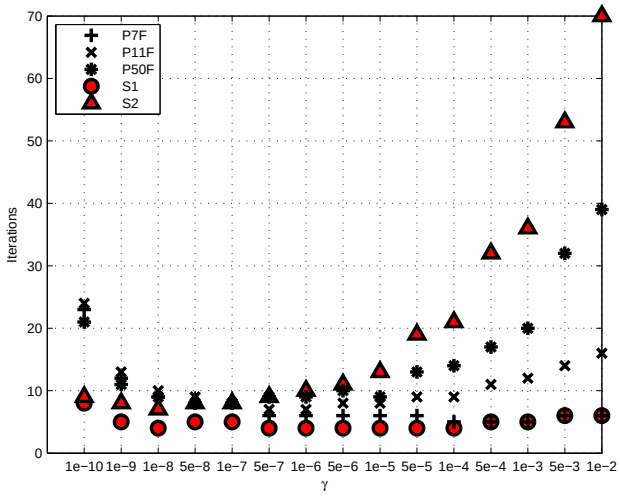


Figure 2.28: SYMMLQ number of iterations versus γ

very low values, and matrix eigenvalues cluster around the values $\{-1, 1\}$.

2.5.5 Large variation of K values

In previous computations we allowed a different transmissivity value K_i on each fracture F_i, $i \in \mathfrak{I}$ (assuming for simplicity K_i constant on the fracture). In real applications, large variations in the (typically very small) values of K_i may occur, from fracture to fracture, possibly spanning several orders of magnitude. This may correspondingly cause a large variation in the orders of magnitude of U, which, representing the co-normal derivative $n^T K \nabla H$, may largely differ from those of H, making the functional J less sensitive to variation in U. In order to deal with this situation, a possible approach consists in properly weighting the terms $\|U_i^S + U_j^S\|$ in the functional, allowing the following modification to J:

$$J(U) = \sum_{S \in \mathcal{S}} J^S(U) = \sum_{S \in \mathcal{S}} \left(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{H}^S} + \frac{1}{(K_{\min}^S)^\alpha} ||U_i^S + U_j^S||^2_{\mathcal{U}^S} \right)$$

where $K_{\min}^S = \min\{K_i,\ K_j\}$ and e.g. $\alpha = 1, 2$. The weights introduced help in balancing the contribution of the various terms of the cost functional, giving more relevance to flux unbalance when large variations of transmissivity occur at intersecting fractures. The following model problem has been used to show the effectiveness of this extension of the method, here applied with $\alpha = 1$. Problem domain is shown in the left of

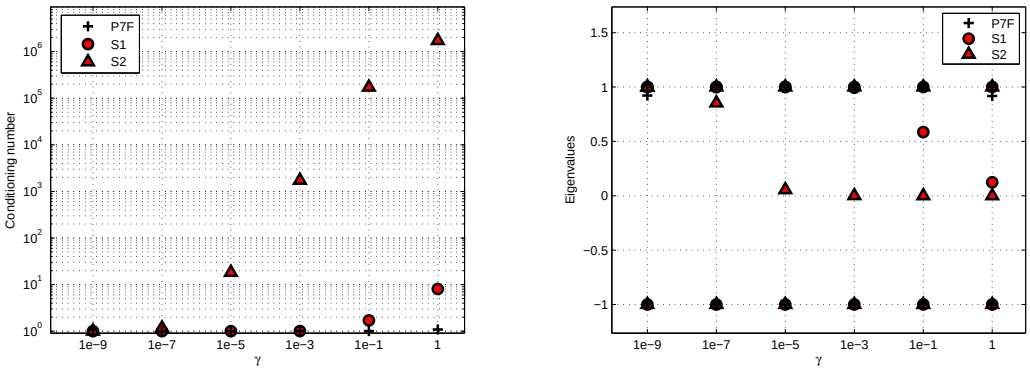
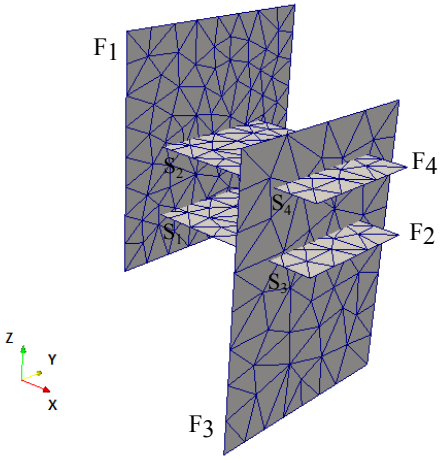


Figure 2.29: Preconditioned system: condition number (left) and eigenvalues (right) versus γ.

Figure 2.30, along with fracture and trace numbering. Fracture F_1 carries a constant value Dirichelet boundary condition $h = 10$ on the top border along the y-axis, while fracture F_3 has a Dirichelet boundary condition $h = 3$ on the bottom border parallel to the y-axis. Fractures F_2 and F_4 have a constant value $h = 1$ Dirichelet boundary condition on the left border parallel to the y-axis. An homogeneous Neumann boundary condition is prescribed on the remaining borders of all fractures. Four different simulations are performed with different sets of fracture transmissivity values as reported in the right of Figure 2.30. It was noted that, with these broad variations of K, the correction helped in obtaining the solution, as we experienced difficulties in convergence of the steepest descent method with the non-modified functional. Results concerning hydraulic head mismatch at traces and flux unbalance are collected in Figures 2.31-2.32. In Figure 2.31 the $\mathrm{L}^2(S)$-norm of the difference of the hydraulic head on intersecting fractures $E_h = ||h_{i|S} - h_{j|S}||$ is reported with solid markers for each trace, along with the average $\mathrm{L}^2(S)$-norm of h, $h_{av} = 1/2\left(||h_{i|S}|| + ||h_{j|S}||\right)$ (in empty markers), in order to compare the mismatch of h at the intersections in relation with the order of magnitude of the solution. Similarly in Figure 2.32 we show flux unbalance at traces in solid markers, $E_u = ||u_i + u_j||$, with the average flux $u_{av} = 1/2\left(||u_i|| + ||u_j||\right)$, in empty markers. It is noticed that the hydraulic head mismatch on traces and flux unbalance are usually orders of magnitude lower than the hydraulic head and flux, respectively, also for fracture transmissivities differing for six orders of magnitude.



case	F_1	F_2	F_3	F_4
$\mathbf{K}_a$	10^{-6}	10^{-6}	10^{-6}	1
$\mathbf{K}_b$	10^{-6}	1	10^{-6}	1
$\mathbf{K}_c$	10^{-3}	1	10^{-3}	10^{-6}
$\mathbf{K}_d$	10^{-3}	1	10^{-3}	1

Figure 2.30: Domain description and fracture transmissivity values

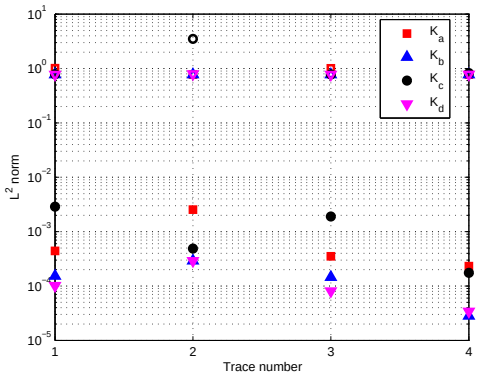


Figure 2.31: L^2-norm of hydraulic head mismatch E_h (filled markers) and average L^2-norm of solution h_{av} (empty markers) on the traces of the system

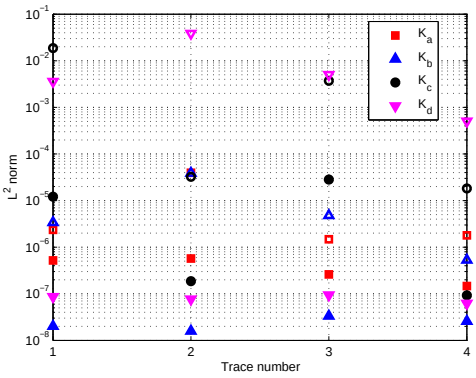


Figure 2.32: L^2-norm of flux unbalance E_u (filled markers) and average L^2-norm of fluxes u_{av} (empty markers) on the traces of the system

2.6 Conclusions

In this paper we have further analyzed the viability in complex systems of a novel method introduced in [7] for the problem of subsurface flow in a system of fractures, which consists in the reformulation of the problem as a PDE constrained optimization problem. Independent meshing processes have been used on the fractures, generating grids which are independent of the mesh on other fractures and of trace number and disposition. This is a crucial point since one of major difficulties in the DFN approach is typically the generation of a trace-matching mesh. The discussion and the experiments here reported show effectiveness of the method in providing good approximation of the solution in complex DFNs.

In future works, more realistic DFN configurations will be investigated. A parallel implementation exploiting the independence of the problems on the sub-fractures is also envisaged. Moreover, we will investigate also the applicability of the method to non steady-state case in conjunction with local time adaptive strategies as in [6].

Bibliography

[1] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[2] S. Bellavia and S. Pieraccini, *Convergence analysis of an inexact infeasible interior point method for semidefinite programming*, Comput. Optim. Appl., 29 (2004), pp. 289–313.

[3] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, Internat. J. Numer. Methods Engrg., 45 (1999), pp. 601–620.

[4] T. Belytschko, N. Möes, S. Usui, and C. Parimi, *Arbitrary discontinuities in finite elements*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 993–1013.

[5] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[6] S. Berrone, *A local-in-space-timestep approach to a finite element discretization of the heat equation with a posteriori estimates*, SIAM J. Numer. Anal., 47 (2009), pp. 3109–3138.

[7] S. Berrone, S. Pieraccini, and S. Scialó, *A PDE-constrained optimization formulation for discrete fracture network flows.* Submitted, SISC Manuscript # 86588R, 2012.

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

[9] J. Chessa, H. Wang, and T. Belytschko, *On the construction of blending elements for local partition of unity enriched finite elements*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 1015–1038.

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

[11] C. Daux, N. Möes, J. Dolbow, N. Sukumar, and T. Belytschko, *Arbitrary branched and intersecting cracks with the extended finite element method*, Internat. J. Numer. Methods Engrg., 48 (2000), pp. 1741–1760.

[12] J.-R. de Dreuzy, G. Pichot, B. Poirriez, and J. Erhel, *Synthetic benchmark for modeling flow in 3D fractured media*, Computers and Geosciences, 50 (2013), pp. 59–71.

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

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

[15] J. Erhel, J.-R. de Dreuzy, and B. Poirriez, *Flow simulation in three-dimensional discrete fracture networks*, SIAM J. Sci. Comput., 31 (2009), pp. 2688–2705.

[16] T.-P. Fries, *A corrected xfem approximation without problems in blending elements*, Internat. J. Numer. Methods Engrg., 75 (2008), pp. 503–532.

[17] T.-P. Fries and T. Belytschko, *The extended/generalized finite element method: an overview of the method and its applications*, Internat. J. Numer. Methods Engrg., 84 (2010), pp. 253–304.

[18] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[19] P. Laborde, J. Pommier, Y. Renard, and M. Salaün, *High-order extended finite element method for cracked domains*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 354–381.

[20] N. Möes, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing*, Internat. J. Numer. Methods Engrg., 46 (1999), pp. 131–150.

[21] H. Mustapha and K. Mustapha, *A new approach to simulating flow in discrete fracture networks with an optimized mesh*, SIAM J. Sci. Comput., 29 (2007), pp. 1439–1459.

[22] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[23] A. W. Nordqvist, Y. W. Tsang, C. F. Tsang, B. Dverstop, and J. Andersson, *A variable aperture fracture network model for flow and transport in fractured rocks*, Water Resource Res., 28 (1992), pp. 1703–1713.

[24] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[25] G. Pichot, J. Erhel, and J.-R. De Dreuzy, *A mixed hybrid Mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629–1643.

[26] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[27] T. Rees and C. Greif, *A preconditioner for linear systems arising from interior point optimization methods*, SIAM J. Sci. Comput., 29 (2007), pp. 1992–2007.

[28] J. R. Shewchuk, *Delaunay refinement algorithms for triangular mesh generation*, Computational Geometry: Theory and Applications, 22 (2002), pp. 21–74.

[29] N. Sukumar, D. Chopp, N. Möes, and T. Belytschko, *Modeling holes and inclusions by level sets in the extended finite element method*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 1209–1233.

[30] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.

Chapter 3

An optimization approach for large scale simulations of discrete fracture network flows

Abstract In recent papers [7, 6] the authors introduced a new method for simulating subsurface flow in a system of fractures based on a PDE-constrained optimization reformulation, removing all difficulties related to mesh generation and providing an easily parallel approach to the problem. In this paper we further improve the method removing the constraint of having on each fracture a non empty portion of the boundary with Dirichlet boundary conditions. This way, Dirichelet boundary conditions are prescribed only on a possibly small portion of DFN boundary. The proposed generalization of the method in [7, 6] relies on a modified definition of control variables ensuring the non-singularity of the operator on each fracture. A conjugate gradient method is also introduced in order to speed up the minimization process.

3.1 Introduction

Efficient numerical simulation of underground flow is of great interest in a large variety of practical applications, as for example enhanced oil/gas recovery, pollutant percolation and diffusion in aquifers, or carbon dioxide storage. The underground fluid flow is a multi-scale heterogeneous phenomenon, occurring in complex geological configurations usually characterized by networks of fractures surrounded by a porous rock

matrix. The Discrete Fracture Network (DFN) approach models underground systems of fractures as 3D networks of intersecting discrete planar fractures. Diffusive phenomena in this system of fractures are governed by the Darcy law. At fracture intersections, called *traces*, mass balance and pressure continuity are preserved. The geological characteristics of the fractures, such as size, orientation, aspect ratio, density, permeability, are usually determined relying on stochastic data [10], and only probability distribution of data are usually available for a specific geological site. A huge number of numerical simulations is then necessary in order to perform sensitivity analysis to the variability of the involved parameters. On the other hand, DFN simulations are very demanding from a computational point of view. Problem size is usually huge, involving a very large number of fractures. Moreover, for intricate fracture geometries, the generation of a good quality finite element triangulation conforming to the traces usually requires the introduction of many unknowns on each fracture, independently of the quality required for the numerical solution.

Many approaches are suggested in literature to circumvent these difficulties. A method based on a conforming mesh with mixed non-conforming finite elements is proposed in [21], while in other cases modifications of the geometry or of the mesh are introduced in order to preserve conformity and achieve a good quality mesh, such as in [14, 21] or in [12]. A different approach is suggested in [17], where the solution in the fractures is expressed as a function of the solution at the intersections. In other works it is suggested to rely on mortar methods to ease meshing procedure, as for example in [19, 20]: with this approach the mesh conformity constraint is relaxed but fracture meshes have to be aligned along the traces. In [8, 18, 11] the DFN is reduced into a system of mono-dimensional pipes connecting the traces with the surrounding fractures both preserving fracture topology and mitigating meshing related problems.

The present work further develops the approach introduced in [7, 6], in which the problem of the computation of the hydraulic head in a DFN is reformulated as a PDE-constrained optimization problem. The overall problem is split in a set of several independent sub-problems on each fracture of the system, coupled by the minimization of a proper functional. The use of Extended Finite Elements allows to capture the correct behaviour of the solution along traces even if grids are not conforming along fracture intersections and traces arbitrarily cut mesh elements. This way the meshes may be generated on each fracture in a completely independent way, disregarding fracture intersections and thus eliminating meshing difficulties.

Despite being applicable to very general DFN configurations, the formulation of the problem in the over-mentioned approach requires a non empty portion of Dirichelet boundary on each fracture of the system. In the present work a modification of the control variable and of the cost functional involved in the optimization problem is introduced, eliminating this constraint and allowing to prescribe Dirichelet boundary conditions only on (portion of) boundaries of a – possibly very small – subset of fractures. The use of a conjugate gradient method for the minimization process is also described. The behaviour of the method on fairly complex networks is shown through several numerical experiments.

The paper is organized as follows. In Section 3.2 we recall the physical model and the mathematical statement of the continuous problem introduced in [7, 6]. In Section 3.3 the PDE-constrained optimization problem is described along with the conjugate gradient algorithm used in the minimization process. Application of XFEM ideas to the DFN context is briefly accounted for in Section 3.4. In Section 3.5 we introduce the discrete version of the algorithm. Numerical experiments showing effectiveness of the method are reported and commented in Section 3.6.

3.2 Description of the problem

3.2.1 Problem formulation

Our target is the computation of the hydraulic head $H = \mathcal{P} + \zeta$ (being $\mathcal{P} = p/(\varrho g)$ the pressure head, p the fluid pressure, g the gravitational acceleration constant, ϱ the fluid density, ζ the elevation) in a DFN given by the union of a set of fractures. Let us model each fracture as an open planar polygonal set, F_i, with index i varying in a set $\mathfrak{I}$. Let us also introduce on each fracture a 2D local coordinate system $\hat{x}_i$. Let Ω be the 3D set

$$\Omega = \bigcup_{i \in \mathfrak{I}} F_i,$$

and $\partial\Omega$ the boundary of Ω, split as usual in a set $\Gamma_D \neq \emptyset$ with Dirichlet boundary conditions and a set Γ_N with Neumann boundary conditions, such that $\Gamma_D \cup \Gamma_N = \partial\Omega$ and $\Gamma_D \cap \Gamma_N = \emptyset$.

Note that the intersection of the closure of each couple of fractures is either an empty set or a set of non vanishing segments called *traces*, denoted by S_m, with index

m varying in an index set $\mathfrak{M}$ with cardinality $\sharp\mathfrak{M}$. For each fracture F_i, $\mathcal{S}_i$ is the set of traces shared by F_i and other fractures while $\mathcal{S}$ indicates the set of all the traces.

In the paper the following is assumed on the DFN: 1) $\bar{\Omega}$ is a connected set; 2) each trace S_m is shared by exactly two polygonal fractures F_i and F_j, $i \neq j$, such that $S_m \subseteq \bar{F}_i \cap \bar{F}_j$. The set of the two indices i and j of the fractures F_i and F_j sharing trace S_m is denoted by $I_{S_m} = \{i, j\}$, while for all $i \in \mathfrak{I}$, the subset $J_i \subset \mathfrak{I}$ contains indices of fractures with a non-empty intersection with F_i.

While referring the reader to [7] for more details, here we briefly recall the variational formulation of the problem. Let us define $\forall i \in \mathfrak{I}$ 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' their dual spaces. The hydraulic head H_i in each fracture belongs to the space

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

where H_i^D is the restriction of the Dirichlet boundary condition $H_{|\Gamma_D} = H^D$ to $\Gamma_{iD} = \Gamma_D \cap \partial F_i$. In what follows Γ_{iD} can be an empty set, but $\Gamma_D = \bigcup_i \Gamma_{iD} \neq \emptyset$.

Let $\mathbf{K}_i(\hat{x}_i)$ be, for all $i \in \mathfrak{I}$, a symmetric and uniformly positive definite tensor called hydraulic conductivity tensor, which we assume dependent on the position and possibly different on each fracture. As documented in [7], the global hydraulic head H in the whole system Ω is obtained solving the following problems $\forall i \in \mathfrak{I}$, which model the diffusion of the hydraulic head on each fracture: find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\int_{F_i} \mathbf{K}_i \nabla H_i \nabla v d\Omega = \int_{F_i} q_i v d\Omega + \int_{\Gamma_N \cap \partial F_i} G_i^N v_{|S} d\Gamma + \sum_{S \in \mathcal{S}_i} \int_S \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S v_{|S} d\Gamma, \quad (3.1)$$

where G_i^N is the restriction to $\Gamma_{iN} = \Gamma_N \cap \partial F_i$ of the Neumann boundary condition G^N imposed on Γ_N. The quantity $\frac{\partial H_i}{\partial \hat{\nu}_S^i} = (\hat{n}_S^i)^T \mathbf{K}_i \nabla H_i$ is the outward co-normal derivative of the hydraulic head, being $\hat{n}_S^i$ the unit vector normal to the trace S. The symbol $\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S$ denotes the jump of the co-normal derivative along $\hat{n}_S^i$, being this jump independent of the orientation of $\hat{n}_S^i$. According to (7.1), the diffusion of H_i is contributed by the following terms: the external load in each facture (first term of the right hand side); the Neumann boundary conditions (second term); the net flow of hydraulic head entering in the fracture at each trace (last term).

Equations (7.1) are coupled by the following matching conditions, which prescribe global continuity of the hydraulic head and conservation of hydraulic fluxes across each

trace S_m, $m \in \mathfrak{M}$:

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i,j \in I_{S_m}, \tag{3.2}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_{S_m}}\right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_{S_m}}\right]\!\right]_{S_m} = 0, \qquad \text{for } i,j \in I_{S_m}. \tag{3.3}$$

Note that due to condition (7.2) the hydraulic head H on the whole domain Ω belongs to the space

$$V^D = \mathrm{H}^1_{\mathrm{D}}(\Omega) = \left\{ v \in \prod_{i\in\mathfrak{I}} V_i^D : (v_{|F_i})_{|S_m} = (v_{|F_j})_{|S_m},\ i,j \in I_{S_m},\ \forall m \in \mathfrak{M} \right\}. \tag{3.4}$$

3.3 Optimization approach

Following the approach described in [7], instead of solving the coupled differential problems on the fractures (7.1) $\forall i \in \mathfrak{I}$ with the corresponding matching conditions (7.2), (7.3), we introduce a PDE-constrained optimization problem. In order to ease notation and for a concise and clear description, in the following of this Section we assume that the traces $S \in \mathcal{S}$ are disjoint, recalling that as stated in [7], this assumption can be dropped replacing occurrences of each single trace S with the union of connected traces. Further, in our discrete formulation the assumption naturally drops thanks to the choice of the functional spaces (see again [7]). Let us introduce for each trace $S \in \mathcal{S}$ a suitable space $\mathcal{U}^S$ and its dual $(\mathcal{U}^S)'$. Similar spaces are introduced on the set of traces belonging to a fracture F_i, $\forall i \in \mathfrak{I}$, and on the full set of traces $\mathcal{S}$:

$$\mathcal{U}^{\mathcal{S}_i} = \prod_{S\in\mathcal{S}_i} \mathcal{U}^S, \qquad \mathcal{U} = \prod_{i\in\mathfrak{I}} \mathcal{U}^{\mathcal{S}_i}.$$

Now, let us fix a trace S and let $S \subseteq \bar{F}_i \cap \bar{F}_j$. We introduce suitable variables $U_i^S, U_j^S \in \mathcal{U}^S$ which will act as control variables, defined as $U_i^S = \alpha H_{i|_S} + \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_S}\right]\!\right]_S$ and $U_j^S = \alpha H_{j|_S} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_S}\right]\!\right]_S$ respectively, where α is a positive fixed parameter. This generalizes the approach proposed in [7] where U_i^S is set equal to flux jump, thus allowing $\Gamma_{iD} = \emptyset$ on possibly all but one fractures. We set

$$U_i = \prod_{S\in\mathcal{S}_i} U_i^S \in \mathcal{U}^{\mathcal{S}_i}, \qquad U = \prod_{i\in\mathfrak{I}} U_i \in \mathcal{U},$$

i.e. U_i is the tuple of functions U_i^S with $S \in \mathcal{S}_i$, and U is the $2(\#\mathfrak{M})$-tuple of control functions on all traces in $\bar{\Omega}$.

We also introduce the Riesz isomorphisms $\Lambda_{\mathcal{U}^S} : \mathcal{U}^S \to \mathcal{U}^{S\prime}$, $\Lambda_{\mathcal{U}^{\mathcal{S}_i}} : \mathcal{U}^{\mathcal{S}_i} \to \mathcal{U}^{\mathcal{S}_i\prime}$ and $\Lambda_{\mathcal{U}} : \mathcal{U} \to \mathcal{U}'$ and the following linear bounded operators and their duals:

$$\begin{aligned}
A_i \in \mathcal{L}(V_i, V_i'), \qquad & \langle A_i H_i^0, v\rangle_{V_i',V_i} = (\mathbf{K}\,\nabla H_i^0, \nabla v) + \alpha \left(H_{i\,|\mathcal{S}_i}^0, v_{|\mathcal{S}_i}\right)_{\mathcal{S}_i}, \\
A_i^D \in \mathcal{L}(V_i^D, V_i'), \qquad & \langle A_i^D\, \mathcal{R}_i\, H_i^D, v\rangle_{V_i',V_i} = (\mathbf{K}\,\nabla\, \mathcal{R}_i\, H_i^D, \nabla v) \\
& \qquad\qquad +\alpha \left((\mathcal{R}_i\, H_i^D)_{|\mathcal{S}_i}, v_{|\mathcal{S}_i}\right)_{\mathcal{S}_i}, \\
B_i^S \in \mathcal{L}(\mathcal{U}^S, V_i'), \qquad & \langle B_i^S U_i, v\rangle_{V_i',V_i} = \langle U_i^S, v_{|S}\rangle_{\mathcal{U}^S,\mathcal{U}^{S\prime}}, \\
B_i = \prod_{S\in\mathcal{S}_i} B_i^S \in \mathcal{L}(\mathcal{U}^{\mathcal{S}_i}, V_i'), \qquad & \langle B_i U_i, v\rangle_{V_i',V_i} = \langle U_i, v_{|\mathcal{S}_i}\rangle_{\mathcal{U}^{\mathcal{S}_i},\mathcal{U}^{\mathcal{S}_i\prime}},
\end{aligned}$$

with $H_i^0 \in V_i$, $H_i^D \in V_i^D$, $v \in V_i$, and the operator $\mathcal{R}_i$ is the lifting of the Dirichlet boundary conditions on Γ_{iD} if not empty. Dual operators are $A_i^* \in \mathcal{L}(V_i, V_i')$,

$$C_i^S = (B_i^S)^* \in \mathcal{L}(V_i, \mathcal{U}^{S\prime}), \quad C_i = (B_i)^* \in \mathcal{L}(V_i, \mathcal{U}^{\mathcal{S}_i\prime}).$$

The operator $B_{iN} \in \mathcal{L}(\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), V_i')$ imposing Neumann boundary conditions is defined such that

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

Problems (7.1) can now be written as follows: $\forall i \in \mathfrak{I}$, 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

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_i^N - A_i^D\, \mathcal{R}_i\, H_i^D, \quad in\ F_i. \tag{3.5}$$

We remark that, if $\alpha > 0$, the solution H_i to (7.6) exists and is unique for a non isolated fracture even if we set Neumann boundary conditions on the whole ∂F_i.

We can now define the differentiable functional $J : \mathcal{U} \to \mathbb{R}$ as

$$\begin{aligned}
J(U) &= \sum_{S\in\mathcal{S}} J^S(U) \\
&= \sum_{S\in\mathcal{S}} \Big(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{U}^{S\prime}} \\
&\qquad +||U_i^S - \alpha\Lambda^{-1}_{\mathcal{U}^S} C_i^S H_i(U_i) + U_j^S - \alpha\Lambda^{-1}_{\mathcal{U}^S} C_j^S H_j(U_j)||^2_{\mathcal{U}^S}\Big) \\
&= \frac{1}{2}\sum_{i\in I}\sum_{S\in\mathcal{S}_i} \Big(||C_i^S H_i(U_i) - C_j^S H_j(U_j)||^2_{\mathcal{U}^{S\prime}} \\
&\qquad +||U_i^S - \alpha\Lambda^{-1}_{\mathcal{U}^S} C_i^S H_i(U_i) + U_j^S - \alpha\Lambda^{-1}_{\mathcal{U}^S} C_j^S H_j(U_j)||^2_{\mathcal{U}^S}\Big)
\end{aligned}$$

$$= \frac{1}{2}\sum_{i\in I} || \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j)\right) ||^2_{\mathcal{U}^{\mathcal{S}_i\prime}}$$

$$+\frac{1}{2}\sum_{i\in I} ||U_i + \prod_{S\in\mathcal{S}_i} U_j^S - \alpha\Lambda^{-1}_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j)\right) ||^2_{\mathcal{U}^{\mathcal{S}_i}}, \tag{3.6}$$

where quantity $\prod_{S\in\mathcal{S}_i}\left(C_i^S H_i(U_i) \pm C_j^S H_j(U_j)\right)$ denotes the tuple of functions $\left(C_i^S H_i(U_i) \pm C_j^S H_j(U_j)\right)$ with $S \in \mathcal{S}_i$, and $i, j \in I_S$. Moreover $H_\ell(U_\ell)$ denotes the solution of (7.6) corresponding to the control variable U_ℓ, $\ell = i, j$.

Proposition 3.1. *Setting $\mathcal{U}^S = H^{-\frac{1}{2}}(S)$ and letting $C_i^S \in \mathcal{L}(V_i, H^{\frac{1}{2}}(S))$ be the trace operator, there exists a unique control variable U vanishing the functional $J(U)$ and a corresponding unique solution H satisfying problems (7.6) $\forall i \in \mathfrak{I}$ that is solution to* (7.1)-(7.3).

Proof. We sketch very briefly the proof as it follows from classical arguments. Resorting to the classical formulation of the problem on sub-fractures as recalled in [7], it can be proven that exists a unique solution $H \in V^D$ for the hydraulic head on the DFN satisfying (7.1), $\forall i \in \mathfrak{I}$, and (7.2), (7.3), $\forall m \in \mathfrak{M}$, that are trivially equivalent to (7.6), $\forall i \in \mathfrak{I}$, and to

$$H_{i|S_m} - H_{j|S_m} = 0,\ U_i^S - \alpha H_{i|_S} + U_j^S - \alpha H_{j|_S} = 0,\ \text{for } i, j \in I_{S_m},\ \forall m \in \mathfrak{M}. \tag{3.7}$$

As in [7], since the vanishing of the two terms of the functional J is equivalent to the imposition of the matching conditions (3.7), the thesis follows. □

Based on previous discussion, problems (7.6) coupled with (3.7) are replaced by the following optimization problem:

$$\min J(U) \quad \text{subject to (7.6)},\ \forall i \in \mathfrak{I}. \tag{3.8}$$

In the following result we state optimality conditions for (3.8).

Proposition 3.2. *The optimal control $U \in \mathcal{U}$ satisfying* (3.8) *is given by the system of conditions* (7.6) *and*

$$B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \left(U_i + \prod_{S\in\mathcal{S}_i} U_j^S\right) - \alpha \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j)\right) = 0, \tag{3.9}$$

$\forall i \in \mathfrak{I}$, *where the functions $P_i \in V_i$ are the solution of equation*

$$A_i^* P_i = C_i{}^* \Lambda^{-1}_{\mathcal{U}^{\mathcal{S}_i}} \Bigg[\prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j)\right)$$

$$+\alpha^2 \prod_{S\in\mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j)\right)\Bigg] - \alpha C_i{}^* \left(U_i + \prod_{S\in\mathcal{S}_i} U_j^S\right), \quad \text{in } F_i, \tag{3.10}$$

in which homogeneus Dirichlet and Neumann boundary conditions on Γ_{iD} and Γ_{iN}, respectively, are prescribed.

Proof. Let us differentiate the cost functional with respect to the control variable U_i:

$$
\begin{aligned}
&J'(U)(v_i - U_i) = \sum_{S\in\mathcal{S}_i} J^{S\prime}(U_i)(v_i - U_i) \\
&= 2\sum_{S\in\mathcal{S}_i}\Big[\big(C_i^S H_i(U_i) - C_j^S H_j(U_j), C_i^S(H_i(v_i) - H_i(U_i))\big)_{\mathcal{U}^{S\prime}} \\
&+\Big(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), v_i^S - U_i^S \\
&\qquad\qquad -\alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(v_i) - C_i^S H_i(U_i))\Big)_{\mathcal{U}^S}\Big] \\
&= 2\left\langle C_i{}^*\Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1}\prod_{S\in\mathcal{S}_i}(C_i^S H_i(U_i) - C_j^S H_j(U_j)), H_i(v_i) - H_i(U_i)\right\rangle_{V_i',V_i} \\
&+2\left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), v_i - U_i\right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime},\mathcal{U}^{\mathcal{S}_i}} \\
&-2\alpha\left\langle C_i^*\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), H_i(v_i) - H_i(U_i)\right\rangle_{V_i',V_i} \\
&= 2\left\langle A_i^* P_i, A_i^{-1}B_i(v_i - U_i)\right\rangle_{V_i',V_i} \\
&+2\left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}}\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), v_i - U_i\right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime},\mathcal{U}^{\mathcal{S}_i}} \\
&= 2\left\langle B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}}\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), v_i - U_i\right\rangle_{\mathcal{U}^{\mathcal{S}_i\prime},\mathcal{U}^{\mathcal{S}_i}}.
\end{aligned}
$$

Thus, the vanishing of this last term yields (3.9). □

Instead of solving equations (7.6), (3.9), (7.9), we set up a minimization process for problem (3.8). This is organized in such a way that only the decoupled solution of the local problems (7.6) is needed. Here we use the Fletcher and Reeves conjugate gradient method [16]. Let us denote by $\nabla J(U_i)$ the Frechet derivative of the functional J with respect to the control variables on the fracture F_i, $\forall i \in \mathfrak{I}$, and by $\nabla J(U)$ the whole derivative:

$$
\begin{aligned}
\nabla J(U_i) &= B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}}\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j))), \quad (3.11)\\
\nabla J(U) &= \prod_{i\in I}\nabla J(U_i). \quad (3.12)
\end{aligned}
$$

The method used is depicted in the following algorithm.

Algorithm 3.1.

1. Set $k = 0$ and initial guess for control variable U^0;
2. find $H^0 = H(U^0)$ solving on each fracture the primal problem (7.6);
3. find $P(U^0)$ solving on each fracture the dual problem (7.9);
4. evaluate $\nabla J(U^0)$ by equation (7.11);
5. set $(\delta U)^0 = -\Lambda_{\mathcal{U}}^{-1} \nabla J(U^0)$;
6. *While* $J(U^k) \neq 0$ *do:*

   6.1. choose a step-size λ^k along direction $(\delta U)^k$;

   6.2. set $U^{k+1} = U^k + \lambda^k (\delta U)^k$;

   6.3. $\forall i \in \mathfrak{I}$ solve primal problem (7.6) to find $H_i(U^{k+1})$;

   6.4. $\forall i \in \mathfrak{I}$ solve dual problem (7.9) to find $P_i(U^{k+1})$;

   6.5. evaluate $\nabla J(U^{k+1})$ by (7.11);

   6.6. set $\beta^{k+1} = \|\nabla J(U^{k+1})\|^2_{\mathcal{U}'} / \|\nabla J(U^k)\|^2_{\mathcal{U}'}$;

   6.7. set $(\delta U)^{k+1} = -\Lambda_{\mathcal{U}}^{-1} \nabla J(U^{k+1}) + \beta^{k+1} \delta U^k$;

   6.8. $k = k + 1$;

   end do.

Let us evaluate the optimal step-size λ which can be used in the previous algorithm at steps 6.1-6.2. Given a variation δU_i for the control variable on each fracture F_i and $\delta U = \sum_{i \in \mathfrak{I}} \delta U_i$, let $\delta H_i \in V_i$, $\forall i \in \mathfrak{I}$, be defined as the solution of the problem

$$A_i \delta H_i = B_i \delta U_i, \quad \text{in } F_i, \tag{3.13}$$

corresponding to homogeneous Dirichlet and Neumann boundary conditions on Γ_{iD} (if non-empty) and Γ_{iN}, respectively.

Proposition 3.3. *Let us increment the control variable U of a step $\lambda \delta U$, the optimal step-size λ is*

$$\lambda = -\langle \nabla J(U), \delta U \rangle_{\mathcal{U}',\mathcal{U}} \left\{ \sum_{S \in \mathcal{S}} \left(||C_i^S \delta H_i - C_j^S \delta H_j||^2_{\mathcal{U}^{S'}} + ||\delta U_i^S + \delta U_j^S||^2_{\mathcal{U}^S} \right.\right.$$
$$\left.\left. + \alpha^2 ||C_i^S \delta H_i + C_j^S \delta H_j||^2_{\mathcal{U}^{S'}} \right) - 2\alpha \sum_{i \in I} \left(\prod_{S \in \mathcal{S}_i} (\delta U_i^S + \delta U_j^S), \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} C_i \delta H_i \right)_{\mathcal{U}^{\mathcal{S}_i}} \right\}^{-1} \tag{3.14}$$

Proof. We have

$$
\begin{aligned}
&J(U+\lambda\delta U) = \sum_{S\in\mathcal{S}} ||C_i^S H_i(U_i) - C_j^S H_j(U_j) + \lambda(C_i^S\delta H_i - C_j^S\delta H_j)||^2_{\mathcal{U}^{S'}} \\
&+\sum_{S\in\mathcal{S}} ||U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)) \\
&\qquad +\lambda(\delta U_i^S + \delta U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S\delta H_i + C_j^S\delta H_j))||^2_{\mathcal{U}^S} \\
&= J(U) + 2\lambda\sum_{i\in I}\sum_{S\in\mathcal{S}_i}\Big(\big(C_i^S H_i(U_i) - C_j^S H_j(U_j), C_i^S\delta H_i\big)_{\mathcal{U}^{S'}} \\
&+\Big(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), \delta U_i^S\Big)_{\mathcal{U}^S} \\
&-\alpha\Big(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j)), \Lambda_{\mathcal{U}^S}^{-1}C_i^S\delta H_i\Big)_{\mathcal{U}^S}\Big) \\
&-2\lambda^2\alpha\sum_{i\in I}\sum_{S\in\mathcal{S}_i}\Big(\delta U_i^S + \delta U_j^S, \Lambda_{\mathcal{U}^S}^{-1}C_i^S\delta H_i\Big)_{\mathcal{U}^S} \\
&+\lambda^2\sum_{S\in\mathcal{S}}\Big(||C_i^S\delta H_i - C_j^S\delta H_j||^2_{\mathcal{U}^{S'}} + ||\delta U_i^S + \delta U_j^S||^2_{\mathcal{U}^S} + \alpha^2||C_i^S\delta H_i + C_j^S\delta H_j||^2_{\mathcal{U}^{S'}}\Big)
\end{aligned}
$$

Moreover,

$$
\begin{aligned}
&J(U+\lambda\delta U) - J(U) + 2\lambda^2\alpha\sum_{i\in I}\Big(\prod_{S\in\mathcal{S}_i}(\delta U_i^S + \delta U_j^S), \Lambda_{\mathcal{U}^S}^{-1}C_i\delta H_i\Big)_{\mathcal{U}^S} \\
&-\lambda^2\sum_{S\in\mathcal{S}}\Big(||C_i^S\delta H_i - C_j^S\delta H_j||^2_{\mathcal{U}^{S'}} + ||\delta U_i^S + \delta U_j^S||^2_{\mathcal{U}^S} + \alpha^2||C_i^S\delta H_i + C_j^S\delta H_j||^2_{\mathcal{U}^{S'}}\Big) \\
&= 2\lambda\sum_{i\in I}\Bigg(\Big(\prod_{S\in\mathcal{S}_i}(C_i^S H_i(U_i) - C_j^S H_j(U_j), C_i\delta H_i\Big)_{\mathcal{U}^{S'}} \\
&+\Big(\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j))), \delta U_i\Big)_{\mathcal{U}^S} \\
&-\alpha\Big(\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j))), \Lambda_{\mathcal{U}^S}^{-1}C_i\delta H_i\Big)_{\mathcal{U}^S}\Bigg) \\
&= 2\lambda\sum_{i\in I}\Bigg(\Big\langle C_i{}^*\Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1}\prod_{S\in\mathcal{S}_i}(C_i^S H_i(U_i) - C_j^S H_j(U_j), \delta H_i\Big\rangle_{V_i',V_i} \\
&-\alpha\Big\langle C_i{}^*\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j))), \delta H_i\Big\rangle_{V_i',V_i} \\
&+\Big\langle\Lambda_{\mathcal{U}^{\mathcal{S}_i}}\prod_{S\in\mathcal{S}_i}(U_i^S + U_j^S - \alpha\Lambda_{\mathcal{U}^S}^{-1}(C_i^S H_i(U_i) + C_j^S H_j(U_j))), \delta U_i\Big\rangle_{\mathcal{U}^{\mathcal{S}_i\prime},\mathcal{U}^{\mathcal{S}_i}}\Bigg)
\end{aligned}
$$

$$
\begin{aligned}
&= 2\lambda \sum_{i\in I} \left\langle A_i^* P_i, A_i^{-1} B_i \delta U_i \right\rangle_{V_i', V_i} \\
&+2\lambda \sum_{i\in I} \left\langle \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S\in\mathcal{S}_i} (U_i^S + U_j^S - \alpha \Lambda_{\mathcal{U}^S}^{-1} (C_i^S H_i(U_i) + C_j^S H_j(U_j))), \delta U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i'}, \mathcal{U}^{\mathcal{S}_i}} \\
&= 2\lambda \sum_{i\in I} \left\langle B_i^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S\in\mathcal{S}_i} (U_i^S + U_j^S - \alpha \Lambda_{\mathcal{U}^S}^{-1} (C_i^S H_i(U_i) + C_j^S H_j(U_j))), \delta U_i \right\rangle_{\mathcal{U}^{\mathcal{S}_i'}, \mathcal{U}^{\mathcal{S}_i}}
\end{aligned}
$$

Then, deriving $\mathcal{J}(\lambda) := J(U + \lambda\delta U)$ with respect to λ and vanishing this derivative, we get (3.14). □

3.4 The Extended Finite Element Method in the DFN context

In this section we briefly describe a discretization approach via extended finite elements for DFN problems that allows us to build the numerical triangulation independently of the traces disposition on each fracture. The solution to Problem (7.1) with matching conditions (7.2)-(7.3) is in general a continuous function with discontinuous gradient along traces. A numerical solution based on standard Finite Elements (FE) would require the triangulation to be conforming to the traces, this in turn requiring a coupling in the meshing process for all the fractures in the system. The Extended Finite Element Method (XFEM) [2, 15, 9, 3], instead, introduces in the FE approximation spaces additional basis functions, called enrichment basis functions, in order to reproduce the irregular behaviour of the solution independently of the mesh. For a detailed description of the XFEM approach we refer the interested reader to the cited references. Let us first consider for simplicity a single trace S on a fixed fracture F. Let V_δ^{fem} be the standard FE trial and test spaces defined on the elements of a triangulation on F non conforming to the trace and spanned by Lagrangian basis functions ϕ_k, for k ranging in an index set $\mathcal{I}$. Let Φ be a function well approximating the irregular behaviour of H in a neighbourhood of the trace S. Starting from Φ and basis functions ϕ_k, using the Partition of Unity Method [1], new basis functions are introduced into the space V_δ^{fem}, enriching its approximation capabilities. The additional basis functions are clearly required only in the elements of the triangulation which are intersected by the trace. In this way the irregular behaviour of the numerical solution is determined by the enrichment functions introduced, and is independent of the position of elements

with respect to the trace. The XFEM space is then:

$$\mathrm{V}_\delta^{xfem} = span\left(\{\phi_k\}_{k\in\mathcal{I}}, \{\phi_k\Phi\}_{k\in\mathcal{J}}\right)$$

where $\mathcal{J}\subset\mathcal{I}$ is the subset of the degrees of freedom involved in the enrichment. Consequently the approximate solution with the XFEM will have the following structure:

$$h_\delta^{xfem} = \sum_{k\in\mathcal{I}} h_k^{xfem}\phi_k + \sum_{k\in\mathcal{J}} a_k^{xfem}\phi_k\Phi,$$

where h_k^{xfem} and a_k^{xfem} are the unknowns related to the standard and enriching basis functions, respectively.

If more traces are present on the fracture F, the approach is simply generalized as follows: the XFEM space is taken as

$$\mathrm{V}_\delta^{xfem} = span\left(\{\phi_k\}_{k\in\mathcal{I}}, \cup_{m\in\mathfrak{M}_F}\{\phi_k\Phi_m\}_{k\in\mathcal{J}_m}\right)$$

where the subset of indices $\mathfrak{M}_F\subset\mathfrak{M}$ corresponds to the traces on F, and Φ_m and $\mathcal{J}_m$ are the enriching function and the set of enriched nodes relative to m-th trace.

We end briefly recalling how enriching functions are selected in the DFN context, referring the reader to [13] for more details in general cases and [7, 6] for details in the DFN simulations. For each fracture F, let S_m, $m\in\mathfrak{M}_F$ be a trace on F. We distinguish two cases: a) S_m is entirely crossing the fracture (the trace is hence a so called closed interface); b) one or more endpoints of S_m lie inside F (open interface). In the case of closed interfaces, the enriching function Φ_m is suitably set as $\Psi^m(\hat{x}) = \|\bar{x}-\hat{x}\|$, where $\bar{x}$ is the projection of $\hat{x}$ on S_m (see [3]).

In the case of open interfaces, Φ_m is still used for reproducing non-smooth behaviour on elements intersecting the trace but not containing trace tips. For each trace tip lying inside F, we also add new enriching functions (see [3]) defined as follows. Let $\sigma_m = \left\{s^1, s^2\right\}$ be the set of trace tips of S_m. If s^ℓ lies inside F, we introduce the enriching functions

$$\Theta_{s^\ell}^m(\mathbf{x}) \in \left\{r\cos\frac{\theta}{2}, r^2\cos\frac{\theta}{2}, \sqrt{r}\cos\frac{\theta}{2}\right\}, \qquad s^\ell\in\sigma_m$$

where r is the distance between the current point and trace tip, and θ is the polar angle of $\hat{x}$ with respect to a reference system centred into trace tip with the x-axis aligned to the trace, and oriented such that the trace lies on the negative side. Tip enrichments are introduced only on elements containing traces endpoints. Functions $\Theta_{s^\ell}^m(\mathbf{x})$ are

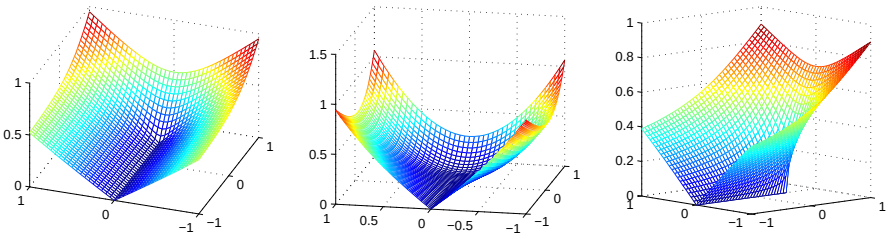


Figure 3.1: Behaviour of trace tip enrichment functions

plotted, from left to right, in Figure 3.1. We remark that the choice of enrichments is quite arbitrary. The selection here adopted is well suited to describe the nonsmooth behaviour of the solution around trace tip. Other choices are possible, as well as the use of a larger number of enrichments around the tip. This latter possibility could improve the description of the solution, but would increase the overall computational cost.

We refer the reader to [13, 7, 6] for more details about implementation of the XFEM, which include for example methods to preserve FEM optimal convergence rates and correctly perform accurate numerical integration of irregular functions.

3.5 Discretization of the constrained optimization problem

Following the paradigm "First optimize then discretize" we now describe the discrete version of the method introduced in the previous section.

Let us introduce an independent triangulation $\mathcal{T}_{\delta,i}$ on each fracture F_i, $\forall i \in \mathfrak{I}$. Let $\mathrm{V}_{\delta,i}$ be the finite dimensional trial and test spaces defined on the elements of $\mathcal{T}_{\delta,i}$ and spanned by Lagrangian basis functions $\phi_{i,k}$, $k \in \mathcal{I}_i = \{1, ..., N_i\}$. Let us denote by h_i the discrete approximation of H_i, $i \in \mathfrak{I}$:

$$h_i(\mathbf{x}) = \sum_{k=1}^{N_i} h_{i,k}\phi_{i,k}(\mathbf{x}), \quad \forall i \in \mathfrak{I}.$$

The algebraic formulation of the operator A_i in equation (7.6) is the usual one:

$$(A_i)_{k\ell} = \int_{F_i} \nabla\phi_{i,k}\nabla\phi_{i,\ell}\, dF_i + \alpha \sum_{s\in\mathcal{S}_i} \int_S \phi_{i,k|_S}\phi_{i,\ell|_S}\, \mathrm{d}\gamma,$$

where, overloading notation, we denote by $A_i \in \mathbb{R}^{N_i\times N_i}$, $i \in \mathfrak{I}$, also the matrix defining the discrete algebraic operator. For all $S \in \mathcal{S}$, let us fix a finite dimensional subspace of $\mathcal{U}^S$ for the discrete approximations u_i^S and u_j^S of the control variables U_i^S and U_j^S. In

the discrete version of the problem we choose $\mathcal{U}^S = \mathrm{L}^2(S)$ and thus we can remove the constraint of disjoint traces made in Section 3.3 (see [7]). Let $\{\psi_k^S\}_{k=1,\dots,N_S}$ be the basis introduced on the discrete control space on trace S. For application of gradient based methods, we choose a common arbitrary basis for u_i^S and u_j^S, $i,j \in I_S$, not necessarily depending neither on the mesh on F_i, nor on the mesh on F_j. So we write

$$u_l^S = \sum_{k=1}^{N_S} u_{l,k}^S \psi_k^S \quad \forall l \in I_S, S \in \mathcal{S}_i\,.$$

For each fracture F_i, we set $N_{\mathcal{S}_i} = \sum_{S\in\mathcal{S}_i} N_S$ as the number of DOFs on traces of F_i. Given an index $i \in \mathfrak{I}$ and a fracture $S \in \mathcal{S}_i$, we define matrices $B_i^S \in \mathbb{R}^{N_i \times N_\mathcal{S}}$ as

$$\left(B_i^S\right)_{k\ell} = \int_S \phi_{i,k|_S} \psi_l^S \,\mathrm{d}\gamma.$$

Matrices B_i^S, $S \in \mathcal{S}_i$, are then grouped row-wise to form the matrix B_i, which acts on a column vector u_i containing all the $N_{\mathcal{S}_i}$ control DOFs corresponding to the traces of F_i, obtained collecting vectors u_i^S, for $S \in \mathcal{S}_i$, with the same ordering introduced for the traces on F_i and used in the definition of B_i. For each fracture F_i let us introduce vectors $h_i \in \mathbb{R}^{N_i}$, $h_i = (h_{i,1},\dots,h_{i,N_i})^T$, and $p_i \in \mathbb{R}^{N_i}$, $p_i = (p_{i,1},\dots,p_{i,N_i})^T$. Furthermore, we define vectors $u \in \mathbb{R}^{N^T}$, with $N^T = \sum_{i\in\mathfrak{I}} N_{\mathcal{S}_i}$, and $h \in \mathbb{R}^{N^F}$, with $N^F = \sum_{i\in\mathfrak{I}} N_i$, as $u = (u_1^T, ...u_{\#\mathfrak{I}}^T)^T$ and $h = (h_1^T, ...h_{\#\mathfrak{I}}^T)^T$. The algebraic formulation of the primal equations (7.6) is then

$$A_i h_i = \tilde{q}_i + B_i u_i, \qquad i \in \mathfrak{I}, \tag{3.15}$$

where $\tilde{q}_i$ accounts for the term q_i in (7.6) and for the weak discrete imposition of boundary conditions on the fracture F_i. We proceed similarly for the equations (7.9), (7.10) and (3.13), in which the operators C_i^S and B_i^*, $i \in \mathfrak{I}$, are nothing but restriction operators. We thus obtain the algebraic equations for the definition of the discrete approximations p_i and δh_i. Further, given two indices $q, r \in \mathfrak{I}$ (possibly $q = r$), we define matrices $C_{q,r}^S$ and $C_{q,r}$ as

$$(C_{q,r}^S)_{k\ell} = \int_S \phi_{q,k|_S} \phi_{r,\ell|_S} \,\mathrm{d}\gamma, \qquad C_{q,r} = \sum_{S\in\mathcal{S}_q} C_{q,r}^S.$$

The discrete counterpart of equations (7.9) and (3.13) $\forall i \in \mathfrak{I}$ are

$$A_i p_i = C_{i,i} h_i - \sum_{j\in J_i} C_{i,j} h_j - \alpha[B_i u_i + \sum_{j\in J_i} B_j u_j - \alpha(C_{i,i} h_i + \sum_{j\in J_i} C_{i,j} h_j)], \tag{3.16}$$

$$A_i \delta h_i = B_i \delta u_i. \tag{3.17}$$

The discrete gradient of the functional $J(U)$ and the optimal step-size λ become

$$\nabla J(u_i) = P_{i|_{\mathcal{S}_i}} + u_i - \alpha h_i(u_i)_{|_{\mathcal{S}_i}} + \sum_{j \in J_i} (u_{j_{|\mathcal{S}_i}} - \alpha h_j(u_j)_{|_{\mathcal{S}_i}}), \tag{3.18}$$

$$\nabla J(u) = \prod_{i \in I} \nabla J(u_i), \tag{3.19}$$

$$\lambda = -\sum_{i \in \mathfrak{I}} (\nabla J(u_i), \delta u_i)_{\mathcal{S}_i} \left\{ -2\alpha \sum_{i \in \mathfrak{I}} (\delta u_i + \delta u_{j_{|\mathcal{S}_i}}, \delta h_{i|_{\mathcal{S}_i}})_{\mathcal{S}_i} \right.$$

$$\left. + \sum_{i \in \mathfrak{I}} \left(||\delta h_{i|_{\mathcal{S}_i}} - \delta h_{j_{|\mathcal{S}_i}}||^2_{\mathcal{S}_i} + ||\delta u_i + \delta u_{j_{|\mathcal{S}_i}}||^2_{\mathcal{S}_i} + \alpha^2 ||\delta h_{i|_{\mathcal{S}_i}} + \delta h_{j_{|\mathcal{S}_i}}||^2_{\mathcal{S}_i} \right) \right\}^{-1}. \tag{3.20}$$

We end this Section with the discrete version of Algorithm 3.1.

Algorithm 3.2.

1. Set $k = 0$ and initial guess for control variable u^0;
2. find $h^0 = h(u^0)$ solving on each fracture (7.19);
3. find $p(u^0)$ solving on each fracture (3.16);
4. evaluate $\nabla J(u^0)$ by (3.19);
5. set $(\delta u)^0 = -\nabla J(u^0)$;
6. *While(stopping criterion not satisfied)*
   - 6.1. compute optimal step-size λ^k along direction $(\delta u)^k$ by (3.20);
   - 6.2. set $u^{k+1} = u^k + \lambda^k (\delta u)^k$;
   - 6.3. $\forall i \in \mathfrak{I}$ find $h_i(u^{k+1})$ by (7.19);
   - 6.4. $\forall i \in \mathfrak{I}$ find $p_i(u^{k+1})$ by (3.16);
   - 6.5. evaluate $\nabla J(u^{k+1})$;
   - 6.6. set $\beta^{k+1} = \|\nabla J(u^{k+1})\|^2_{\mathcal{S}} / \|\nabla J(u^k)\|^2_{\mathcal{S}}$
   - 6.7. set $(\delta u)^{k+1} = -\nabla J(u^{k+1}) + \beta^{k+1} \delta u^k$
   - 6.8. $k = k + 1$;

We notice that, thanks to the minimization approach adopted, only the solution of many simple problems on the fractures is required, with a small exchange of trace-related data among them. This algorithm is therefore suitable for massively parallel computers and GPU-based computers.

3.5.1 Stopping criterion

The stopping criterion used in Algorithm 3.2 plays a relevant role for efficiency reasons. In fact, due to the arbitrary intersections of the traces with elements independently placed on each fracture, the two terms of the functional J do not vanish, in general. This follows from the fact that on each trace S the discrete functions $h_{i|_S}$ and $h_{j|_S}$ with $i, j \in I_S$ are piecewise polynomials on different partitions of the trace. As a consequence, $\delta h_{i|_S} - \delta h_{j|_S}$ is typically different from zero. Appropriate choice for stopping criteria is crucial in order to prevent a premature stop of the algorithm far from the minimum of the functional, avoiding at the same time useless iterations when we are already close to the minimum, when only negligible further reduction of the functional could be achieved at the expenses of a large number of additional iterations. We will discuss this in the next Section.

3.6 Numerical Results

In this section we present some numerical experiments aiming at showing the behaviour of our algorithm in relation to various stopping criteria and the quality of the solution obtained. After introducing the DFN problems used for the simulations, and discussing stopping criteria used in our computations, we analyze the solution obtained on the most complex DFN configuration investigated.

3.6.1 DFN configurations

A set of four different DFN configurations is considered with an increasing number of fractures and traces as described in Table 3.1.

Table 3.1: Problems description

			DOFs (coarse grid)		DOFs (fine grid)	
Label	$\#\mathfrak{I}$	$\#\mathfrak{M}$	h	u	h	u
7fract	7	11	1140	163	4007	378
11fract	11	26	2244	337	7172	825
50fract	50	153	13211	2187	42161	5166
100fract	100	313	26512	4637	85900	10978

In Figure 3.2 we show section on the $x-y$ plane of fracture systems. All fractures extend, orthogonally to $x-y$ plane, from $z=0$ to $z=1$, except for fractures in dashed line that range between $z=0$ and $z=0.5$. Homogeneous or non-homogeneous Dirichelet boundary conditions are prescribed on the sides marked with a dark circle or with a dark rectangle respectively, while homogeneous Neumann conditions are set on the other edges. Problem formulation is as in equation (7.1), where the transmissivity

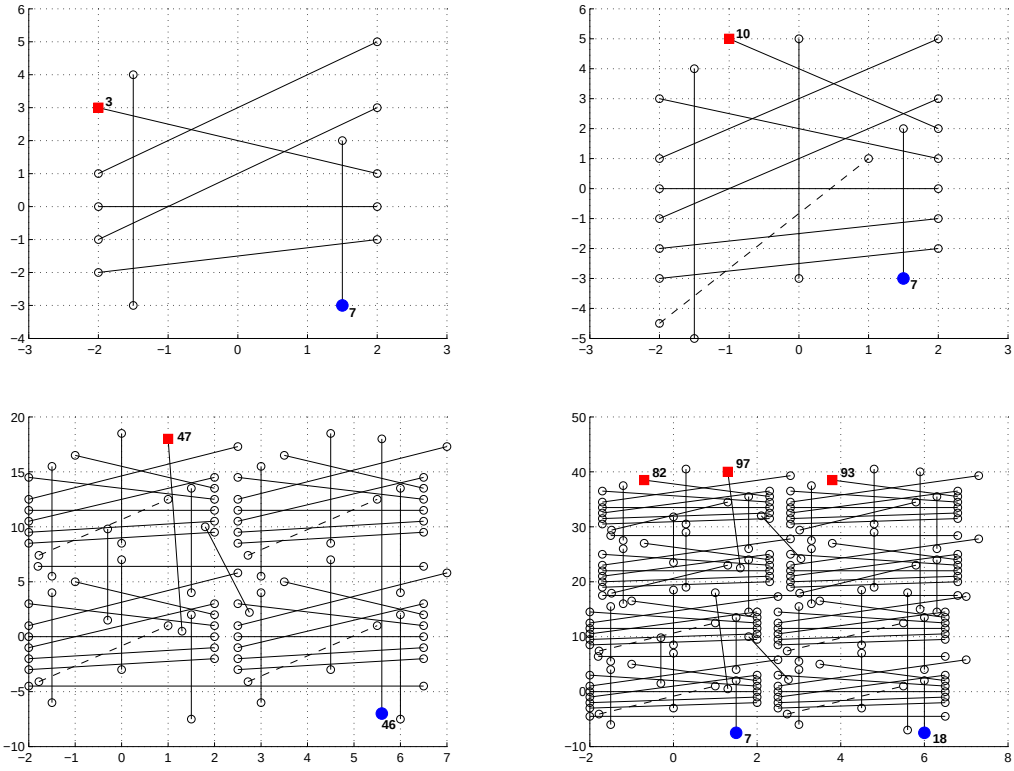


Figure 3.2: DFN configurations, section on $x-y$ plane. Left to right, top to bottom: **7fract**, **11fract**, **50fract**, **100fract**. Number is reported for fractures carrying Dirichelet boundary conditions (squared edge non homogeneous, filled circle homogeneous).

is assumed constant and equal to 1, and the source term is $q=0$ on all the fractures. For the discretization we use first order Lagrangian finite elements and two different grids: a coarse grid with about 35 elements per unit area and a finer grid with about 100 elements per unit area. The corresponding number of DOFs is reported in Table 3.1. In all cases we set the parameter $\alpha=0.5$ in the definition of the control variable and the starting guess for the control variable is $u^0=0$. For each configuration and grid, we

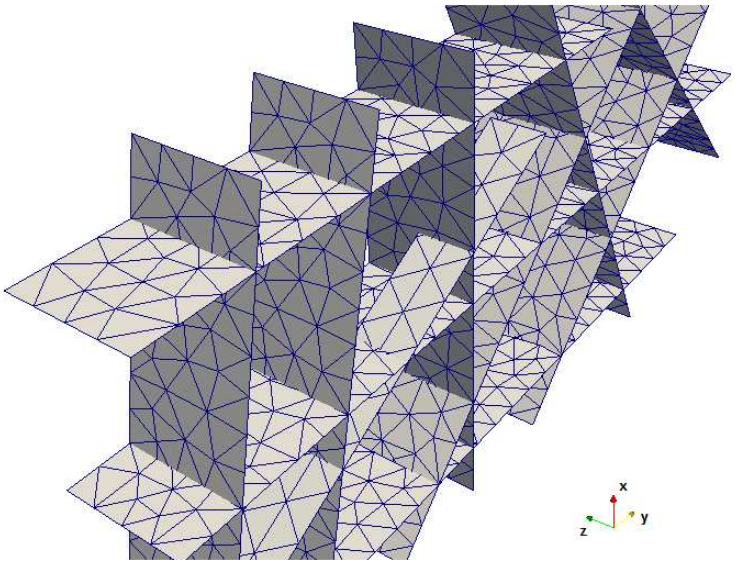


Figure 3.3: Zoom of grid for **11fract** problem.

Table 3.2: Exit criteria used in simulations

Label	Criterion
t_1	$\mathcal{R}_1 = J^k - J^{k-1} < tol_1$
t_2	$\mathcal{R}_2 = \|\|u^k - u^{k-1}\|\| < tol_2$
t_3	$\mathcal{R}_3 = J^k(J^k - J^{k-1}) < tol_3$

define a reference solution obtained performing a large number of gradient iterations in order to safely approach the minimum of the functional. As an example, to highlight the complete non conformity of the mesh to the traces, we show in Figure 3.3 a zoom of the coarse grid for the DFN problem with eleven fractures.

3.6.2 Stopping criteria

For each problem and grid a large set of simulations is performed, considering the different stopping criteria described in the following.

In Figure 3.4 and Figure 3.5 we plot, for the various problems considered and for increasing number of iterations, scaled by the number of problem traces, the distance

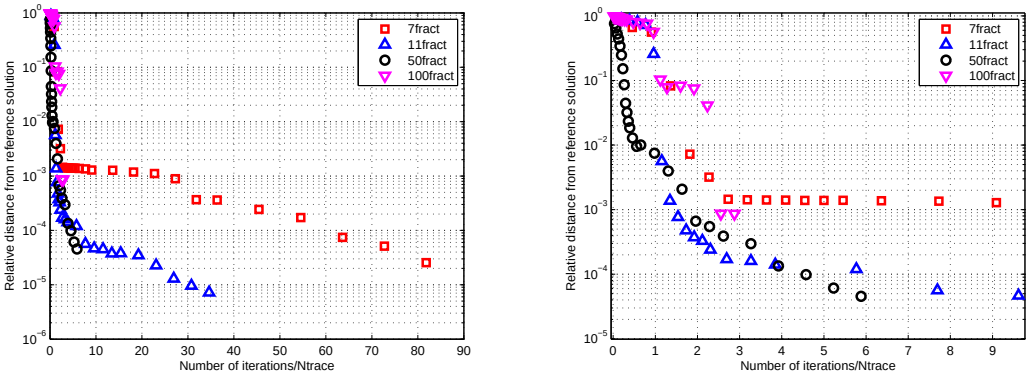


Figure 3.4: Relative distance in H^1-norm of solution at different number of iterations, coarse grid. Right: zoom at lower number of iterations.

in H^1-norm between the reference solution and the current solution, relative to the H^1-norm of the reference solution: $\|h_{curr} - h_{ref}\|_{H^1}/\|h_{ref}\|_{H^1}$. The reference solution is obtained on the same grid, performing a very large number of conjugate gradient iterations. Figure 3.4, on the left, gives an overview on a wide range of iterations for the coarse grid, while on the right provides a zoom at lower iterations. Figure 3.5 provides a similar zoom for the finer grid. It should be noticed that the curves show initially a strong decreasing trend and, after a number of iterations that is few times the number of problem traces, variations of the solution with respect to the reference solution become smaller than 1%. Afterwards, the curves become almost flat and a large number of iterations would be required for negligible improvements in the solution. Therefore, we can see that the algorithm can provide a solution close to the best solution achievable within a reasonably small number of iterations, this number being proportional to the total number of traces in the system, with a proportionality factor in the order of few units.

As mentioned in Subsection 3.5.1, functional minimum is an unknown value different from zero. Hence, the choice of a exit criterion able to stop iterations when we are close enough to the solution, while avoiding useless iterations, is a crucial point. In Table 3.2 we report three possible criteria. Condition t_1 detects small variations in the functional values. Since the functional descent path can be step-like (see Figure 7.4 for an example), in order to avoid premature stops, the algorithm is terminated when $\mathcal{R}_1 < tol_1$ for a fixed number of subsequent iterations (six, in our computations). Approaching functional minimum we have that $\mathcal{R}_1 \to 0$. In Figure 3.7, left, we show the relative distance of

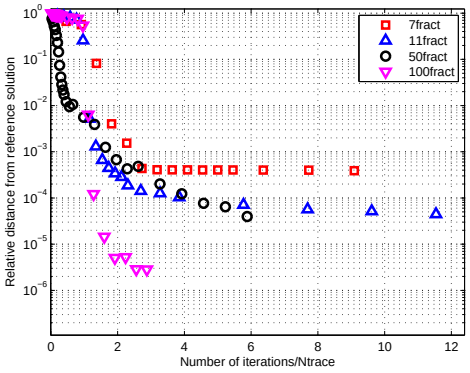


Figure 3.5: Relative distance in H^1-norm of solution at different number of iterations, finer grid. Zooming al lower number of iterations.

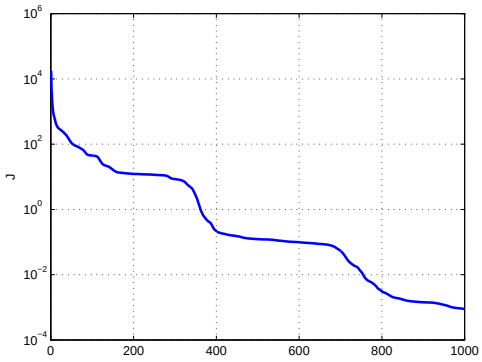


Figure 3.6: Example of functional step-like descent path. Problem **100fract** on the coarse grid.

the computed solution from the reference solution, corresponding to several values of the tolerance tol_1. It can be noticed that a value around $tol_1 = 10^{-6}$ provides a good solution for all the problems considered.

Similarly, condition t_2 seeks small variations in the control variable. Again, to take into account possible temporary stagnation during the descent process, iterations are stopped when $\mathcal{R}_2 < tol_2$ six times subsequently. Also in this case as the functional approaches its minimum $\mathcal{R}_2$ tends to zero. We can see in Figure 3.7, middle, the behaviour of the solution in relation to the choice of tol_2. The value $tol_2 = 10^{-7}$ appears to be a good choice.

As a possible alternative, criterion t_3 aims at detecting functional minimum, again avoiding premature stop at large values of the functional due to local stagnation. The rationale behind this criterion is to avoid stopping the iterates when $J^k - J^{k-1}$ is small but J^k is not small as well. Algorithm is then stopped the first time that $\mathcal{R}_3 < tol_3$. Also in this case $\mathcal{R}_3$ can be arbitrarily reduced with iterations. We plot solution behaviour in relation to tol_3 in Figure 3.7, right. We notice that in this case low tolerance values, around $tol_3 = 10^{-8}$, should be chosen.

3.6.3 DFN system solution

We now show the quality of the numerical solution obtained on the more complex DFN configuration considered herein. First we show in details the results obtained on

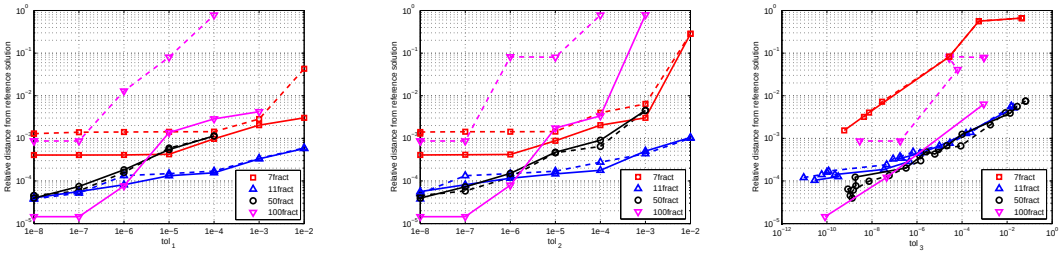


Figure 3.7: Relative distance in H^1-norm from reference solution for different tolerances and stopping criteria. Left: condition t_1; middle: condition t_2; right: condition t_3. Coarse grid in dashed line, finer grid in solid line.

two of the fractures in the **100fract** system: the source fracture 82 and the sink fracture 18 (see Figure 3.2). On the coarse grid, in Figures 3.8 and 3.9, left, we compare the solution on fracture traces, $\left\{h_{i_{|S}}\right\}_{S\in\mathcal{S}_i}$, $i=\{18,82\}$, and the solution on the traces of intersecting fractures, $\{h_j\}$ with $j\in J_i$. We can see a very good agreement between these values, ensuring the global continuity of the hydraulic head. In the right part of the same figures, we compare the co-normal derivative of solution on the traces of the current fracture and on trace-twin fracture (with opposite sign). In these figures $\phi(h)=\left[\!\left[\frac{\partial h}{\partial \tilde{\nu}_S}\right]\!\right]_S$. Again, we can observe, as expected, a very good agreement between these values, ensuring flux conservation.

In Figure 3.10 we show, for the same fractures, the solution on the traces obtained with four different meshes. Reported results show that, under grid refinement, the computed solutions are clearly approaching the same values. In Figure 3.11 we plot the whole solution obtained with the coarse grid on the fractures 82 and 18. In Figures 3.12 and 3.13 we report 3D pictures representing the DFN. The computing meshes are drawn and the solution is reported on the fractures using a colorbar. The arrows point the source fracture 93 and the sink fracture 7.

In Figure 3.14, left, the L^2-norm of solution against iterations is plotted. The table of Figure 3.14, right, gives an indication of the conservativity of the method on the whole network of fractures, as it reports the values of the total fluxes discharged by the source fractures to the system and the total flux received by the sink fractures from the system. As expected the data match very well.

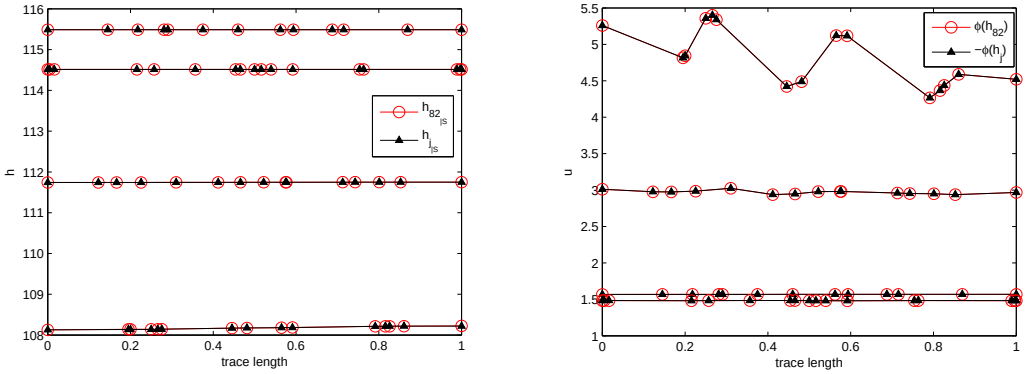


Figure 3.8: Problem **100fract**, source fracture 82, coarse grid. Solution on the traces (left) and co-normal derivative (right) compared with corresponding values computed on trace-twin fractures.

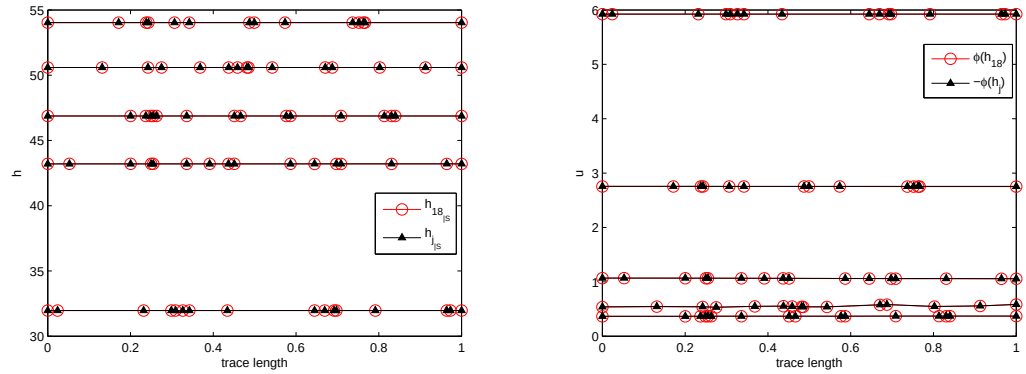


Figure 3.9: Problem **100fract**, sink fracture 18, coarse grid. Solution on the traces (left) and co-normal derivative (right) compared with corresponding values computed on trace-twin fractures.

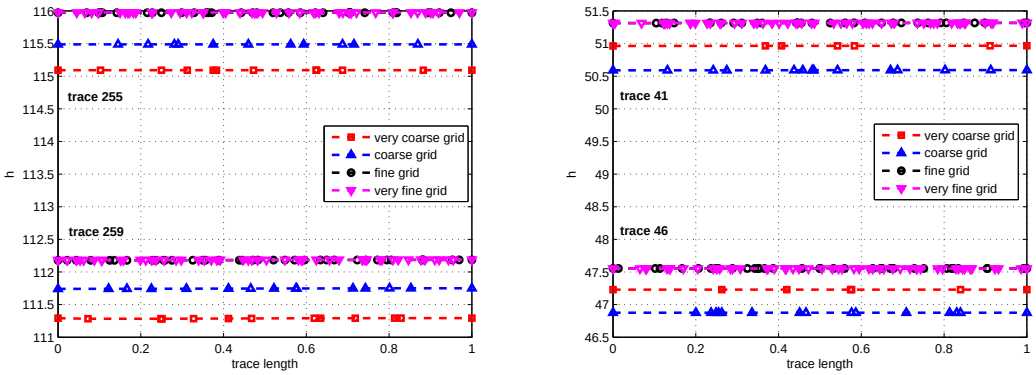


Figure 3.10: Solution on the traces of source fracture 82 (left) and sink fracture 18 (right) for two different grids, **100fract** problem.

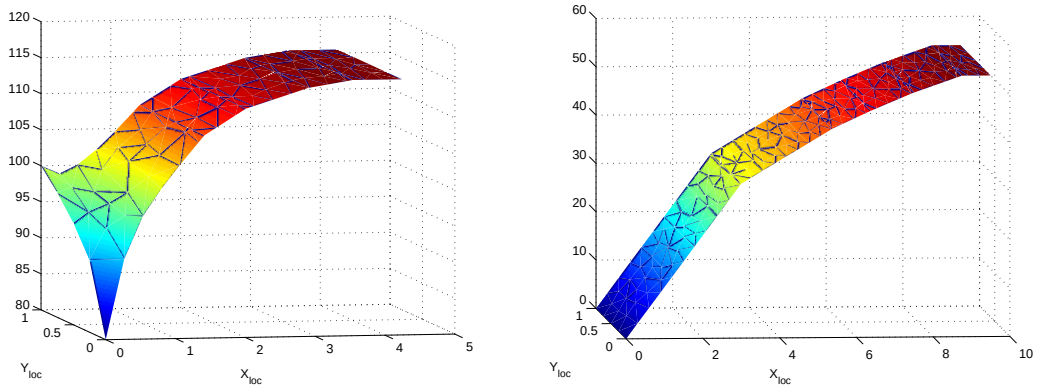


Figure 3.11: Solution on source fracture 82 (left) and on sink fracture 18 (right) for **100fract** problem. Reference system is local. Coarse grid.

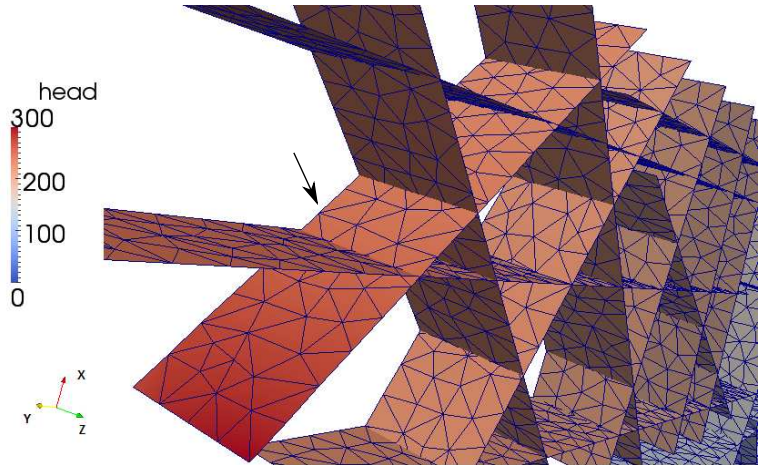


Figure 3.12: Solution on the DFN **100fract**. Arrow points source fracture 93.

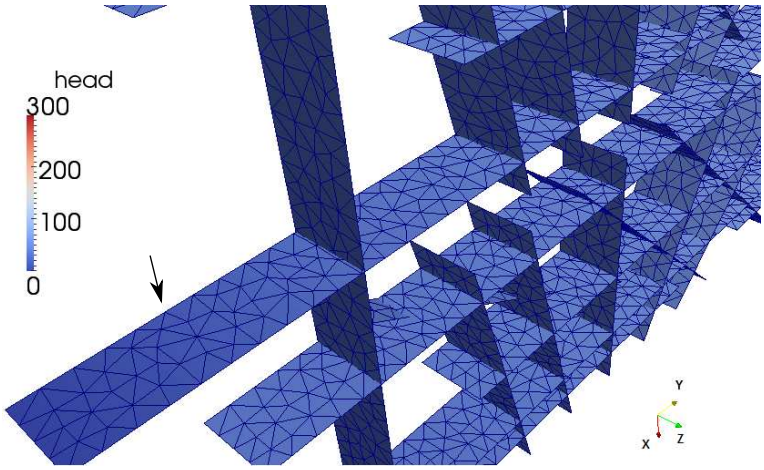
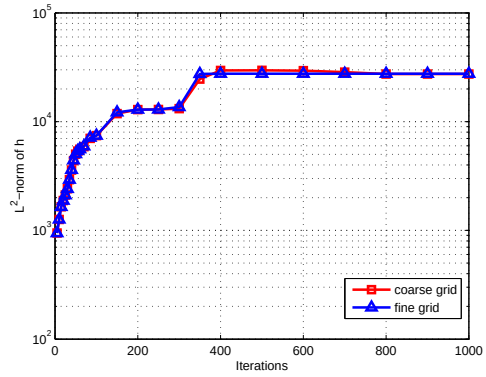


Figure 3.13: Solution on the DFN **100fract**. Arrow points sink fracture 7.



Grid	Source	Sink
very coarse grid	−15.12350	15.11294
coarse grid	−15.34119	14.99621
fine grid	−15.21692	15.21691
very fine grid	−15.21841	15.21653

Figure 3.14: Problem **100fract**: L^2-norm of solution against iterations (left) and total fluxes on the source and sink fractures (right).

3.7 Conclusions

Major drawbacks in DFN numerical simulations lie in the definition of a good quality finite element triangulation and in the huge computational demand. The method introduced in [7, 6] and further developed in the present work provides a possible approach for circumventing these difficulties. The proposed method allows a fully independent triangulation on each fracture, thus eliminating any mesh related problem. Further, the method can be easily implemented on parallel machines, since the DFN simulation is split in many sub-problems on each fracture that can be solved independently by parallel processes, with little exchange of trace related data between trace-twin processes.

The contribution of the present work to the method is twofold. We introduce a new definition of the control variable for the optimal problem in order to eliminate the requirement of having a non-empty portion of the boundary of each fracture with Dirichelet boundary condition. We also introduce a conjugate gradient method for the optimization process in order to speed up convergence. By means of several numerical results we show that our algorithm provides a good quality solution within a small number of iterations that increases linearly with the number of traces in the system. The proportionality factor is in the order of few units for all the problems considered. The main computational effort in each iteration is devoted to the resolution of the linear systems on the fractures, that however are independent each other. Assuming that these linear systems have a comparable dimension, the total cost of each iteration scales as the number of fractures. Effectiveness of some stopping criteria for the gradient iterations

is also discussed.

Further developments on the topic should include on one side an investigation of the scalability of the algorithm using an increasing number of parallel processes on different parallel architectures, and on another side the analysis of non-stationary problems. In the non-steady case, in order to reduce the computational effort, the application of reliable and efficient space-time a posteriori error estimates and adaptive algorithms, following the approaches of [4, 5], could be fruitful.

Acknowledgments

This work was financed by Italian funds from Miur (PRIN grant) and INdAM-GNCS funds.

Bibliography

[1] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[2] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, Internat. J. Numer. Methods Engrg., 45 (1999), pp. 601–620.

[3] T. Belytschko, N. Möes, S. Usui, and C. Parimi, *Arbitrary discontinuities in finite elements*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 993–1013.

[4] S. Berrone, *Robust a posteriori error estimates for finite element discretization of the heat equation with discontinuous coefficients*, M2AN, 40 (2006), pp. 991–1021.

[5] ——, *A local-in-space-timestep approach to a finite element discretization of the heat equation with a posteriori estimates*, SIAM J. Numer. Anal., 47 (2009), pp. 3109–3138.

[6] S. Berrone, S. Pieraccini, and S. Scialó, *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM Journal on Scientific Computing, (2013). To appear. Preprint available at http://porto.polito.it/id/eprint/2499005.

[7] ——, *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM Journal on Scientific Computing, (2013). To appear. Preprint available at http://porto.polito.it/id/eprint/2495599.

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

[9] C. Daux, N. Möes, J. Dolbow, N. Sukumar, and T. Belytschko, *Arbitrary branched and intersecting cracks with the extended finite element method*, Internat. J. Numer. Methods Engrg., 48 (2000), pp. 1741–1760.

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

[11] W. S. Dershowitz and C. Fidelibus, *Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method*, Water Resource Res., 35 (1999), pp. 2685–2691.

[12] J. Erhel, J.-R. de Dreuzy, and B. Poirriez, *Flow simulation in three-dimensional discrete fracture networks*, SIAM J. Sci. Comput., 31 (2009), pp. 2688–2705.

[13] T.-P. Fries and T. Belytschko, *The extended/generalized finite element method: an overview of the method and its applications*, Internat. J. Numer. Methods Engrg., 84 (2010), pp. 253–304.

[14] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[15] N. Möes, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing*, Internat. J. Numer. Methods Engrg., 46 (1999), pp. 131–150.

[16] J. Nocedal and S. Wright, *Numerical Optimization*, Springer, 1999.

[17] B. Nœtinger and N. Jarrige, *A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks*, J. Comput. Phys., 231 (2012), pp. 23–38.

[18] A. W. Nordqvist, Y. W. Tsang, C. F. Tsang, B. Dverstop, and J. Andersson, *A variable aperture fracture network model for flow and transport in fractured rocks*, Water Resource Res., 28 (1992), pp. 1703–1713.

[19] G. Pichot, J. Erhel, and J.-R. De Dreuzy, *A mixed hybrid Mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629–1643.

[20] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[21] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.

Chapter 4

The eXtended Finite Element Method for subsurface flow simulations

Abstract In this paper the application of the extended finite element method (XFEM) to a novel approach to Discrete Fracture Network (DFN) simulations is fully described. The proposed DFN simulation approach does not require any conformity of the triangulation at fracture intersections, thus overcoming one of the major limitations in DFN simulations. Furthermore the initial problem complexity is split in a large number of quasi-independent simple problems on the fractures that can be easily handled by parallel computers. The use of the XFEM allows a good-quality reproduction of the solution also at fracture intersections, despite the non conformity of the mesh. The issue of enrichment function selection is addressed, and suitable simple enrichment functions are identified in order to keep computational cost as low as possible without compromising representation capabilities of the enriched space. All the relevant aspects of XFEM implementation are thoroughly discussed and numerical examples reproducing critical configuration are provided and commented. Simulations on complex DFN configurations are also reported in order to show the effectiveness of the method.

4.1 Introduction

Efficient simulation and investigation of subsurface flow is an up-to-date open research topic: the complexity of the problem and the increasing interest of many applications, such as analysis of pollutant diffusion in aquifers, oil/gas extraction, gas storage, make this research issue of great interest. In these applications, the computational domain for the simulations consists of underground geological reservoirs, that usually have huge complex heterogeneous structure and for which only stochastic data are typically available. Among others, Discrete Fracture Network (DFN) models are widely used for the simulation. A DFN model describes a geological reservoir as a system of intersecting planar polygons representing the network of fractures in the underground. Fracture intersections are called *traces*. In the present work we consider impervious surrounding rock matrix, so that no flux exchange occurs with the surrounding medium. The quantity of interest is the flow potential, called hydraulic head, given by the sum of pressure and elevation. The hydraulic head is ruled by Darcy law in each fracture, with additional matching conditions which ensure hydraulic head continuity and flux balance at fracture intersections. Thanks to these matching conditions, hydraulic head is continuous across traces but jumps of gradients may occur as a consequence of flux exchange between intersecting fractures. Hence, traces are typically interfaces of discontinuities for the gradient of the solution.

Standard finite element methods or mixed finite elements are widely used for obtaining a numerical solution also in this context, but they require mesh elements to conform with the traces in order to correctly describe the irregular behaviour of the solution. This poses a severe limitation, since realistic fracture networks are typically very intricate, with fractures intersecting each other with arbitrary orientation, position, density and dimension. A conforming meshing process may result infeasible, or might generate a poor quality mesh, since a coupled meshing process on all the fractures of the system may lead to elongated elements. The following strategies are mainly suggested in the literature in order to overcome these difficulties. In some cases mesh and/or geometry modifications and simplifications are proposed to ease meshing process, as for example in [12, 8, 17]. Another approach consists in developing methods which allow for a so called *partial* nonconformity. For example in [14, 15] mortar methods are used in order to relax mesh conformity constraints on intersecting fractures, but still requiring that element edges lie on the traces. A different strategy is used in [4, 5, 6], in which the authors pro-

pose a PDE-constrained optimization approach in which neither fracture/fracture nor fracture/trace mesh conformity is required. The method is based on the minimization of a quadratic functional constrained by the state equations describing the flow on the fractures. Extended Finite Elements (XFEM) are used in order to enrich the solution description and correctly reproduce irregularities in the solution.

The XFEM [2, 13, 7, 16, 3] allows the description of irregular solutions regardless of the position of mesh elements with respect to the irregularity interfaces, so that the numerical triangulation for DFN simulations can be generated independently on each fracture, without any kind of matching constraint along the traces, thus circumventing any problem related to mesh generation. As proved by the numerical results, the behaviour of the solution is well reproduced thanks to special enrichment functions that influence the numerical approximation locally around the traces. Simple, easily integrable enrichment functions should be preferred, in order to limit the number of the related additional unknowns and the computational cost in general.

In the present work we discuss in full details the application of XFEM to the approach described in [4, 5, 6]. Suitable enrichment functions for very complex DFNs are proposed. Furthermore, other issues ensuring an effective implementation of the method are fully addressed.

The present work is organized as follows: in Section 7.2 the PDE-constrained optimization model for DFN flow simulations is briefly recalled. In Section 4.3 a thorough description of the XFEM in the DFN context is provided, as well as implementation choices. In Section 4.4 the numerical solver is depicted. Section 4.5 is devoted to numerical experiments on test problems and DFNs of increasing complexity, which highlight the effectiveness of the XFEM in this context. We end with some conclusions in Section 4.6.

4.2 Problem description

Let us consider a DFN Ω given by the union of open planar polygonal sets F_i, with $i = 1, \ldots, I$, called fractures, and 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 Dirichelet boundary and Γ_N the Neumann boundary. Similarly, the boundary of each fracture is divided in a Dirichelet 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 $\Gamma_{iD} \cap \Gamma_{iN} = \emptyset$. Note that $\Gamma_{iD} = \emptyset$ is allowed whenever $\partial F_i \cap \Gamma_D = \emptyset$. Fractures have arbitrary orientations in

space, so that Ω is a 3D domain. Traces are denoted by S_m, $m = 1, \ldots, M$; $\mathcal{S}$ denotes the set of all the traces of the system, and $\mathcal{S}_i$, for $i = 1, \ldots, I$, denotes the subset of $\mathcal{S}$ corresponding to traces belonging to F_i. Each S_m uniquely identifies a couple of indices $I_{S_m} = \{i, j\}$, such that $S_m \subseteq \bar{F}_i \cap \bar{F}_j$.

According to Darcy law, the hydraulic head H in Ω is given by a system of equations on each fracture, defined as follows. For the sake of simplicity of notation, in this section let us assume that traces are non-intersecting. We remark that the numerical method described in the following is not affected by this assumption. Let H_i denote the restriction of the solution H to fracture F_i and let $\mathbf{K}_i$ be a symmetric and uniformly positive definite tensor (the fracture transmissivity). 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

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

Then H_i satisfies, for $i = 1, \ldots, I$, the following problem: find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\begin{aligned} \int_{F_i} \mathbf{K}_i \nabla H_i \nabla v d\Omega \;\; &= \;\; \int_{F_i} q_i v d\Omega + \langle G_i^N, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} \\ &\quad + \sum_{S \in \mathcal{S}_i} \langle \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(S), \mathrm{H}^{\frac{1}{2}}(S)}, \end{aligned} \tag{4.1}$$

where q_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. 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 Dirichelet and Neumann boundary conditions respectively on the boundary ∂F_i.

Equations (4.1) for $i = 1, ..., I$ are coupled with the following additional matching

conditions imposing hydraulic head continuity and flux balance across the traces:

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i,j \in I_{S_m},\ \forall m = 1,...M, \tag{4.2}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_{S_m}}\right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}^j_{S_m}}\right]\!\right]_{S_m} = 0, \qquad \text{for } i,j \in I_{S_m}. \tag{4.3}$$

Following the method described in [4, 5, 6], instead of solving the coupled system of equations (4.1)-(4.3), the solution is obtained solving a PDE constrained optimization problem.

For each trace in each fracture let us introduce the control variables $U_i^S \in \mathcal{U}^S = \mathrm{H}^{-\frac{1}{2}}(S)$, defined as $U_i^S = \alpha H_{i|_S} + \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}^i_S}\right]\!\right]_S$, where α is a fixed positive parameter. Equation (4.1), prescribed on the fractures, can be equivalently restated as:

$$\int_{F_i} \mathbf{K}_i \nabla H_i \nabla v d\Omega + \alpha \sum_{S\in\mathcal{S}_i} \int_S H_{i|_S} v_{|_S} d\Gamma = \tag{4.4}$$

$$\int_{F_i} q_i v d\Omega + \langle G_i^N, v_{|_S}\rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}),\mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} + \sum_{S\in\mathcal{S}_i} \langle U_i^S, v_{|_S}\rangle_{\mathcal{U}^S,\mathcal{U}^{S\prime}}.$$

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

$$\begin{aligned}
A_i &\in \mathcal{L}(V_i, V_i'), \quad \langle A_i H_i^0, v\rangle_{V_i',V_i} = (\mathbf{K}_i \nabla H_i^0, \nabla v) + \alpha \left(H_i^0{}_{|_{\mathcal{S}_i}}, v_{|_{\mathcal{S}_i}}\right)_{\mathcal{S}_i},\\
A_i^D &\in \mathcal{L}(V_i^D, V_i'), \langle A_i^D \mathcal{R}_i H_i^D, v\rangle_{V_i',V_i} = (\mathbf{K}_i \nabla \mathcal{R}_i H_i^D, \nabla v) + \alpha \left((\mathcal{R}_i H_i^D)_{|_{\mathcal{S}_i}}, v_{|_{\mathcal{S}_i}}\right)_{\mathcal{S}_i},\\
B_i^S &\in \mathcal{L}(\mathcal{U}^S, V_i'), \quad \langle B_i^S U_i^S, v\rangle_{V_i',V_i} = \langle U_i^S, v_{|_S}\rangle_{\mathcal{U}^S,\mathcal{U}^{S\prime}},\\
B_i &= \prod_{S\in\mathcal{S}_i} B_i^S \in \mathcal{L}(\mathcal{U}^{\mathcal{S}_i}, V_i'), \quad \langle B_i U_i, v\rangle_{V_i',V_i} = \langle U_i, v_{|_{\mathcal{S}_i}}\rangle_{\mathcal{U}^{\mathcal{S}_i},\mathcal{U}^{\mathcal{S}_i\prime}},
\end{aligned}$$

with $H_i^0 \in V_i$, $H_i^D \in V_i^D$, $v \in V_i$, 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 $A_i^* \in \mathcal{L}(V_i, V_i')$. The operator $B_{iN} \in \mathcal{L}(\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), V_i')$ imposing Neumann boundary conditions is defined such that

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

With these definitions at hand, problems (4.1) are rewritten 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

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_i^N - A_i^D \mathcal{R}_i H_i^D, \quad \text{in } F_i. \tag{4.5}$$

We remark that, if $\alpha > 0$, for a given U_i, the solution H_i to (4.5) exists and is unique for a non isolated fracture even if we set Neumann boundary conditions on the whole ∂F_i.

Now let us introduce the functional

$$\begin{aligned} J(H,U) &= \sum_{m=1}^{M} \left\| H_{i|_{S_m}} - H_{j|_{S_m}} \right\|^2_{\mathrm{H}^{\frac{1}{2}}(S)} \\ &\quad + \sum_{m=1}^{M} \left\| U_i^{S_m} + U_j^{S_m} - \alpha \left(H_{i|_{S_m}} + H_{j|_{S_m}} \right) \right\|^2_{\mathrm{H}^{-\frac{1}{2}}(S)} . \end{aligned} \tag{4.6}$$

The functional J is quadratic and using the same arguments as in [4], it can be shown that its unique minimum is obtained for values of H and of the control functions U that correspond to the fulfilment of conditions (4.2) and (4.3) on the traces. In other words, the solution of the problem

$$\min J \quad \text{subject to (4.5)} \tag{4.7}$$

corresponds to the solution of the coupled system of equations (4.1)-(4.3).

4.3 The XFEM for DFN simulations

According to the approach depicted in the previous section, matching conditions along traces are not exactly imposed but they are made as small as possible via an optimization approach. Only *local* problems on fractures (i.e. problems (4.5)) are independently solved. As a consequence, meshes on the fractures are neither required to conform to each other, nor to conform to the traces. Clearly, the finer the grid, the smaller is the global mismatch provided by J. In order to provide a better description of the solution also near traces, which represent possible nonsmoothness interfaces, the XFEM turns out to be a convenient approach.

The XFEM can reproduce irregular solutions by means of custom enrichment functions that are added to the trial and test functional spaces of standard finite elements, in order to give the required behaviour to the numerical approximation, independently of the position of mesh elements with respect to the interfaces. A key point of our approach is that we *a priori* know that the solution displays derivative discontinuities at the traces: the solution is in general a continuous function with discontinuous normal derivatives across the traces due to the term representing flux jump. Standard finite

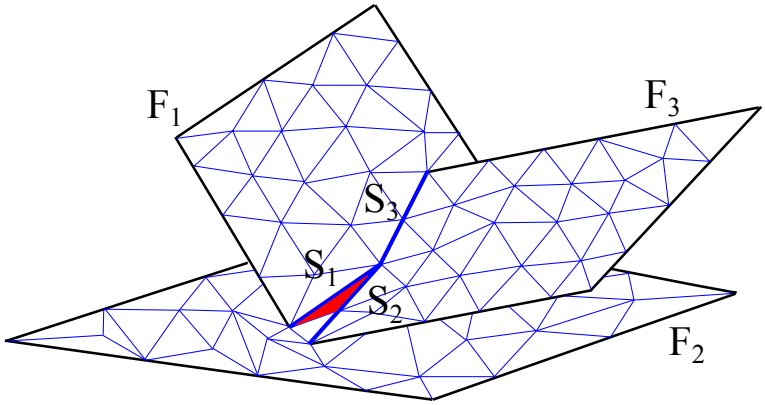


Figure 4.1: Example of a conforming mesh with three traces intersecting with a small angle. $S_1 = \bar{F}_1 \cap \bar{F}_2$, $S_2 = \bar{F}_2 \cap \bar{F}_3$, $S_3 = \bar{F}_3 \cap \bar{F}_1$.

element methods reproduce this behaviour only if mesh element edges lie on the traces, thus requiring the simultaneous conforming triangulation of all the fractures in the system. As discussed, this process often results infeasible for DFNs of realistic size and geometry, or might lead to meshes of poor quality due to the presence of elongated elements trapped between intersecting traces. This situation is described in Figure 4.1 for a simple DFN composed by three fractures and three intersecting traces with a conforming mesh. Due to the reciprocal position of traces, the coloured element could display a very small angle. This problem can be overcome by the use of XFEM; an example of non-conforming mesh suitable for our approach is displayed in Figure 4.2.

In the following of this section, we fully account for details concerning use of XFEM, such as selection of enrichment functions for DFN problems and implementation strategies adopted for this specific application of the XFEM. Before proceeding, we briefly recall some key points concerning XFEM in the context of DFN simulations.

Let us consider a standard finite element description of the hydraulic head in a given fracture $F \subset \mathbb{R}^2$, with a local reference system $\hat{x}$, and M_F traces S_m, $m = 1, \ldots, M_F$. Here and in the sequel of the paper, we use lowercase letters h, u for finite element approximations of the corresponding quantities H and U. Let us introduce a triangulation $\mathcal{T}_\delta$ of F, with N^{el} elements $\tau_e \subset \mathbb{R}^2$ such that $\bar{F} = \bigcup_{1 \le e \le N^{el}} \tau_e$. Let V_δ^{fem} be the standard finite element trial and test space defined on the elements of $\mathcal{T}_\delta$ and spanned by Lagrangian basis functions ϕ_k compactly supported with support Δ_k, with $k \in \mathcal{I}$ set of degrees of freedom (DOF). We remark that discontinuities of the gradient of the solution h occur at traces, which are always segments. If elements of $\mathcal{T}_\delta$ are

conforming to the traces, the solution is given by

$$h^{fem}(\hat{x}) = \sum_{k \in \mathcal{I}} h_k \phi_k(\hat{x}),$$

where h_k is the degree of freedom corresponding to ϕ_k. In the more general case in which we allow elements to be non-conforming to traces, we use our a-priori knowledge on the irregularity of the solution, and use the XFEM on the non-conforming grid, introducing, for each trace S_m, a *global enrichment function* Φ_m that well matches the behaviour of the solution across the trace (see for example Figure 4.3). Additional basis functions, called *local enrichment functions* are generated from functions Φ_m by means of the Partition of Unity Method [1] on the partition of unity given by the standard Lagrange basis functions ϕ_k on the triangulation $\mathcal{T}_\delta$. The numerical approximation given by the XFEM is built on the enriched functional space V_δ^{xfem}

$$V_\delta^{xfem} = span\left(\{\phi_k\}_{k\in\mathcal{I}}, \{\phi_k\Phi_1\}_{k\in\mathcal{J}_1}, \ldots, \{\phi_k\Phi_{M_F}\}_{k\in\mathcal{J}_{M_F}}\right),$$

and has the following structure:

$$h^{xfem}(\hat{x}) = \sum_{k\in\mathcal{I}} h_k^{xfem}\phi_k(\hat{x}) + \sum_{m=1}^{M_F}\sum_{k\in\mathcal{J}_m} \hat{h}_{k,m}^{xfem}\phi_k(\hat{x})\Phi_m(\hat{x}), \tag{4.8}$$

where h_k^{xfem} are the unknowns related to standard finite element basis functions and $\hat{h}_{k,m}^{xfem}$ are the DOFs of the enrichment basis functions related to the m-th trace. The set $\mathcal{J}_m \subset \mathcal{I}$ collects the active DOFs for the m-th enrichment (called *enriched DOFs*). By properly choosing $\mathcal{J}_m$, we can control locality of the enrichments. Indeed, each local enrichment function $\phi_k\Phi_m$ has compact support equal to the support of ϕ_k, Δ_k, and, as a consequence, the region Δ_m of the domain subject to the enrichment Φ_m is determined by the set of active standard FE DOFs: $\Delta_m = \left\{\bigcup_{k\in\mathcal{J}_m}\Delta_k\right\}$.

In the remaining of this Section, we focus on three major issues concerning the use of XFEM in the context of DFN simulation: (i) enrichment function choice, (ii) preservation of optimal convergence rates, and (iii) ill conditioning prevention. To simplify the notation, also in the sequel our discussion refers to a single fracture plane F with M_F traces. We remark that all considerations are independent of the number of fractures in the DFN, being the discretization of the governing equations on each fracture independent from the others. Our discussion is also independent of possible traces intersections, as thanks to additivity property highlighted by (4.8), no special enrichment is considered for traces intersection (see the next Subsection 4.3.1).

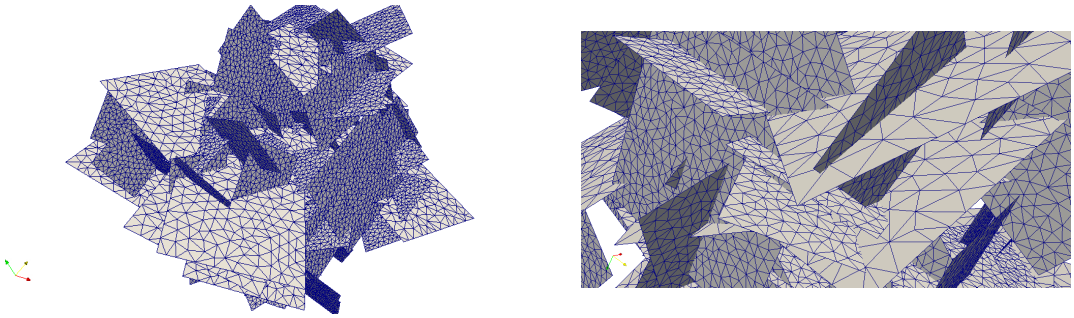


Figure 4.2: Example of DFN with a nonconforming mesh (left). Zoom of mesh detail on the right.

4.3.1 Selection of enrichment functions

Enrichment function selection is a key issue for XFEM implementation, and it is ruled by the kind of irregular behaviour to be reproduced and by the nature of the interfaces (see [10] for a comprehensive review). In the context of DFN models concerning the first point we have to face continuous solutions with discontinuous derivatives; going to the second point, irregularity interfaces are usually classified as closed or open interfaces: closed interfaces extend throughout the whole computational domain, whereas open interfaces end and/or begin inside the domain. In DFN models traces can be arbitrarily placed inside the fractures, thus originating open and closed interfaces which might have multiple intersections among each other.
In addition to this geometrical complexity, since the number of interfaces may be large, a high number of enrichment unknowns could be required. As a consequence, in order to mitigate complexity of the enriched space, we choose a rather simple enriching function, given by the distance function:

$$\Phi_m(\hat{x}) = d(\hat{x}, S_m) \qquad \forall m = 1, \dots, M_F, \tag{4.9}$$

where, following standard notation, $d(x, S)$ denotes the distance of point x from the set S. Setting $\mathcal{J}_m = \{k \in \mathcal{I} : \Delta_k \cap S_m \neq \emptyset\}$, the influence of each global enrichment is limited to the elements with a non-empty intersection with the trace. This choice can strongly reduce the number of DOFs if compared with [5, 6], where more enrichment functions are used for the tips of the traces. The typical behaviour of functions (4.9) is shown in Figure 4.3, and they are used for both open and closed interfaces, thus keeping as low as possible the number of required enrichments (and consequently the number

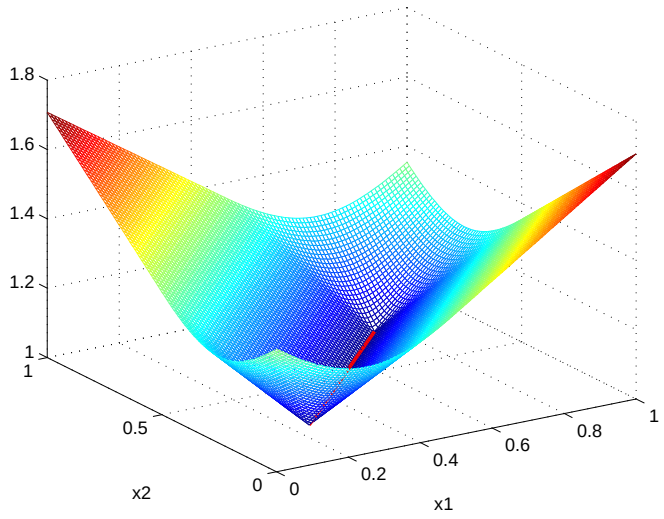


Figure 4.3: Example of global enrichment function Φ_m.

of unknowns), but still giving a good approximation of the behaviour of the solution around traces, as shown in the numerical examples of Section 4.5.

The gradient of the enrichment functions has a discontinuous component normal to the trace, and therefore special care is needed for the numerical integration. To this aim, mesh elements crossed by traces are divided in sub-elements, in such a way that only sub-elements edges or vertices lie on the traces (see for example Figure 4.6, right). Low order Gaussian formulae are then used on the sub-elements without loss of accuracy, thanks to the simple structure of the enrichment functions, and with a moderate number of function evaluations. This point is of paramount importance in order to limit the computational cost when a large number of traces is considered.

We remark that no specific enrichment functions are required in the case of intersecting traces, since the enrichments enjoy an additivity property, as emphasized by the structure of (4.8). The linear combination of the enrichments (4.3) introduced for each intersecting trace is sufficient to approximate the irregular behaviour of the solution. Figure 4.4 shows a linear combination of functions (4.3) for two intersecting traces with triangular first order finite elements. This simple example shows that it is possible to reproduce a solution which is continuous across the traces (Figure 4.4) but with a different value of the normal component of the gradient in each of the four regions separated by the traces.

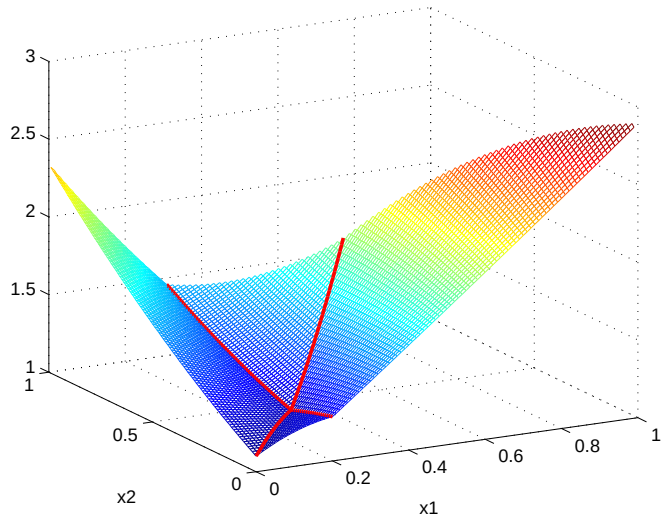


Figure 4.4: Example of approximated solution on the reference triangle for two crossing traces and first order basis functions.

4.3.2 Convergence rates

As previously discussed, for each global enrichment Φ_m a set of additional enrichment basis functions is generated and correspondingly some DOFs are added. Mesh elements might therefore have a variable number of DOFs, depending on the number of enriched DOFs and additional basis functions hosted. Hence, mesh elements are classified as follows: 1) standard elements, if no enrichment acts on the element; 2) reproducing elements, if the full set of DOFs is enriched with a given enrichment Φ_m; 3) blending elements, if only some DOFs are enriched with a given function Φ_m [9]. Figure 4.5 depicts this classification in the case of a single trace with first order triangular elements. Note that each mesh element can be involved by several enrichments, and it can be of different type in relation to different enrichment functions. The behaviour of enrichment function Φ_m can be correctly reproduced only in reproducing elements, where the whole set of enrichment basis functions is available, whereas in blending elements only a partial reconstruction of Φ_m is possible, and spurious terms are generated which might affect the optimal convergence rates expected for standard finite elements of the same order. At the same time, blending elements, sharing the DOFs of neighbouring reproducing elements preserve the continuity of the numerical solution.

In order to restore optimal convergence rates, a modified version of the XFEM is

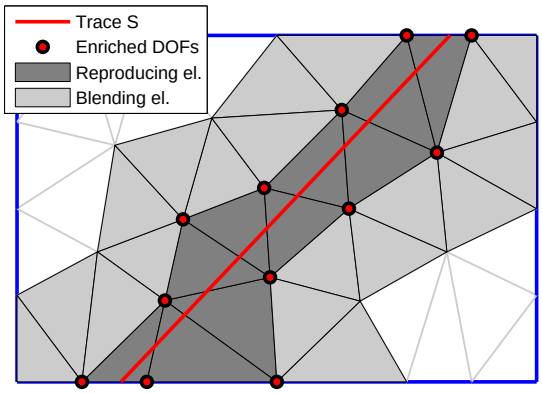


Figure 4.5: Classification of mesh elements.

adopted, as suggested in [9]. The global enrichment functions Φ_m are replaced by functions $\Phi_m R_m$ where $R_m = \sum_{k \in \mathcal{J}_m} \phi_k$ and is called ramp function. The set of active DOFs, $\mathcal{J}_m$, is replaced by the set $\tilde{\mathcal{J}}_m = \left\{k \in \mathcal{I} : \Delta_k \cap \bar{\Delta}_m \neq \emptyset\right\}$. With these choices, elements formerly classified as blending elements, become reproducing elements for the modified enrichment functions $\Phi_m R_m$, thus avoiding spurious terms, and thanks to the structure of ramp functions the continuity of the solution is preserved.

It is also beneficial to introduce a shifting of the enrichment basis functions to restore the Lagrangian property to the discrete functional space. The XFEM test and trial space is then:

$$V_\delta^{xfem} = span\left(\{\phi_k\}_{k\in\mathcal{I}}, \left\{\phi_k\left(\Phi_1 R_1 - \Phi_1(\hat{x}^k)R_1(\hat{x}^k)\right)\right\}_{k\in\mathcal{J}_1}, \ldots, \left\{\phi_k\left(\Phi_{M_F} R_{M_F} - \Phi_{M_F}(\hat{x}^k)R_{M_F}(\hat{x}^k)\right)\right\}_{k\in\mathcal{J}_{M_F}}\right),$$

where $\hat{x}^k$ is the node such that $\phi_k(\hat{x}^k) = 1$.

4.3.3 Ill conditioning prevention

The XFEM stiffness matrix (here and in the following denoted by A) might result ill conditioned or even singular due to the presence of redundant basis functions in the enriched functional space V_δ^{xfem}. When two (or more) parallel traces are present in the

same mesh element, the local enrichment functions are generated starting from global functions that differ only for a translation. This translation, besides being necessary in order to reproduce the behaviour of the solution, is also enough to provide linear independence of enrichment functions in the mesh element under consideration. On the other hand, linear dependencies in the local enrichment functions of neighbouring elements can arise. Almost parallel traces may also result in a ill-conditioned stiffness matrix, or even numerically sigular. Preventive detection of redundant basis functions, which is a typical choice in some cases [9], is infeasible in this context due to the complex geometrical configuration of realistic DFNs. For this reason, we adopt here a different approach which consists in detecting (almost) linearly dependent rows and columns in A after having assembled the matrix on each fracture. This is done operating a rank revealing QR-factorization of A (see for example [11]), exploiting the special structure of the stiffness matrix. Indeed, while referring the reader to the Appendix for details, we briefly mention here that the matrix A is a block diagonal matrix, being the A_i block given by the stiffness matrix built on fracture F_i. Therefore, the QR factorization is actually independently computed for each diagonal block, and since on each fracture we have a moderate amount of DOFs, the cost for computing the QR factorizations is acceptable. After having computed the rank revealing QR factorization for each diagonal block, i.e. $A_i = Q_i R_i$, with diagonal entries of the upper triangular matrix R_i in descending order with respect to their absolute value, we neglect rows and columns corresponding to diagonal entries with modulus lower than a given tolerance. Factors Q_i and R_i are then used in the resolution of the linear systems. In order to reduce computational cost, this procedure is performed only for fractures with parallel traces far from each other less than maximum element diameter, since the detection of parallel traces and computation of their distance is a cheap task.

4.4 Solution of the optimization problem

As shown in Section 7.2, the problem has been reformulated as a PDE-constrained optimization problem (see equation (7.7)) in which the quadratic functional J has to be minimized subject to linear constraints. In this section, following a first-discretize-then-optimize approach, we give some details about the numerical approach for computing a solution to the problem.

While referring the reader to the Appendix for all the details, we just sketch here

the structure of the finite dimensional optimization problem to be solved.

Let us introduce a finite dimensional basis on each fracture F_i and on each trace S_m, with a total number of N^F DOFs on fractures and N^T DOFs on traces. Referring to equation (4.6), we 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 set of traces. With suitable definitions, given in the Appendix, the functional J is written

$$J(h,u) := \frac{1}{2}h^T G^h h - \alpha h^T B u + \frac{1}{2}u^T G^u u, \tag{4.10}$$

where $G^h \in \mathbb{R}^{N^F \times N^F}$, $G^u \in \mathbb{R}^{N^T \times N^T}$ are symmetric positive semidefinite sparse matrices, $B \in \mathbb{R}^{N^F \times N^T}$ is a sparse matrix, and vectors $h \in \mathbb{R}^{N^F}$ and $u \in \mathbb{R}^{N^T}$ collect all DOFs for the hydraulic head on fractures and for the control variable on traces, respectively. The constraints are written

$$Ah - \mathcal{B}\,u = q, \tag{4.11}$$

where $A \in \mathbb{R}^{N^F \times N^F}$ is the stiffness matrix, $\mathcal{B} \in \mathbb{R}^{N^F \times N^T}$ is a sparse matrix, and $q \in \mathbb{R}^{N^F}$ is a vector which accounts for possible source terms and boundary conditions. The problem under consideration is therefore the equality constrained quadratic programming problem

$$\min\ J(h,u) \text{ subject to } (7.21) \tag{4.12}$$

The first order optimality conditions for problem (7.20) are the following:

$$\begin{pmatrix} G^h & -\alpha B & A^T \\ -\alpha B^T & G^u & -\mathcal{B}^T \\ A & -\mathcal{B} & 0 \end{pmatrix} \begin{pmatrix} h \\ u \\ -p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix} \tag{4.13}$$

being p the vector of Lagrange multipliers.

The previous saddle point problem is known to be a symmetric indefinite system. Note that it is 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

$$h = A^{-1}(\mathcal{B}\,u + q), \tag{4.14}$$

we obtain the following equivalent unconstrained problem :

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

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

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

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

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

and recalling (7.23), one has

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

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

$$A^T p = G^h h - \alpha Bu. \tag{4.15}$$

We have

$$\nabla\hat{J}(u) = \mathcal{B}^T p + G^u u - \alpha B^T h. \tag{4.16}$$

Note that setting to zero previous expression for obtaining stationary points for $\hat{J}(u)$, and collecting such equation together with (7.23) and (7.24), we obtain system (4.13).

Concerning the numerical solution of the otimization problem, we mention here two possible approaches. The first one consists in solving the linear system (4.13). An iterative solver is clearly a recommended choice, and `symmlq` would be a suitable choice; this approach has been used in [5]. An other 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 direction of movement.

Conjugate gradient method

1. Choose an initial guess u^0
2. Compute h_0 and p_0 solving (7.23) and (7.24) and g_0 by (7.25)
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

$$\lambda_k = -\frac{d_k^T g_k}{d_k^T \hat{G} d_k}. \tag{4.17}$$

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 (see the Appendix). 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 typically are in a moderate number. Hence, these independent linear systems can be efficiently solved on parallel computers.

4.5 Numerical results

Some numerical results are now provided to show the effectiveness of XFEM implementation in the context of DFN simulations. All numerical simulations are performed with first order finite elements on triangular meshes. The presentation is organized as follows: two test problems are introduced, in order to highlight the performances of the enrichment functions and the convergence properties of the adopted XFEM; then, a simple DFN configuration with a critical geometrical configuration is used for discussing ill-conditioning issues; finally, the solution of a complex DFN configuration is shown.

4.5.1 Test problems

The first two test cases aim at showing the effectiveness of XFEM implementation in representing irregular solution on each fracture of a given DFN, therefore, a single problem of the form (4.1) is solved on a sample fracture, using the known exact value of fluxes on the traces. Results obtained with the full algorithm described in Section 4.4 are presented afterwards.

The domain of the first problem (TP1) is a single rectangular fracture $F_1 \subset \mathbb{R}^2$, with two traces S_1 and S_2, defined by:

$$F_1 = \left\{(x,y) \in \mathbb{R}^2 : x \in (0,3), y \in (0,1)\right\}$$
$$S_1 = \left\{(x,y) \in \mathbb{R}^2 : x - y - 1 = 0\right\} \qquad S_2 = \left\{\mathbf{x} \in \mathbb{R}^2 : 2 - x - y = 0\right\},$$

and $\mathcal{S} = S_1 \cup S_2$. The domain is shown in Figure 4.6 where a coarse mesh with parameter $\delta_{max} = 0.25$ is also plotted. Here and in the sequel the mesh parameter corresponds to the square root average area of the mesh elements. The problem is set as follows:

$$\begin{aligned} -\Delta H_1 &= -\Delta H_1^{ex} && \Omega \setminus \mathcal{S}, \\ H_1 &= 0 && on\ \partial F_1, \\ U_1 &= f_{S_1} && on\ S_1, \\ U_2 &= f_{S_2} && on\ S_2, \end{aligned}$$

with

$$H_1^{ex}(x,y) = \begin{cases} xy(y-1)(x-y-1)(x+y-2)|A_2|/(4c_1) & \text{in } A_1, \\ (1-y)(x-y-1)(x+y-2) & \text{in } A_2, \\ y(x-y-1)(x+y-2) & \text{in } A_3, \\ y(1-y)(x-3)(x-y-1)(x+y-2)|A_3|/(4c_2) & \text{in } A_4, \end{cases}$$

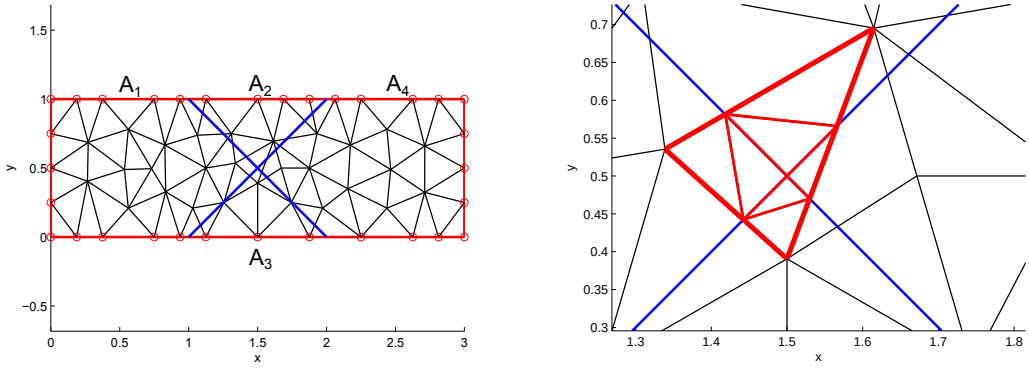


Figure 4.6: Problem TP1. Domain with coarse grid $\delta_{max} = 0.25$. Right: a detail of sub-elements division.

where A_1, A_2, A_3 and A_4 denote the four regions in which F_1 is divided by the traces, as indicated in Figure 4.6, and c_1 and c_2 are two constants used to rescale the solution. We set $f_{S_1} = \left[\!\left[\frac{\partial H_1^{ex}}{\partial \tilde{\nu}_{S_1}}\right]\!\right]_S$ and $f_{S_2} = \left[\!\left[\frac{\partial H_1^{ex}}{\partial \tilde{\nu}_{S_2}}\right]\!\right]_S$. We set $c_1 = 7$ and $c_2 = 5$ and being $|A_2| = |A_3| = 1/4$ we have

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

and

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

In Figure 4.6, right, a detail of traces intersection is given: in particular, for the element containing the intersection, the sub-elements introduced for quadrature are shown. Figure 4.7 reports the analytical solution, while Figure 4.8 displays the numerical solution on a fine mesh with parameter $\delta_{max} = 0.1$. On elements cut by the traces, the solution is represented using the same sub-elements introduced for quadrature. We can notice that the irregular trend across traces is well reproduced, without requiring any conformity between mesh elements and traces.

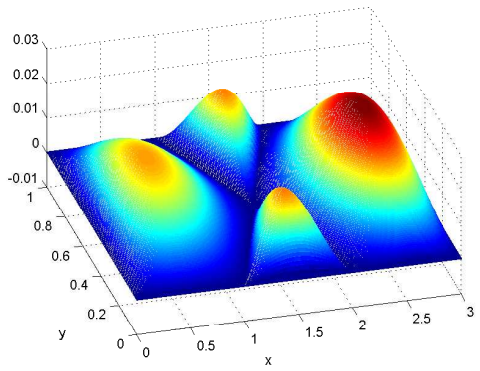


Figure 4.7: Problem TP1. Exact solution.

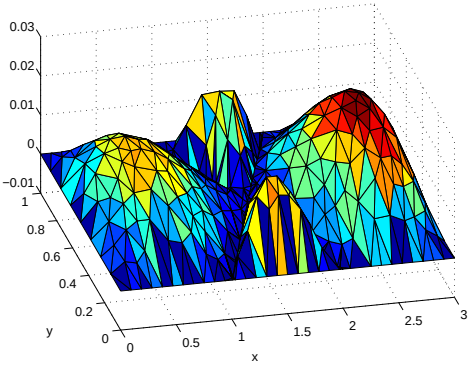


Figure 4.8: Problem TP1. Numerical solution with XFEM on the mesh with $\delta_{max} = 0.1$.

We consider now a modified version of TP1, problem TP1-X7, in which the angle underlying the intersecting traces is rather small (7° instead of 90°). This is a potentially critical situation. The configuration is shown in Figure 4.9. The two problems, original TP1 and TP1-X7, are solved both with the XFEM on nonconforming grids and standard finite elements on conforming grids. Figure 4.10 shows the L^2 and H^1-error norms against grid refinement, with grid parameters ranging from $\delta_{max} = 0.32$ to $\delta_{max} = 0.025$. In the original TP1 problem (curves labelled $X90°$ in Figure 4.10), the behaviour of XFEM and FEM is comparable, with convergence orders that approach the optimal values for both H^1 and L^2 error norms. When the angle between traces reduces (curves $X7°$), the performance of standard finite elements in H^1 norm deteriorates, while it remains unaffected for the XFEM. This is an expected behaviour and is a consequence of the poor quality of the conforming mesh for standard finite elements.

The second test problem (TP2) considers a trace ending inside the fracture, i.e. an open interface. This test problem has been considered also in [5] with different tip enriching functions, in order to analyze behavior of the solution close to an open interface. Here again we want to show quality of the solution but with the different enrichment functions here adopted, as now the same enrichment function (4.3) is used to describe the behaviour of the solution close to trace tips and away from trace tips. Furthermore, for each trace tip, just one enrichment function is used here instead of

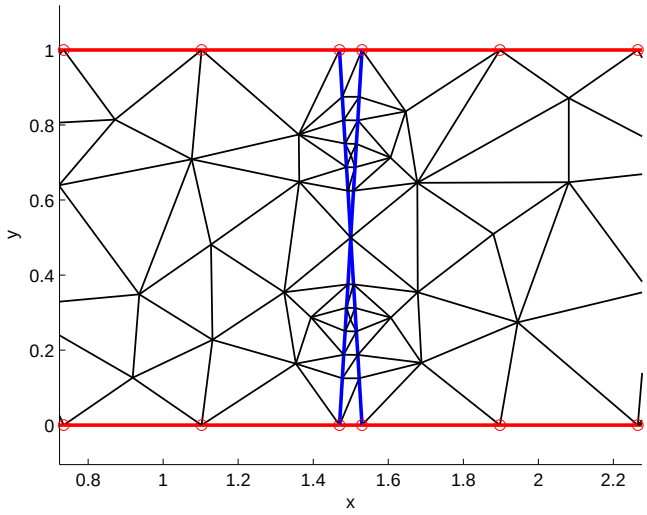


Figure 4.9: Problem TP1-X7. Detail of a conforming mesh with $\delta_{max} = 0.25$.

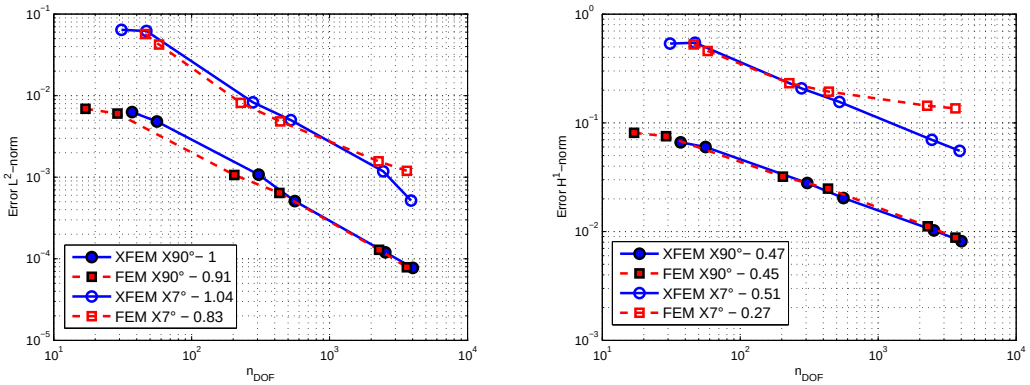


Figure 4.10: Problems TP1 and TP1-X7. Convergence curves in L^2 (left) and H^1-norm (right).

three as in [5, 6]. Let us define the domain F_2 as

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

with a single trace $S = \left\{(x, y) \in \mathbb{R}^2 : y = 0 \text{ and } -1 \leq x \leq 0\right\}$ thus ending in the interior of F_2. We introduce the function $H_2^{ex}(x, y)$ in F_2 as:

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

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 $\arctan(\cdot)$ for placing the angle in the correct quadrant. The function H_2 is the solution of the system:

$$\begin{aligned} -\Delta H_2 &= -\Delta H_2^{ex} && \text{in } \Omega \setminus S, \\ H_2 &= 0 && \text{on } \partial F_2, \\ U &= x - x^3 && \text{on } S, \end{aligned}$$

where U is the exact value of the jump of fluxes across the trace S. In Figure 7.6 we report the numerical solution obtained with the XFEM on a nonconforming grid with $\delta_{\max} = 0.1$, while in Figure 4.12 error norms for the numerical solution are shown both with the XFEM and with standard finite elements on conforming grids. The curves are perfectly overlapped and convergence orders reported in the figure are optimal, thus proving good approximation capabilities for the chosen enrichments.

4.5.2 DFN problems

We now show some numerical results on DFN-like configurations obtained with the PDE constrained optimization method described in Section 4.4. Here we focus on the main aspects related to the use of extended finite elements, referring to [5, 6] for a detailed analysis of the behaviour of the optimization algorithm.

The first example of this section, problem DFN3, is a simple network composed of three fractures as shown in Figure 4.13. Here $\Omega = F_1 \cup F_2 \cup F_3$; $S_1 = F_1 \cap F_2$; $S_2 = F_1 \cap F_3$. We solve $-\Delta H = 0$ in $\Omega \setminus (S_1 \cup S_2)$, with Dirichelet boundary conditions $H_{|\Gamma_{D,1}} = 1$ on $\Gamma_{D,1}$, $H_{|\Gamma_{D,2}} = 1.5$ on $\Gamma_{D,2}$, $H_{|\Gamma_{D,3}} = -0.5$ on $\Gamma_{D,3}$ and homogeneous Neumann boundary conditions on the other sides (see Figure 4.13). This configuration reproduces a critical situation for the fracture F_1, in which two parallel traces very

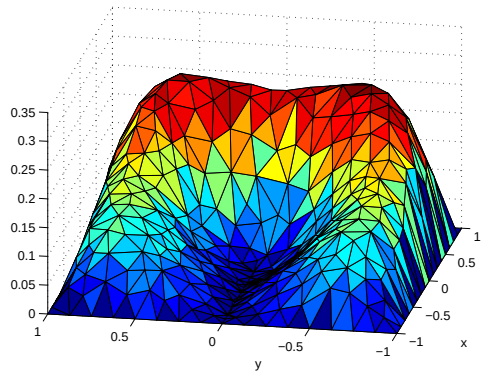


Figure 4.11: Problem TP2. Numerical solution with XFEM on a non-conforming grid with $\delta_{max} = 0.1$

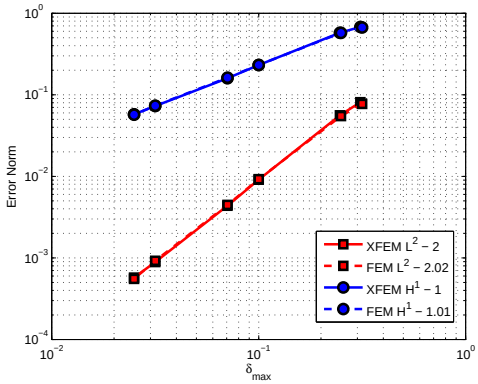


Figure 4.12: Problem TP2. H^1 and L^2 error norms against grid refinement for XFEM and FEM.

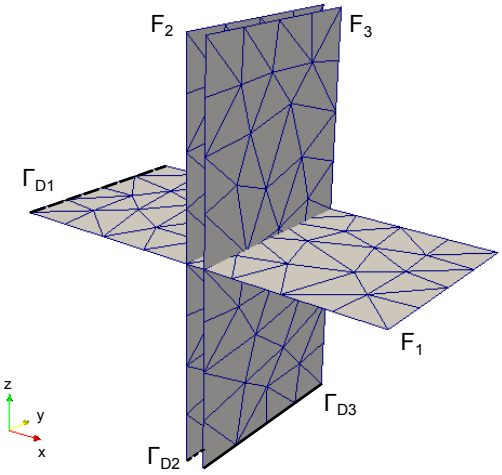


Figure 4.13: Problem DFN3. DFN configuration.

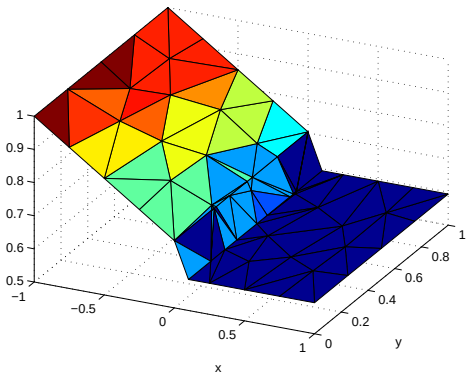


Figure 4.14: Problem DFN3. Solution on fracture F_1.

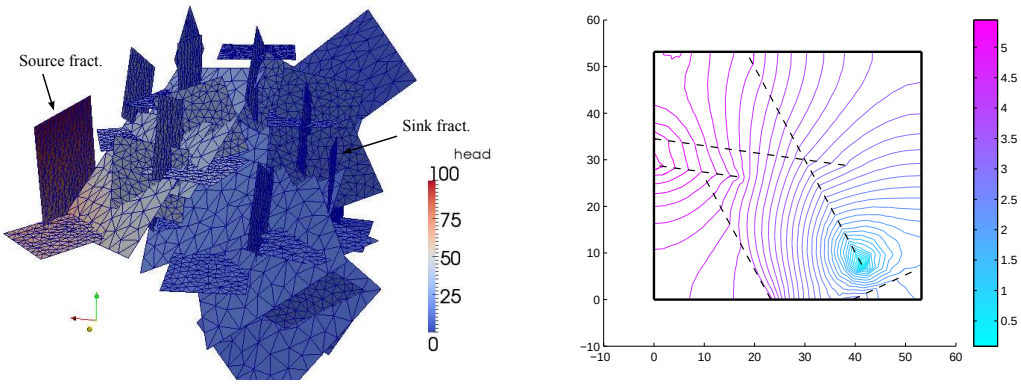


Figure 4.15: Problem DFN40. Left: DFN configuration and solution (colorbar). Right: hydraulic head isolines on a selected fracture

close each other are present. A conforming mesh would be constrained by the presence of these traces, with a large number of elements to be placed between the traces in order to preserve quality. The XFEM do not require a conforming mesh, but in this case the set of local enrichment functions introduced for the two traces could be not linearly independent, as detailed in Subsection 4.3.3. Applying the described strategy for redundant basis functions removal with a tolerance of 10^{-14}, a new matrix with a condition number of 10^4 is extracted from the formerly singular stiffness matrix of the proposed problem, removing four redundant DOFs. The quality of the solution is not affected as shown by Figure 4.14, where the solution on F_1 is plotted. It can be noticed that the numerical approximation reproduces the expected behaviour for the exact solution that is piecewise linear and displays jumps of derivatives in the direction normal to the traces. Since the solution belongs to the discrete subspace spanned by the FEM and XFEM basis functions, the exact solution is correctly reproduced up to machine error.

We finally present the numerical results on a realistic DFN configuration composed of 40 fractures and 96 traces (problem DFN40). The fractures have an average size of 4×10^3 m^2. The problem is solved with several non-conforming meshes with maximum element sizes ranging from 2 to 25m^2. As in problem DFN3, a simple Laplace problem for the hydraulic head is considered, with forcing term equal to zero and constant Dirichelet boundary conditions applied to one edge of a "source" fracture ($H = 100$) and of a "sink" fracture ($H = 0$). All other edges are treated as insulated, imposing homogeneous Neumann boundary conditions. Figure 4.15, left, shows the geometrical configuration

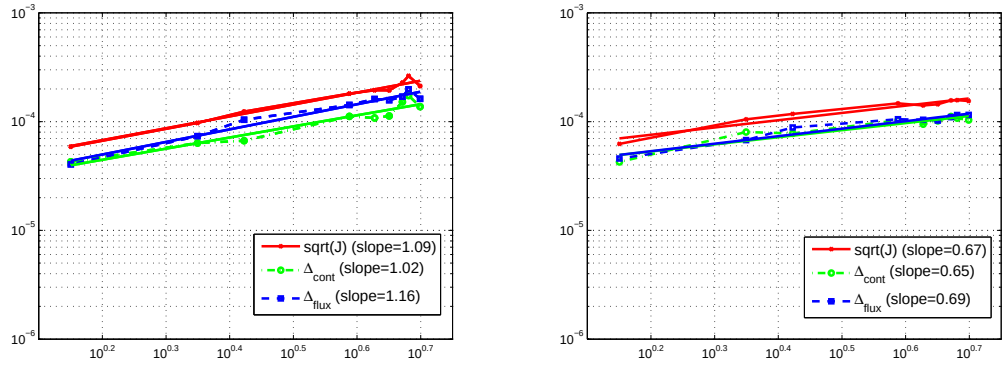


Figure 4.16: Problem DFN40. Convergence history for global continuity error and flux mismatch. Left: XFEM; right: FEM

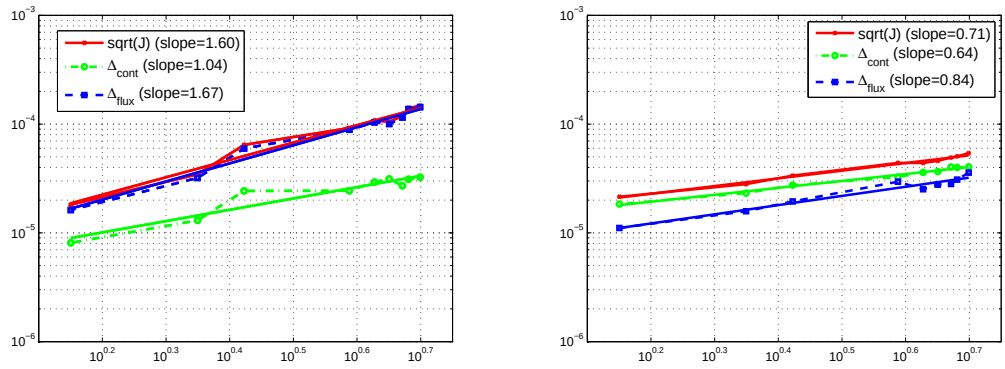


Figure 4.17: Problem DFN40. Convergence history for global continuity error and flux mismatch. Left: XFEM; right: FEM. Grids on traces twice as fine as the previous case

of the DFN along with a shading of the obtained solution on the 7m^2 elements mesh, while in Figure 4.15, right, we plot isolines for hydraulic head computed on a selected fracture with the same mesh. Dashed lines in this figure represent traces on the fracture. It can be noticed that the isolines have sudden variations across the traces, showing that discontinuities in gradients are correctly reproduced by the XFEM.

Finally, we analyze on problem DFN40 the numerical conservation properties of the method, using both enriched and non-enriched basis. Indeed, we recall that our approach does not exactly impose matching conditions (4.2) and (4.3), but it minimizes the sum of global continuity error and flux mismatch. The label FEM in the table and figures which follow, refers to results obtained with the optimization approach on non-

Table 4.1: Discontinuity errors and flux mismatches

	XFEM		FEM	
40fract				
Grid	Δ_{cont}	Δ_{flux}	Δ_{cont}	Δ_{flux}
25	1.375e-04	1.623e-04	1.033e-04	1.154e-04
23	1.738e-04	1.979e-04	1.077e-04	1.151e-04
22	1.520e-04	1.698e-04	1.116e-04	1.101e-04
20	1.128e-04	1.577e-04	1.016e-04	1.024e-04
18	1.081e-04	1.616e-04	9.477e-05	1.041e-04
15	1.117e-04	1.425e-04	1.029e-04	1.053e-04
7	6.675e-05	1.041e-04	7.787e-05	8.834e-05
5	6.362e-05	7.359e-05	8.032e-05	6.766e-05
2	4.274e-05	4.055e-05	4.253e-05	4.580e-05
40fract2x				
Grid	Δ_{cont}	Δ_{flux}	Δ_{cont}	Δ_{flux}
25	3.251e-05	1.433e-04	4.053e-05	3.582e-05
23	3.109e-05	1.373e-04	3.997e-05	3.09e-05
22	2.712e-05	1.152e-04	4.031e-05	2.82e-05
20	3.140e-05	1.005e-04	3.665e-05	2.776e-05
18	2.936e-05	1.039e-04	3.600e-05	2.521e-05
15	2.439e-05	8.868e-05	3.263e-05	2.956e-05
7	2.432e-05	5.973e-05	2.747e-05	1.945e-05
5	1.304e-05	3.202e-05	2.316e-05	1.579e-05
2	8.095e-06	1.624e-05	1.842e-05	1.110e-05
40fract3x				
Grid	Δ_{cont}	Δ_{flux}	Δ_{cont}	Δ_{flux}
25	1.946e-05	1.329e-04	3.503e-05	1.776e-05
23	1.969e-05	1.262e-04	3.326e-05	1.635e-05
22	1.696e-05	1.121e-04	3.408e-05	1.736e-05
20	1.779e-05	1.012e-04	3.137e-05	1.571e-05
18	1.764e-05	1.016e-04	3.099e-05	1.453e-05
15	1.719e-05	7.957e-05	2.772e-05	1.624e-05
7	1.522e-05	5.072e-05	2.521e-05	1.301e-05
5	9.098e-06	2.631e-05	2.099e-05	8.104e-06
2	6.608e-06	1.594e-05	1.613e-05	6.373e-06

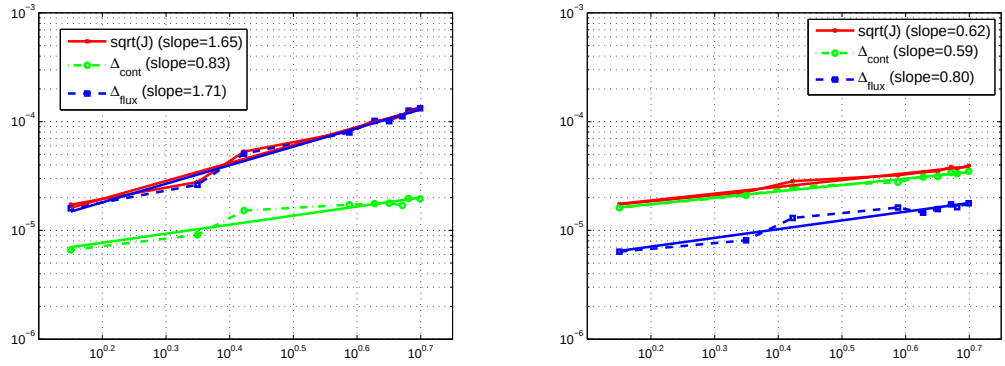


Figure 4.18: Problem DFN40. Convergence history for global continuity error and flux mismatch. Left: XFEM; right: FEM. Grids on traces three times as fine as the previous case

conforming meshes without enrichment functions. In Table 4.1 we report values of the total continuity error and the total flux mismatch relative to total trace length, defined respectively as:

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

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

The table referes to all the non-conforming meshes used on fractures both using enrichment functions (XFEM label) and without enrichments (FEM label), and to three different grids used on traces obtained doubling (label 40fract2x) and tripling (label 40fract3x) the initial number of DOFs for the control variables on the traces. Figures 4.16-4.18 show, under fracture mesh refinement, the convergence behaviour of global continuity error and flux mismatch. The figures also show the behaviour of $\sqrt{J}$, again relative to total trace length. Abscissas correspond to the square root of the maximum element sizes. Despite on coarser grid the starting mismatch errors are larger for XFEM, it can be noted that for XFEM vanishing rates (the slopes reported in the legend of the figures) are close to 1, whereas for FEM it is closer to 0.5. Concerning refinement of trace grids, it can be seen that, as expected, flux mismatch benefits from refinement to a larger extent with respect to continuity error.

4.6 Conclusions

The use of the XFEM for DFN simulations is very promising for the possibility of using non-conforming meshes on the fractures but a number of issues are to be considered in order to ensure an effective implementation. In the present work we address some of them.

The enrichment functions suggested have a very simple structure and represent a unifying approach to handle open, closed and intersecting interfaces, thus simplifying implementation, limiting the computational cost for the enrichment part of the approximation and still ensuring good accuracy for DFN simulation purposes.

A thorough description of the implementation strategy suggested in [9] to restore optimal convergence rates is provided in the case of interest, and numerous numerical examples are reported showing the expected convergence performances.

The major source of ill-conditioning in DFN simulations is identified in the possibility of having linear dependence or almost linear dependence in the enrichment basis function space, and a strategy to overcome this problem is identified and successfully implemented.

Finally, the optimization approach results to be very effective in dealing with very complex DFNs.

4.7 Appendix

In this section we give some details concerning the discrete form (7.20) of the optimization problem (7.7).

In order to simplify the discussion, let us 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. 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 7.2 by equation

(4.6). Denoting by $\phi_{i,k}$ the k-th basis function of the XFEM approximation of h_i on fracture F_i, the discrete form of the functional is

$$J = \frac{1}{2}\sum_{i=1}^{I}\sum_{S\in\mathcal{S}_i}\left(\int_S (\sum_{k=1}^{N_i} h_{i,k}\phi_{i,k|S} - \sum_{k=1}^{N_j} h_{j,k}\phi_{j,k|S})^2\,\mathrm{d}\gamma + \right. \tag{4.18}$$
$$\left.\int_S (\sum_{k=1}^{N_m^-} u^-_{m,k}\psi^-_{m,k} + \sum_{k=1}^{N_m^+} u^+_{m,k}\psi^+_{m,k} - \alpha\sum_{k=1}^{N_i} h_{i,k}\phi_{i,k|S} - \alpha\sum_{k=1}^{N_j} h_{j,k}\phi_{j,k|S})^2\,\mathrm{d}\gamma\right).$$

The first integral in (7.18) after straightforward manipulation rewrites as

$$I_{ij}^{S,1} = h_i^T C_{i,i}^S h_i + h_j^T C_{j,j}^S h_j - 2h_i^T C_{i,j}^S h_j$$

where $C_{p,q}^S$, for either $p = q$ or $p, q \in I_S$ for some trace S, is the matrix defined by

$$(C_{p,q}^S)_{k,\ell} = \int_S \varphi_{p,k|_S}\varphi_{q,\ell|_S}\,\mathrm{d}\gamma.$$

Note that since $(C_{ij}^S)^T = C_{ji}^S$, we can also write $I_{ij}^{S,1} = h_i^T C_{i,i}^S h_i + h_j^T C_{j,j}^S h_j - h_i^T C_{i,j}^S h_j - h_j^T C_{j,i}^S h_i$.

The second integral after some straightforward algebraic manipulation rewrites

$$\begin{aligned}
I_{ij}^{S,2} = {} & \sum_{k=1}^{N_m^-} {u^-_{m,k}}^2 \int_S {\psi^-_{m,k}}^2\,\mathrm{d}\gamma + 2\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_m^-} u^-_{m,k}u^-_{m,\ell}\int_S \psi^-_{m,k}\psi^-_{m,\ell}\,\mathrm{d}\gamma \\
& + \sum_{k=1}^{N_m^+} {u^+_{m,k}}^2 \int_S {\psi^+_{m,k}}^2\,\mathrm{d}\gamma + 2\sum_{k=1}^{N_m^+}\sum_{\ell=1}^{N_m^+} u^+_{m,k}u^+_{m,\ell}\int_S \psi^+_{m,k}\psi^+_{m,\ell}\,\mathrm{d}\gamma \\
& + 2\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_m^+} u^-_{m,k}u^+_{m,\ell}\int_S \psi^-_{m,k}\psi^+_{m,\ell}\,\mathrm{d}\gamma + \alpha^2\sum_{k=1}^{N_i} h_{i,k}^2\int_S {\phi_{i,k|_S}}^2\,\mathrm{d}\gamma \\
& + 2\alpha^2\sum_{k,\ell=1}^{N_i} h_{i,k}h_{i,\ell}\int_S \phi_{i,k|_S}\phi_{i,\ell|_S}\,\mathrm{d}\gamma + \alpha^2\sum_{k=1}^{N_j} h_{j,k}^2\int_S {\phi_{j,k|_S}}^2\,\mathrm{d}\gamma \\
& + 2\alpha^2\sum_{k,\ell=1}^{N_j} h_{j,k}h_{j,\ell}\int_S \phi_{j,k|_S}\phi_{j,\ell|_S}\,\mathrm{d}\gamma + 2\alpha^2\sum_{k=1}^{N_i}\sum_{\ell=1}^{N_j} h_{i,k}h_{j,\ell}\int_S \phi_{i,k|_S}\phi_{j,\ell|_S}\,\mathrm{d}\gamma \\
& - 2\alpha\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_i} u^-_{m,k}h_{i,\ell}\int \psi^-_{m,k}\phi_{i,\ell|_S}\,\mathrm{d}\gamma - 2\alpha\sum_{k=1}^{N_m^-}\sum_{\ell=1}^{N_j} u^-_{m,k}h_{j,\ell}\int \psi^-_{m,k}\phi_{j,\ell|_S}\,\mathrm{d}\gamma \\
& - 2\alpha\sum_{k=1}^{N_m^+}\sum_{\ell=1}^{N_i} u^+_{m,k}h_{i,\ell}\int \psi^+_{m,k}\phi_{i,\ell|_S}\,\mathrm{d}\gamma - 2\alpha\sum_{k=1}^{N_m^+}\sum_{\ell=1}^{N_j} u^+_{m,k}h_{j,\ell}\int \psi^+_{m,k}\phi_{j,\ell|_S}\,\mathrm{d}\gamma.
\end{aligned}$$

Let us introduce the following matrices: 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:

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

If fractures F_i and F_j share trace S_m, we define matrices $B_{i,m}^- \in \mathbb{R}^{N_i \times N_m^-}$ and $B_{i,m}^+ \in \mathbb{R}^{N_i \times N_m^+}$ defined as

$$(B_{i,m}^-)_{k\ell} = \int_{S_m} \psi_{m,k}^- \phi_{i,\ell|_{S_m}} \, \mathrm{d}\gamma, \qquad (B_{i,m}^+)_{k\ell} = \int_{S_m} \psi_{m,k}^+ \phi_{i,\ell|_{S_m}} \, \mathrm{d}\gamma.$$

An analogous definition holds for matrices $B_{j,m}^-$ and $B_{j,m}^+$. Integral $I_{ij}^{S,2}$ is then written in compact form as

$$\begin{aligned} I_{ij}^{S,2} \;=\; & (u_{m,k}^-)^T \mathcal{C}_m^- u_{m,k}^- + (u_{m,k}^+)^T \mathcal{C}_m^+ u_{m,k}^+ + 2(u_{m,k}^-)^T \mathcal{C}_m^\pm u_{m,k}^+ + \\ & \alpha^2 h_i^T C_{i,i}^S h_i + \alpha^2 h_j^T C_{j,j}^S h_j + 2\alpha^2 h_i^T C_{i,j}^S h_j - \alpha(h_i^T B_{i,m}^- u_{m,k}^-) \\ & -\alpha(h_i^T B_{i,m}^+ u_{m,k}^+) - \alpha(h_j^T B_{j,m}^- u_{m,k}^-) - \alpha(h_j^T B_{j,m}^+ u_{m,k}^+) \\ & -\alpha((u_{m,k}^-)^T (B_{i,m}^-)^T h_i) - \alpha((u_{m,k}^+)^T (B_{i,m}^+)^T h_i) \\ & -\alpha((u_{m,k}^-)^T (B_{j,m}^-)^T h_j) - \alpha((u_{m,k}^+)^T (B_{j,m}^+)^T h_j). \end{aligned}$$

We have therefore

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

We now allow for a more compact form of $J(u)$ 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 $B_{i,m}$, with 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 $N_{\mathcal{S}_i} = \sum_{S_m \in \mathcal{S}_i} (N_m^- + N_m^+)$, which acts on a column vector u_i obtained appending the blocks u_m 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 $u_i = R_i u$. Also, 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 $G^u = \text{diag}(\mathcal{C}_m, m = 1, ..., M)$ and finally set

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

Due to previous observations, matrix G is straightforwardly symmetric. Furthermore, it is positive semidefinite by construction. With these definitions at hand, the functional J is rewritten

$$J = \frac{1}{2} w^T G w, \qquad w = (h, u)$$

being h obtained appending vectors h_i, $i = 1, ..., I$.

Constraints (4.5) 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

$$(A_i)_{k\ell} = \int_{F_i} \nabla \varphi_{i,k} \nabla \varphi_{i,\ell} \, dF_i + \alpha \sum_{S \in \mathcal{S}_i} \int_S \phi_{i,k|_S} \phi_{i,\ell|_S} \, \mathrm{d}\gamma,$$

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 $\mathcal{B}_i \in \mathbb{R}^{N_i \times N_{\mathcal{S}_i}}$ 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 collecting vectors u^S_i, for $S \in \mathcal{S}_i$, with the same ordering introduced for the traces on F_i and used in the definition of $\mathcal{B}_i$.

The algebraic formulation of the primal equations (4.5) is then

$$A_i h_i = \tilde{q}_i + \mathcal{B}_i u_i, \qquad i = 1, ..., I, \tag{4.19}$$

where $\tilde{q}_i$ accounts for the term q_i in (4.5) and for the boundary conditions on the fracture F_i.

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

$$\mathcal{B} = \begin{pmatrix} \mathcal{B}_1 R_1' \\ \vdots \\ \mathcal{B}_I R_I' \end{pmatrix}$$

where the operator R_i' now extracts from u only subvectors $u_m^\star$ corresponding to control function on the "correct side" of the trace. Setting $C = (A \;\; -\mathcal{B})$ and $\tilde{q} = (\tilde{q}_1, \ldots, \tilde{q}_I)$, constraints (4.19) are then written $Cw = \tilde{q}$. The overall problem is then reformulated as follows:

$$\min_w \; \frac{1}{2} w^T G w, \qquad \text{s.t. } Cw = \tilde{q}. \tag{4.20}$$

Bibliography

[1] I. Babuška and J. M. Melenk, *The partition of unity method*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 727–758.

[2] T. Belytschko and T. Black, *Elastic crack growth in finite elements with minimal remeshing*, Internat. J. Numer. Methods Engrg., 45 (1999), pp. 601–620.

[3] T. Belytschko, N. Möes, S. Usui, and C. Parimi, *Arbitrary discontinuities in finite elements*, Internat. J. Numer. Methods Engrg., 50 (2001), pp. 993–1013.

[4] S. Berrone, S. Pieraccini, and S. Scialó, *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM Journal on Scientific Computing, 35 (2013), pp. B487–B510.

[5] ——, *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM Journal on Scientific Computing, 35 (2013), pp. A908–A935.

[6] S. Berrone, S. Pieraccini, and S. Scialò, *An optimization approach for large scale simulations of discrete fracture network flows*, SIAM J. Sci. Comput., J. Comput. Phys. (2013).

[7] C. Daux, N. Möes, J. Dolbow, N. Sukumar, and T. Belytschko, *Arbitrary branched and intersecting cracks with the extended finite element method*, Internat. J. Numer. Methods Engrg., 48 (2000), pp. 1741–1760.

[8] J. Erhel, J.-R. de Dreuzy, and B. Poirriez, *Flow simulation in three-dimensional discrete fracture networks*, SIAM J. Sci. Comput., 31 (2009), pp. 2688–2705.

[9] T.-P. Fries, *A corrected xfem approximation without problems in blending elements*, Internat. J. Numer. Methods Engrg., 75 (2008), pp. 503–532.

[10] T.-P. Fries and T. Belytschko, *The extended/generalized finite element method: an overview of the method and its applications*, Internat. J. Numer. Methods Engrg., 84 (2010), pp. 253–304.

[11] P. E. Gill, W. Murray, and M. H. Wright, *Numerical linear algebra and optimization*, Addison-Wesley Pub. Co., 1991.

[12] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[13] N. Möes, J. Dolbow, and T. Belytschko, *A finite element method for crack growth without remeshing*, Internat. J. Numer. Methods Engrg., 46 (1999), pp. 131–150.

[14] G. Pichot, J. Erhel, and J.-R. De Dreuzy, *A mixed hybrid Mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629–1643.

[15] G. Pichot, J. Erhel, and J.-R. de Dreuzy, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[16] N. Sukumar, D. Chopp, N. Möes, and T. Belytschko, *Modeling holes and inclusions by level sets in the extended finite element method*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 1209–1233.

[17] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.

Part II

Resolution of complex DFN configurations

The material collected in the present Part of the Thesis consists of a large number of very recent simulations on complex DFN configurations, and consequently the analysis of these numerical results is not sufficiently detailed and analysed in depth yet. We believe, however that the material presented can be of great help in showing the performances of the proposed method. Also, through the analysis of these results many implementation details can be highlighted and discussed.

Chapter 5

On the resolution of complex DFN configurations

This chapter is devoted to the presentation of a large number of numerical results obtained with complex DFN configurations, collecting and describing in a systematic way the performances of different implementation choices for the optimization algorithm described in Section 4.4. Further we show a preliminary investigation on the scalability properties of the proposed method.

On a mesh non conforming to the traces, the use of the enrichment functions of the XFEM can give an accurate description of the solution around the traces, as thoroughly discussed in the previous chapters. Standard finite elements can also be used on the same nonconforming computational mesh, with the advantage of a slightly reduced number of unknowns (the degrees of freedom related to the enrichment functions) but at the expenses of a less accurate representation of the result. This possibility was already discussed in Chapter 2 and some results are also shown in Chapter 3. Here a deeper analysis is presented and numerical results on realistic DFN configurations are provided and discussed with both these approaches. The description of a the method with a different discretization choice involving the new Virtual Element Method is deferred to Chapter 7.

The discretization of the control variables on the traces can be performed completely independently from the discretization on the fractures. The discrete functional space for the control variables chosen is the space of discontinuous piecewise linear polynomials, and two different node dispositions are envisaged. Let us consider a generic trace $S =$

$\bar{F}_i \cap \bar{F}_j$ in a DFN, we can have:

- a disposition of nodes on S for u_i^S and u_j^S given by the intersection points between S and the element edges of the nonconforming meshes on F_i and F_j respectively; this node configuration is called *induced*, labelled IN;

- or simply equally spaced nodes on S for u_i^S and u_j^S, completely independent from the discretization of the state-variables on the fractures; this strategy is termed *equally-spaced*, label EN.

Results are described with both these configuration of nodes.

The quality of the obtained solutions is evaluated in terms of three error indicators, Δ_{cont}, Δ_{flux} and $\Delta_{\text{source-sink}}$, as detailed in the following. The two first indicators measure how well the numerical solution satisfies the continuity and flux balance conditions across the traces, while the third indicator evaluates the global mismatch between the flux injected in the network of fractures and the total flux that leaves the network through the non insulated fracture edges.

After a description of the various DFN configurations considered, some results on the conditioning of the problem in relation to key parameters are presented in Section 5.2. Numerical simulations are then shown and discussed in Section 5.3 and in Section 5.4 where DFNs with non-uniform fracture transmissivity are considered and some conclusions on convergence properties of the method are also proposed. Section 5.5 ends this Chapter reporting some results on a preliminary investigation on the scalability of the proposed approach.

5.1 Problems description

The panel of problems considered is composed of six DFN configurations of increasing complexity, as summarized in Table 5.1. In the networks considered, fractures have dimensions ranging between 2.8×10^3 m^2 and 1.2×10^4 m^2 and traces intersecting in fractures form angles of about 35°, 45°, 55°, 70° or 90°, while the minimum distance between non intersecting traces varies between 0.5 m and 1.1×10^2 m. Trace length spans between 4.2×10^{-2} m to 2.3×10^2 m. All the DFNs share the same two boundary fractures F_1 and F_2, while all the other fractures might be different from a system to another. Homogeneous Dirichelet boundary condition is prescribed on one edge of fracture F_1

Table 5.1: DFN configurations

Label	N° of fractures	N° of traces
11F	11	13
27F	27	57
36F	36	65
55F	55	120
68F	68	142
120F	120	256

(*sink* fracture), while a constant value Dirichelet condition of 100m is prescribed on one edge of F_2 (*source* fracture) for all systems. All other fracture edges are insulated.

Numerical simulations are performed with first order finite elements and triangular meshes for the state-variable h on the fractures and results are reported using both the XFEM on nonconforming grids and with the standard FEM on the same nonconforming meshes. The discrete subspace of the control variable u is chosen as the space of discontinuous piece-wise linear polynomials with induced or equally-spaced nodes. When equally-spaced nodes are used the number of nodes can be arbitrarily chosen. We define a reference number of nodes for the equally-spaced configuration as a number of nodes close to the number of nodes of the discretization induced and a parameter n_U is introduced to express the number of equally-spaced nodes in terms of the ratio with the reference value.

The computational mesh is identified by means of the maximum element area, and labelled in the figures with this value without unit of measure (m^2). Meshes with maximum element area ranging between $120m^2$ and $7m^2$ are considered.

5.2 Study of system conditioning

In Section 7.2 we have formally written the unconstrained formulation of the DFN problem with the proposed approach, and an explicit formulation of the unconstrained functional gradient, (7.25):

$$\begin{aligned} \nabla \hat{J}(u) &= (\mathcal{B}^T A^{-T} G^h A^{-1} \mathcal{B} + G^u - \alpha(\mathcal{B}^T A^{-T} B + B^T A^{-1} \mathcal{B}))u + \\ &\quad (\mathcal{B}^T A^{-T} G^h - \alpha B^T) A^{-1} q \\ &= \hat{G}u + \hat{q}. \end{aligned} \tag{5.1}$$

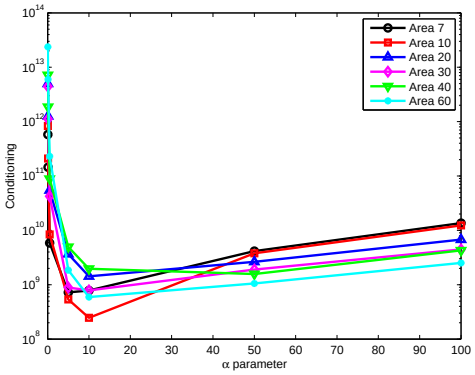


Figure 5.1: Condition number of the 27F DFN system matrix for α ranging from 0.05 to 100. Induced nodes

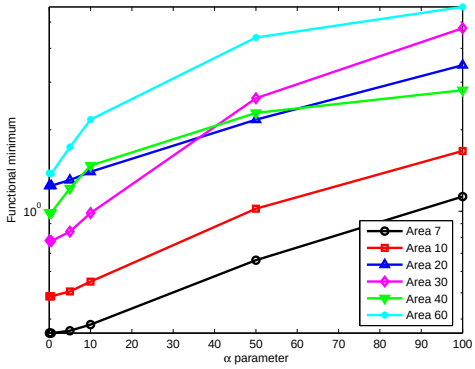


Figure 5.2: Stagnation functional values of the 27F DFN for α ranging from 0.05 to 100. Induced nodes

For small DFN configurations it is possible to resort to this formulation in order to evaluate the effect of some implementation choices on the conditioning of the discrete problem, analysing the conditioning of matrix $\hat{G}$. The DFN 27F, 36F and 68F are considered in this analysis, with induced or equally-spaced nodes for the control variables.

Figures 5.1-5.6 show the behaviour of the condition number of $\hat{G}$ and of functional minimum in logarithmic scale for different values of the parameter α appearing in the definition of the control variable U given in Chapter 3, and of mesh element maximum area. In these figures the XFEM is chosen for the description of the solution and induced nodes are used on the traces. Looking at Figures 5.1, 5.3 and 5.5 we can see that, for each configuration and mesh there is an optimal value of α for good conditioning. This optimal value is contained in a range of values of few units for all the cases considered, with a weak dependence from the size of the mesh or from the complexity of the problem. Concerning functional values, Figures 5.2, 5.4 and 5.6, show that lower minimum values are reached reducing α. However over the entire range of α values considered, the variations of functional is quite small for all the problems and grids. A possible optimal choice appears to be $\alpha = 1$, since this value gives low condition numbers and functional minimum and has the desirable property of reducing matrix G^h block diagonal, as follows immediately from the definition given in Section 4.7. This value is used to obtain all the results presented in this Chapter.

Figures 5.7-5.12 show the condition number and functional minimum in function

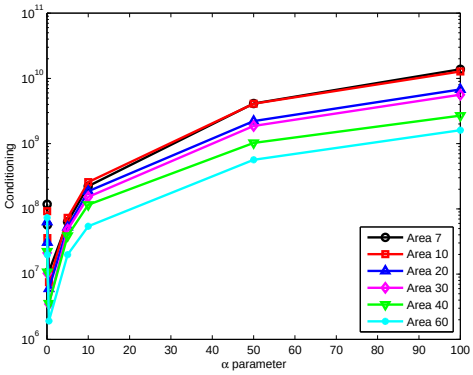


Figure 5.3: Condition number of the 36fract DFN system matrix for α ranging from 0.05 to 100. Induced nodes

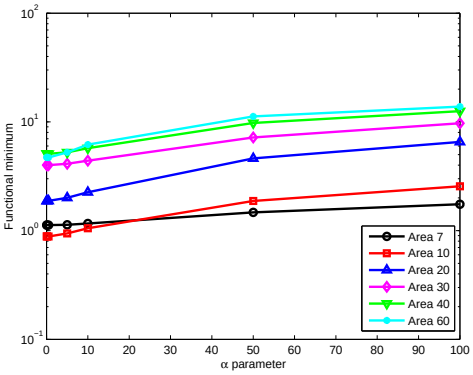


Figure 5.4: Stagnation functional values of the 36fract DFN for α ranging from 0.05 to 100. Induced nodes

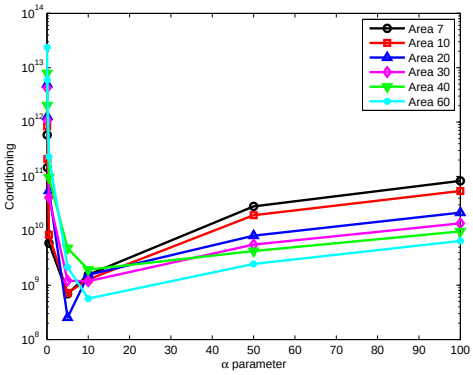


Figure 5.5: Condition number of the 68F DFN system matrix for α ranging from 0.05 to 100. Induced nodes

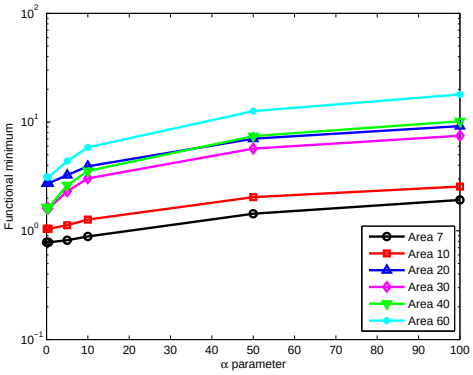


Figure 5.6: Stagnation functional values of the 68F DFN for α ranging from 0.05 to 100. Induced nodes

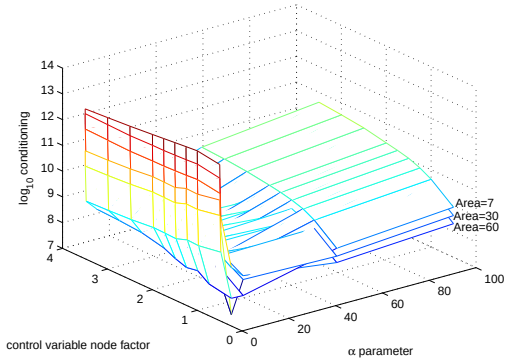


Figure 5.7: Condition number of the 27F DFN system matrix for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

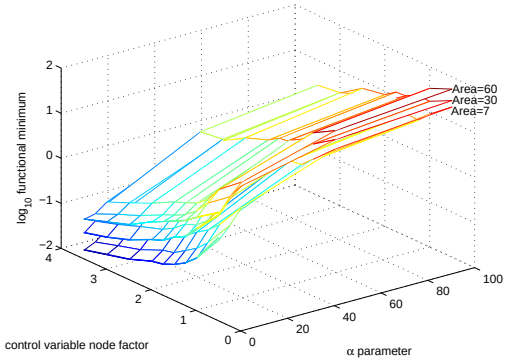


Figure 5.8: Stagnation functional values of the 27F DFN for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

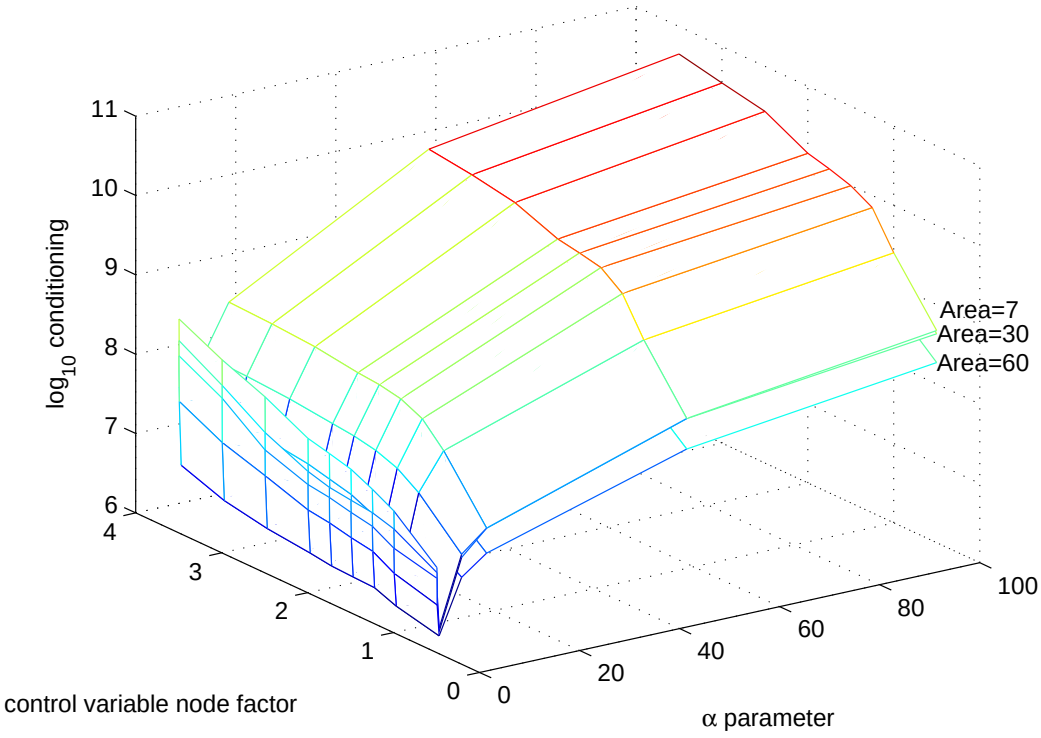


Figure 5.9: Condition number of the 36F DFN system matrix for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

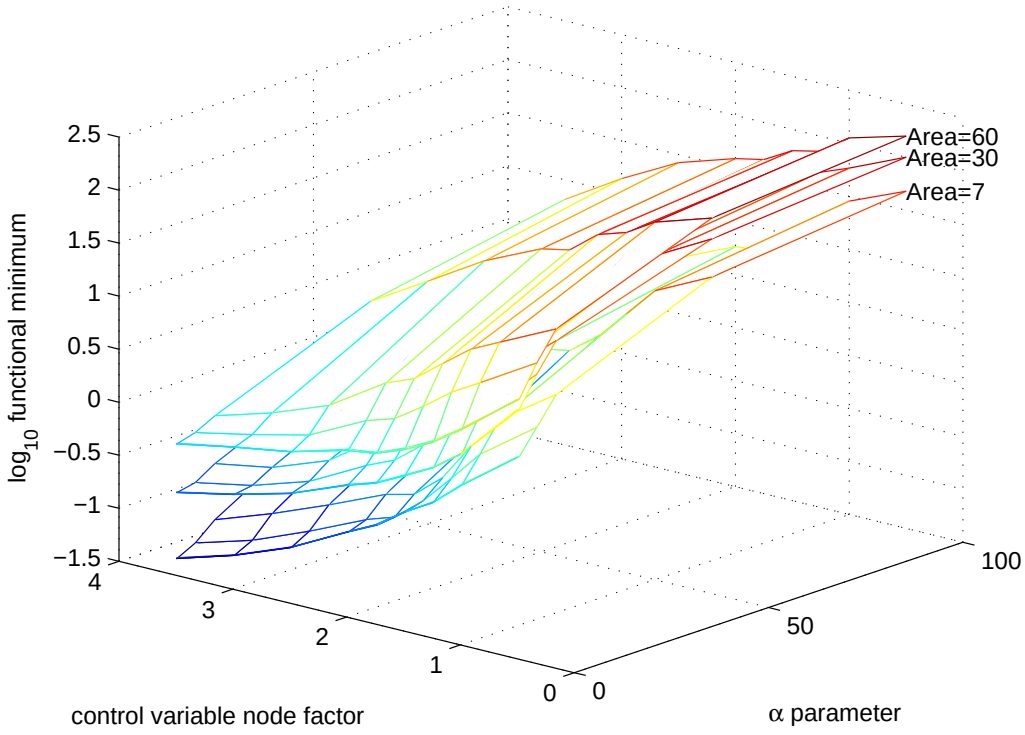


Figure 5.10: Stagnation functional values of the 36F DFN for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

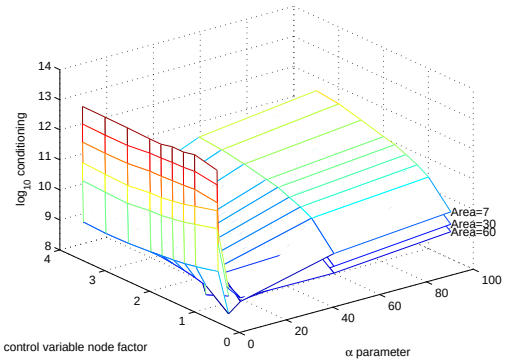


Figure 5.11: Condition number of the 68F DFN system matrix for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

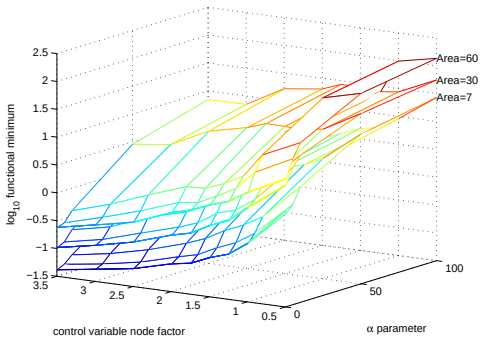


Figure 5.12: Stagnation functional values of the 68F DFN for α ranging from 0.05 to 100 and n_U from 0.5 to 3.5. Equally-spaced nodes

of α and of the number of DOFs for the control variables, expressed in terms of the parameter n_U. In this case equally-spaced nodes are placed on the traces and the XFEM is used for the discretization of h. Looking at Figures 5.7, 5.9 and 5.11 we can see that increasing the number of nodes for the control variable has a detrimental impact on the conditioning of the system, but with a moderate trend, and this in independent of the value of α and of problem complexity or mesh size. At the same time higher values of n_U give lower functional minimum for all the configurations examined, as Figures 5.8, 5.10 and 5.12 show. For these reasons increasing n_U is a viable option for improving the quality of the solution, clearly at the cost of an increase in the number of unknowns. The effect of variations of α is similar to the previous case. More in general significant differences are not observed between equally-spaced and induced node strategies.

Concerning the effect of mesh size on the conditioning of the system, it is possible to conclude that reducing mesh size has the effect of an increase of the condition number of the problem, as expected. At the same time functional minimum can be reduced by mesh refinement.

DFNs with a larger number of fractures are expected to have a worse conditioning that simpler configurations as can be seen comparing, for example, Figure 5.9 with Figure 5.11, but this is not true in general, as can be noticed comparing Figure 5.1 with Figure 5.3 or Figure 5.7 with Figure 5.9. The 27F DFN configuration has a higher trace-to-fracture ratio than the 36F DFN, as can be seen looking at Table 5.1, such that an influence of this parameter on system conditioning could be envisaged. The influence of trace-to-fracture ratio to problem conditioning has not been investigated and could be the object of a deeper analysis.

Some results on system conditioning when standard finite elements are used for the discretization of the solution on the fractures are reported in Table 5.2 for the 36F DFN with equally-spaced nodes, compared to the results obtained with the XFEM on the same grid with maximum element area of 30m^2. The two approaches have a similar impact on the conditioning of the discrete problem.

Table 5.2: Condition numbers for the DFN 36F with XFEM and FEM for different values of n_U. Equally-spaced nodes, Area=30.

n_U	XFEM	FEM
0.5	1.75973e+06	2.64085e+06
1.0	3.22126e+06	3.66881e+06
1.5	4.50964e+06	3.34477e+06
2.0	5.24292e+06	3.96676e+06

5.3 Numerical results with constant fracture transmissivity

This Section shows some numerical results on the DFN configurations named 27F, 36F, 68F and 120F in Table 5.1 with an uniform distribution of the fracture transmissivity $\mathbf{K} = 1$ across the fractures. Three different mesh sizes are considered, with maximum element area of 120m^2, 30m^2 and 7m^2.

Table 5.3: Results for DFNs 27F, 36F, 68F and 120F with nodes IN and EN. XFEM and standard FEM compared.

		XFEM			FEM		
Grid	**Node**	Δ_{cont}	Δ_{flux}	**Iter**	Δ_{cont}	Δ_{flux}	**Iter**
		27 fractures EN					
120	725	0.0009953	0.0007214	1317	0.001516	0.001224	2111
30	1201	0.0008213	0.0007262	960	0.001152	0.0007263	1390
7	2207	0.0004234	0.0004367	779	0.0006694	0.0005569	1084
		36 fractures IN					
120	744	0.001363	0.001596	1174	0.002536	0.00174	1749
30	1292	0.001344	0.001109	1118	0.00156	0.001239	1522
7	2474	0.0007618	0.0005185	1353	0.000947	0.0005337	1708
		36 fractures EN					
120	833	0.00139	0.001395	915	0.002395	0.001721	1396
30	1390	0.001169	0.001066	810	0.001628	0.001295	1096
7	2567	0.0009253	0.0006411	771	0.001015	0.0005989	934
		68 fractures EN					
120	1887	0.0006116	0.0004216	2238	0.0008863	0.0005681	4271
30	3179	0.0004667	0.0003912	1906	0.0006633	0.0003817	2501
7	5906	0.0002358	0.0002511	1605	0.0003713	0.0003195	2117
		120 fractures IN					
120	2676	0.000561	0.000557	4737	0.0006866	0.0004544	18177
30	4616	0.0003636	0.0002812	3075	0.0004421	0.0002824	7137
7	8793	0.0001875	0.0001517	4639	0.0002496	0.0001703	6075
		120 fractures EN					
120	3016	0.0004186	0.0003841	4042	0.0005124	0.0004198	12928
30	4993	0.0003235	0.0002657	3235	0.0004044	0.0003239	5917
7	9169	0.0001919	0.0001912	2892	0.0002522	0.0002195	3558

Table 5.3 reports the results obtained for all the configurations considered, with both the XFEM and standard FE for the description of the solution h. Results for the 27F

and 68F DFNs are shown using equally-spaced nodes on the traces, while the 36F and 120F DFNs are solved with both equally-spaced and induced nodes.

The quality of the results is evaluated in terms of the global continuity error and the global flux mismatch error relative to trace length, defined respectively in Section 4.5 as:

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

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

In Table 5.3 also the number of nodes for the control variable (column **Node**) and the number of iterations required to obtain stagnation of the functional at its minimum value (column **Iter**) are reported for each problem. The number of iterations should be interpreted as the *maximum* number of iterations for the problem and grid considered, since the use of a stopping criterion could considerably decrease the iterations required avoiding a large number of iterations close to functional minimum that do not substantially affect the quality of the solution, as discussed in the next Section.

Looking at the values In Table 5.3 concerning error indicators, we can see that the global continuity and flux mismatch errors are comparable between XFEM and FEM discretization, the former being in general slightly more accurate than the latter, while a considerably lower number of iterations is required with the XFEM based discretization to reach functional minimum.

Figures 5.13-5.14 display the convergence of the global continuity and flux mismatch errors relative to trace length against mesh size, indicated by the parameter δ representing the square root of grid maximum element area. Results are plotted for the DFN configurations 36F and 120F. Mesh refinement can reduce the global continuity and flux mismatch errors, and an higher trend is observed with induced nodes on the traces than with equally-spaced nodes. Since the number of nodes for the traces is similar for all grids for induced and equally-spaced nodes, the motivation of this difference is to be found in the disposition of nodes, and the induced disposition conforms better to the structure of the discrete solution h than the equally-spaced disposition. The reduction trend is in general comparable between XFEM and FEM. Superior performances of the XFEM on very coarse grids can also be noticed. This is expected, since the XFEM discretization relies on special enrichment functions to describe solution behaviour across

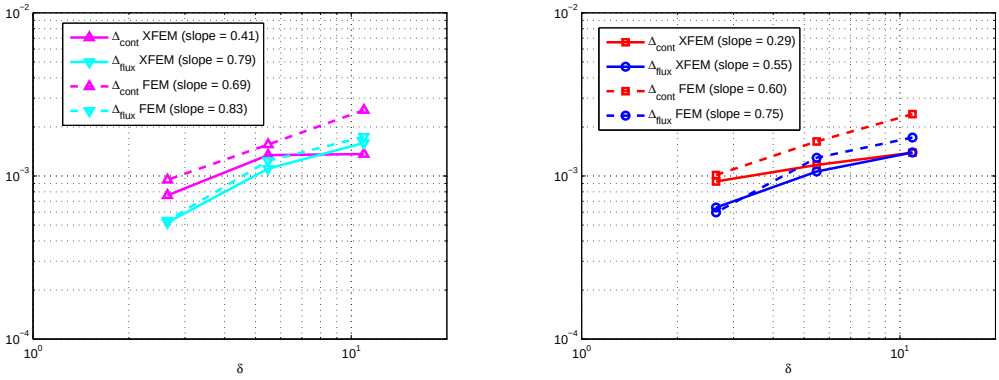


Figure 5.13: Continuity and flux mismatch errors relative to trace length against grid refinement for the DFN 36F. Induced (left) and equally-spaced (right) nodes.

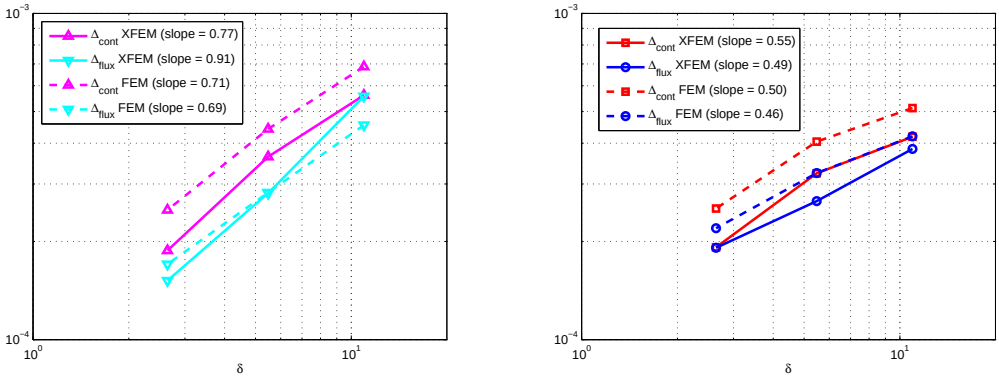


Figure 5.14: Continuity and flux mismatch errors relative to trace length against grid refinement for the DFN 120F. Induced (left) and equally-spaced (right) nodes.

the traces, and this reproduction capabilities are less affected from grid refinement than the nonconforming FEM discretization. In fact standard finite elements on nonconforming grids would correctly reproduce the jump in derivatives of the solution across the traces only on an infinitely refined grid.

The solution obtained for the 36F network with the XFEM discretization (Area=30m^2) and induced nodes is shown in Figure 5.15, where iso-h lines are also plotted to show the distortion of gradient across the traces. Figure 5.16 instead shows a source fracture view of the solution on the 120F DFN with FEM (Area=30m^2) and induced nodes on the traces, and Figure 5.17 reports a detail of the computational mesh, highlighting the non-conformity of mesh elements to fracture intersections.

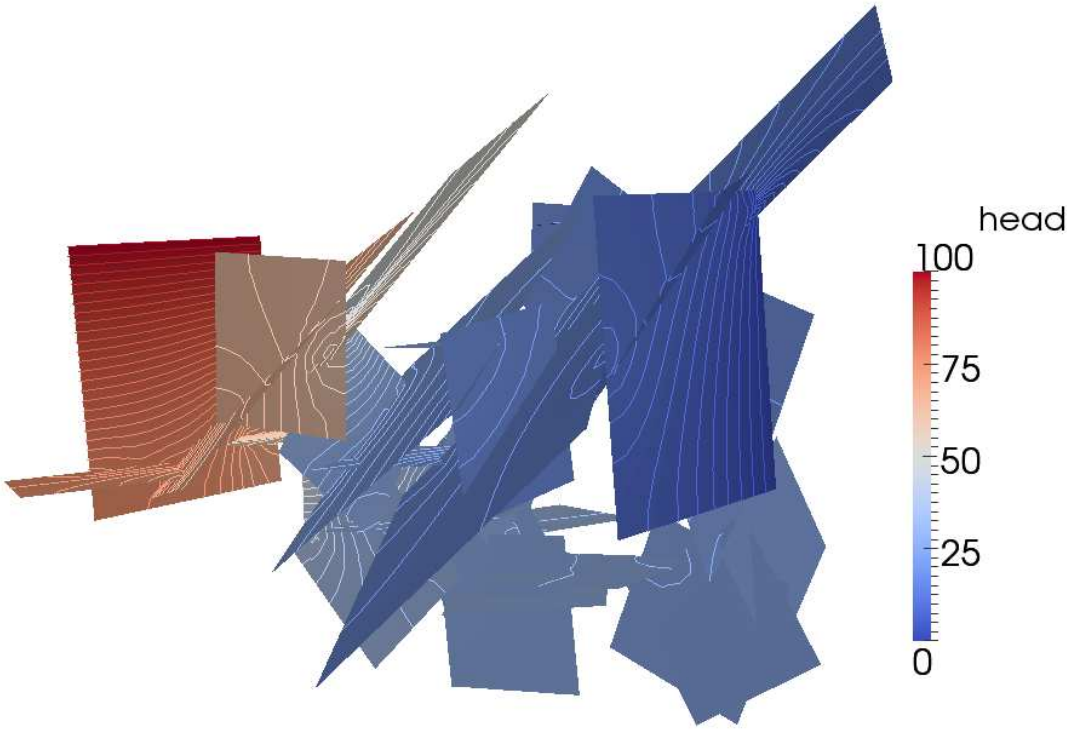


Figure 5.15: Solution for DFN 36F with the XFEM and induced nodes on the traces. Area=30.

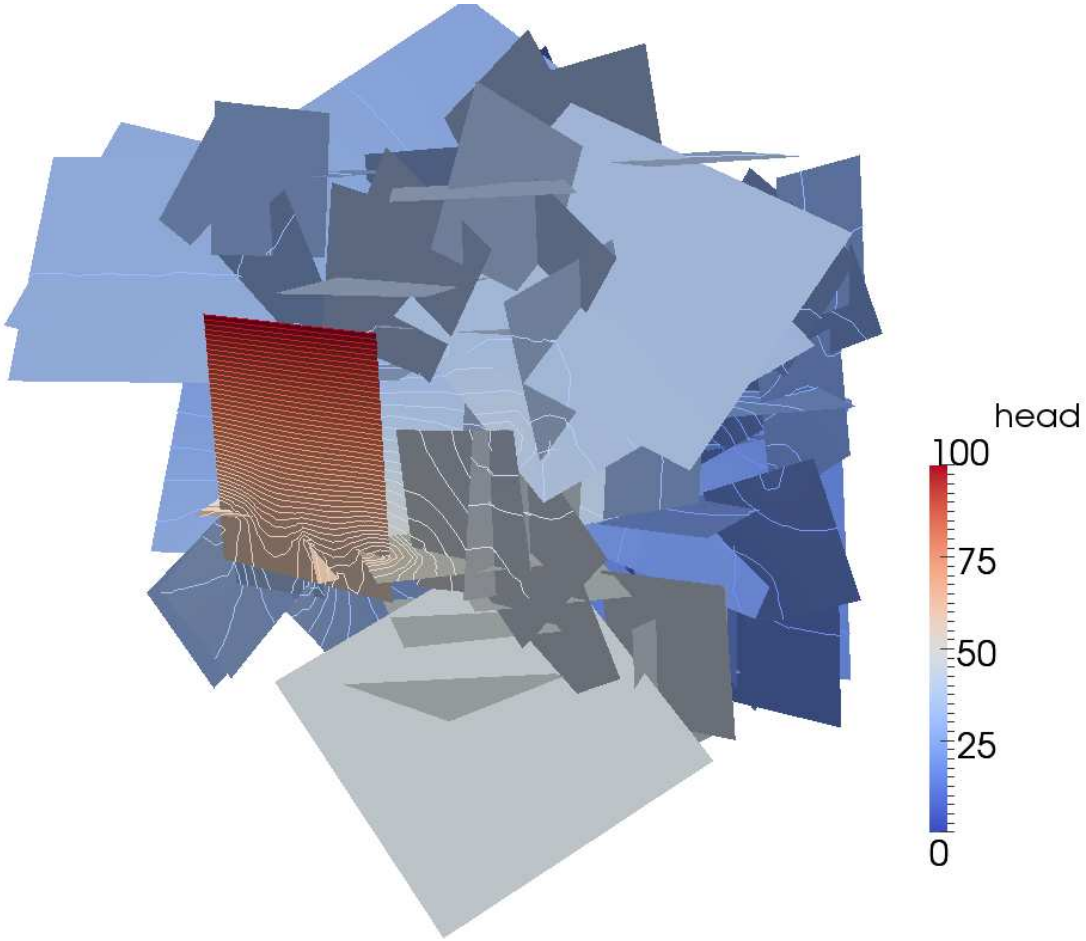


Figure 5.16: Solution for DFN 120F with the FEM and induced nodes on the traces. Area=30.

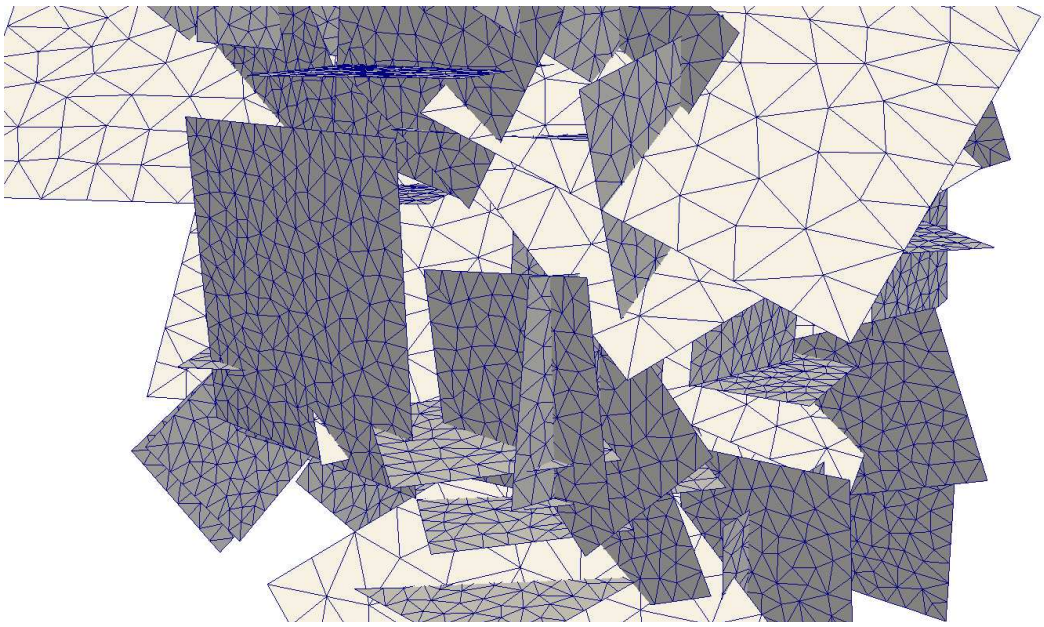


Figure 5.17: Detail of the computational grid with Area=30 for the DFN 120F.

As envisaged in Section 5.2 using equally-spaced nodes on the traces, increasing the number nodes for the control variables has the potential of reducing functional minimum for the same grid for the state variable h, with a corresponding reduction of the global continuity and flux mismatch errors. The results of this analysis are reported in Table 5.4 for the 27F, 36F, 68F and 120F DFN configurations with both XFEM and FEM based discretizations. It is possible to observe that increasing n_U both the global continuity and flux mismatch error are reduced with a small increase in the number of iterations required for functional stagnation.

When dealing with complex networks of fractures, another error indicator that can be considered to evaluate solution quality is the mismatch between the flux injected in the system by the source fracture and the total flux received from the network by the sink fracture. To this end a new error indicator is introduced, defined as:

$$\Delta_{\text{source-sink}} = \sum_{k \in F_\Gamma} \left(\sum_{m \in J_k} \int_{S_m} u_k^{S_m} - \alpha h_{k|_{S_m}} \right) / \sum_{k \in F_\Gamma} \sum_{m \in J_k} |S_m|$$

where F_Γ represents the set of fracture indexes carrying boundary conditions, and J_k collects the indexes to the traces on F_k. Numerical evidence shows that in order to control the source-sink flux mismatch it is beneficial to introduce penalty factors in

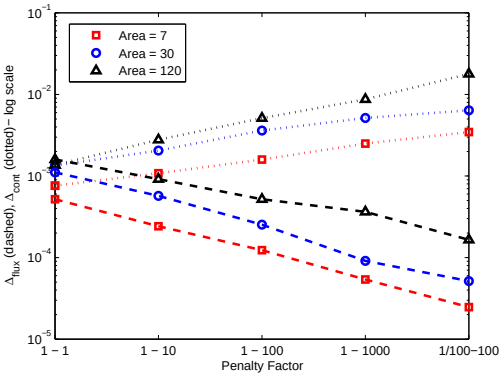


Figure 5.18: 36F DFN: Δ_{cont} and Δ_{flux} for different penalty factors with the XFEM. Induced nodes

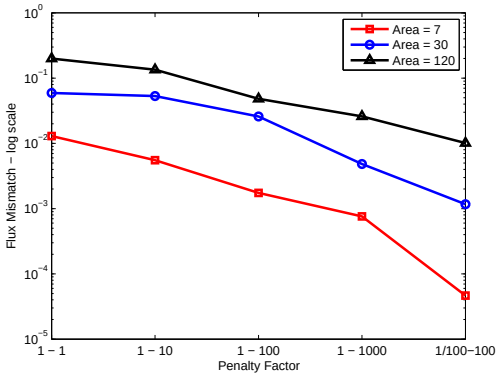


Figure 5.19: 36F DFN: $\Delta_{\text{source-sink}}$ for different penalty factors with the XFEM. Induced nodes

the definition of the functional, and differentiating the weight of continuity and flux mismatch. The cost functional rewrites as:

$$J(h,u) = \sum_{m=1}^{M} \left(Pf_1 \left\| h_{i|_{S_m}} - h_{j|_{S_m}} \right\|^2 + Pf_2 \left\| u_i^{S_m} + u_j^{S_m} - \alpha \left(h_{i|_{S_m}} + h_{j|_{S_m}} \right) \right\|^2 \right).$$

The results for various values of the penalty factors are reported in Table 5.5 for the 36F and 120F DFNs with both XFEM and FEM discretizations for h and induced nodes for the control variables, while Figures 5.18-5.23 report the plots of table data for the 36F DFN with XFEM and FEM discretization and for the 120F with standard FE. It can be noticed that increasing the weight of the flux term of the functional with respect to the continuity term has a strong effect in reducing both the flux mismatch error and the source-sink flux mismatch with a relatively small penalization of the continuity error. Since the continuity error remains in an acceptable range of values it appears that the use of a penalty on the flux term is advisable, mainly for complex DFN configurations, to improve solution quality. On the other hand, increasing Pf_2 and reducing Pf_1 causes a significant increase in the maximum number of iterations required for functional stagnation (columns **Iter** in Table 5.5), such that a trade-off between accuracy and computational effort is necessary.

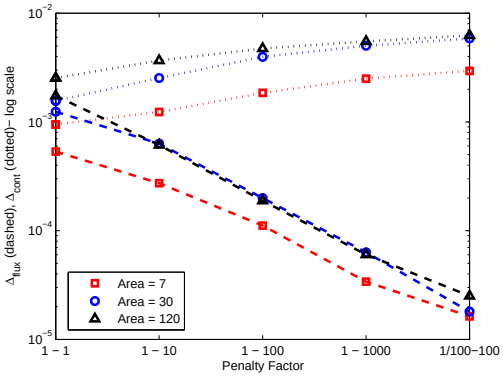


Figure 5.20: 36F DFN: Δ_{cont} and Δ_{flux} for different penalty factors with standard FE. Induced nodes

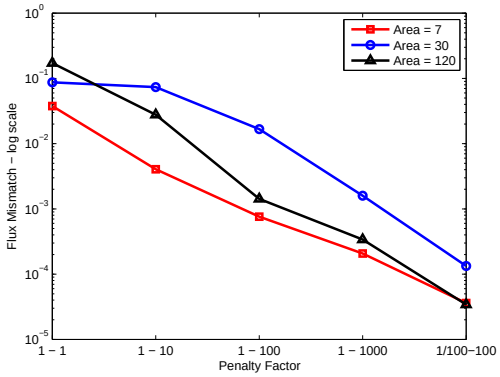


Figure 5.21: 36F DFN: $\Delta_{\text{source-sink}}$ for different penalty factors with standard FE. Induced nodes

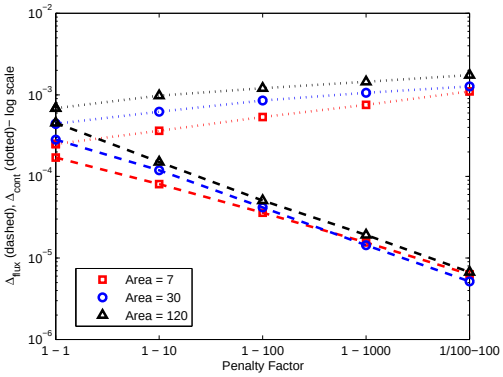


Figure 5.22: 120F DFN: Δ_{cont} and Δ_{flux} for different penalty factors with standard FE. Induced nodes

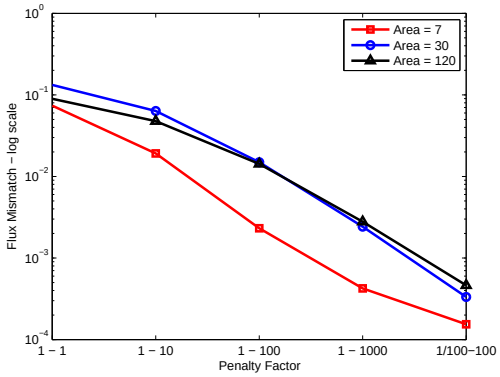


Figure 5.23: 120F DFN: $\Delta_{\text{source-sink}}$ for different penalty factors with standard FE. Induced nodes

Table 5.4: Effect of n_U on continuity and flux balance errors. XFEM and standard FEM compared, equally-spaced nodes

		XFEM			FEM		
Grid	$\mathbf{n_U}$	Δ_{cont}	Δ_{flux}	**Iter**	Δ_{cont}	Δ_{flux}	**Iter**
		27 fractures					
	1	0.0009953	0.0007214	1317	0.001516	0.001224	2111
120	1.5	0.0008252	0.0004977	1513	0.001201	0.0008163	2614
	2	0.0007432	0.000445	1432	0.001083	0.0006971	3218
	1	0.0008213	0.0007262	960	0.001152	0.0007263	1390
30	1.5	0.0005528	0.0005619	1190	0.0009182	0.0006567	1787
	2	0.0004097	0.0004008	1172	0.0007408	0.0005357	2024
	1	0.0004234	0.0004367	779	0.0006694	0.0005569	1084
7	1.5	0.0002907	0.0002599	908	0.0004729	0.0003669	1307
	2	0.000241	0.000198	1032	0.0004183	0.0002822	1463
		36 fractures					
	1	0.00139	0.001395	915	0.002395	0.001721	1396
120	1.5	0.001059	0.00109	1067	0.002033	0.001397	1691
	2	0.0008266	0.0009029	1103	0.001715	0.001297	1775
	1	0.001169	0.001066	810	0.001628	0.001295	1096
30	1.5	0.0008244	0.0007246	921	0.001244	0.0009339	1349
	2	0.0006507	0.0005739	976	0.00109	0.0007246	1499
	1	0.0009253	0.0006411	771	0.001015	0.0005989	934
7	1.5	0.0006953	0.0005292	916	0.0008782	0.000456	1161
	2	0.0005169	0.0004915	1049	0.0008088	0.0003994	1315
		68 fractures					
	1	0.0006116	0.0004216	2238	0.0008863	0.0005681	4271
120	1.5	0.0004791	0.0003322	2536	0.0007222	0.0004366	5737
	2	0.0004361	0.0002868	2650	0.0006526	0.0003947	4859
	1	0.0004667	0.0003912	1906	0.0006633	0.0003817	2501
30	1.5	0.0003171	0.0002996	2100	0.0005367	0.0003535	3113
	2	0.0002329	0.0002247	2130	0.0004263	0.0003194	3776
	1	0.0002358	0.0002511	1605	0.0003713	0.0003195	2117
7	1.5	0.0001519	0.0001623	1713	0.0002551	0.0002304	2408
	2	0.0001204	0.0001199	1893	0.0002071	0.0001832	2679
		120 fractures					
	1	0.0004186	0.0003841	4042	0.0005124	0.0004198	12928
120	1.5	0.0003191	0.0002713	4125	0.0004246	0.0003007	9470
	2	0.000274	0.0002298	4132	0.0003856	0.0002686	10520
	1	0.0003235	0.0002657	3235	0.0004044	0.0003239	5917
30	1.5	0.0002589	0.0001893	3521	0.0003091	0.0002383	7017
	2	0.000225	0.0001684	3761	0.0002771	0.0002026	6995
	1	0.0001919	0.0001912	2892	0.0002522	0.0002195	3558
7	1.5	0.0001509	0.0001323	3150	0.0002068	0.0001578	4043
	2	0.0001287	0.0001054	3329	0.000183	0.0001282	4629

Table 5.5: Effect of penalty factors on Δ_{cont}, Δ_{flux} and $\Delta_{\text{source-sink}}$. DFNs 36F and 120F with induced nodes. XFEM and standard FEM compared.

		XFEM				FEM			
Grid	$\mathbf{Pf_1 - Pf_2}$	Δ_{cont}	Δ_{flux}	$\Delta_{\text{source-sink}}$	**Iter**	Δ_{cont}	Δ_{flux}	$\Delta_{\text{source-sink}}$	**Iter**
		36 fractures							
120	1 - 10	0.002782	0.0009164	-0.1347	969	0.003668	0.0006137	-0.028	1787
	1 - 100	0.005125	0.0005185	-0.0477	1539	0.004736	0.0001887	0.0014	3182
	1 - 1000	0.008723	0.0003634	-0.026	2409	0.005499	6.032e-05	3.367e-4	5870
	1/100 - 100	0.01785	0.000165	-0.0102	2943	0.006258	2.507e-05	-3.425e-5	7690
30	1 - 10	0.00205	0.0005705	-0.0531	1146	0.002536	0.0006313	-0.0735	2056
	1 - 100	0.003612	0.0002532	-0.0258	1536	0.003966	0.0001995	-0.0167	2914
	1 - 1000	0.005158	9.107e-05	-0.0048	3062	0.005023	6.316e-05	-0.0016	5094
	1/100 - 100	0.00639	5.143e-05	-0.0011	4889	0.005853	1.8e-05	-1.335e-4	9742
7	1 - 10	0.001084	0.0002419	-0.0055	1394	0.001234	0.0002728	-0.004	1991
	1 - 100	0.001593	0.0001234	-0.0017	1741	0.00185	0.0001114	-7.605e-4	3075
	1 - 1000	0.0025	5.381e-05	-7.62e-4	2721	0.002501	3.397e-05	-2.074e-4	4360
	1/100 - 100	0.003462	2.471e-05	-4.657e-5	5326	0.002942	1.626e-05	-3.602e-5	9406
		120 fractures							
120	1 - 10	0.001004	0.0002823	-0.1088	4310	0.0009781	0.0001503	-0.1295	17601
	1 - 100	0.001487	0.0001964	-0.0789	7658	0.001208	5.068e-05	-0.0142	37118
	1 - 1000	0.002703	0.0001445	-0.0464	11457	0.001448	1.917e-05	-0.0028	30585
	1/100 - 100	0.006056	0.0001023	-0.0242	10941	0.001749	6.727e-06	-4.627e-4	31170
30	1 - 10	0.0005485	0.0001584	-0.0958	3625	0.0006226	0.0001192	-0.0635	11663
	1 - 100	0.0008711	0.0001008	-0.0211	3562	0.0008539	4.175e-05	-0.0149	13285
	1 - 1000	0.001414	8.061e-05	-0.0037	4327	0.001061	1.436e-05	-0.0024	21996
	1/100 - 100	0.003473	6.136e-05	-0.0026	5575	0.001269	5.167e-06	-3.33e-4	22205
7	1 - 10	0.0002967	7.43e-05	-0.0228	4404	0.0003628	8.06e-05	-0.0191	7481
	1 - 100	0.0004575	3.686e-05	-0.003	5791	0.0005353	3.589e-05	-0.0023	9666
	1 - 1000	0.0007086	1.936e-05	-6.224e-4	8804	0.0007546	1.55e-05	-4.237e-4	18468
	1/100 - 100	0.00111	1.147e-05	-2.449e-4	16228	0.001103	6.314e-06	-1.539e-4	25349

5.4 Variable fracture transmissivity

In this section DFN configurations with values of fracture transmissivity constant on each fracture but different from a fracture to another are considered. The DFN analysed are reported in Table 5.6 along with the range of fracture transmissivity allowed for the various configurations. The XFEM and induced nodes are used throughout this Section.

When dealing with large variations of fracture trasmissivities, occurring possibly between intersecting fractures, a possible choice for the penalty factors introduced in the previous Section is to set $Pf_1 = 1$ and $Pf_2 = 1/\mathbf{K}_{min} = 1/\min_i(\mathbf{K}_i)$. As shown in the sequel, this improves the numerical behaviour of the method for complex DFN configurations, since it magnifies the influence of the control variable u on the solution. As usual we set $\alpha = 1$, and all simulations are started with an initial guess for the control variable $u^0 = 0$. Simulations are performed on three different grids characterized by maximum element area of 7m^2, 15m^2 and 30m^2.
In Figure 5.24 the coarse grid for problem 11F is shown. It should be noticed that elements are arbitrarily placed with respect to the traces, and the mesh on each fracture is independent from the mesh on the other fractures. The solution is shown in Figure 5.25 along with iso-h lines, in order to highlight that, as expected, the highest gradients in the solution occur in fractures with the lower values of fracture transmissivity, which can be noticed looking at Figure 5.26 where the values of $\mathbf{K}$ on the fractures of the system are reported. Figures 5.27-5.31 refer to the 68F system on the intermediate grid. In addition to previous considerations, looking at iso-h lines in Figure 5.27 we can see that the flux tends to stagnate in fractures that are a dead end or that are connected to the system by fractures with low transmissivity values. This is again an expected behaviour. Figure 5.28 shows the distribution of $\mathbf{K}$ for this system, while Figure 5.29

Table 5.6: DFN fracture transmissivity

Label	K_{min}	K_{max}
11F	2.46×10^{-3}	9.66×10^{-2}
27F	5.43×10^{-4}	9.66×10^{-2}
37F	5.43×10^{-4}	9.66×10^{-2}
55F	5.43×10^{-4}	9.67×10^{-2}
68F	5.43×10^{-4}	9.67×10^{-2}

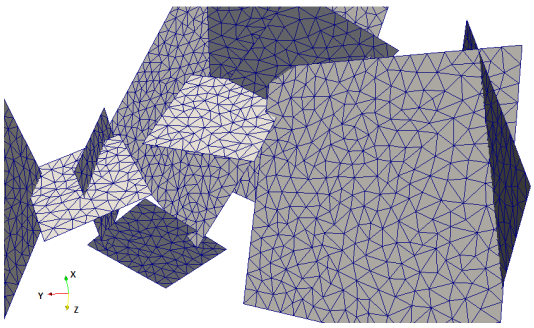


Figure 5.24: Problem 11F: coarse grid

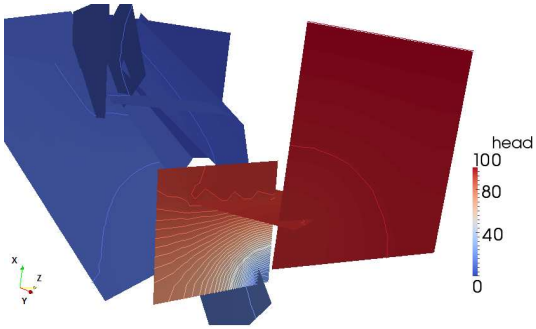


Figure 5.25: Problem 11F: solution with iso-h lines on the coarse grid

provides a detail of the intermediate grid. The Figure shows that complex geometries and intricate fracture intersections can be easily handled with no requirement for mesh adjustments and without compromising the description of the numerical solution, as it can be seen looking at Figure 5.30-5.31 where the solution on selected fractures are plotted not on the computational grid but on sub-triangles not crossing the traces, for graphical reasons. The irregular behaviour across traces and around trace tips is well defined, regardless of the reciprocal position of traces and mesh elements.
In Table 5.7 the fluxes entering the system through the traces of the source fracture (column *in*), the fluxes leaving the system from the sink fracture (column *out*) and the mismatch between these two quantities (column *diff*) are reported for each system and grid considered. We can observe that flux conservation is very good and is stable under grid refinements for each problem. Moreover flux mismatch remains stable also for increasing problem complexity.

The proposed approach can easily deal with non-uniform transmissivities on each fracture plane, requiring either a different implementation of the integrals for the discrete operators on the fractures either the approximation of the fracture transmissivity function on each fracture with a piecewise constant function on each element of the mesh. We remark that the latter approach would not affect the accuracy of the method. A deeper investigation with this kind of configurations will be the objective of future analysis.

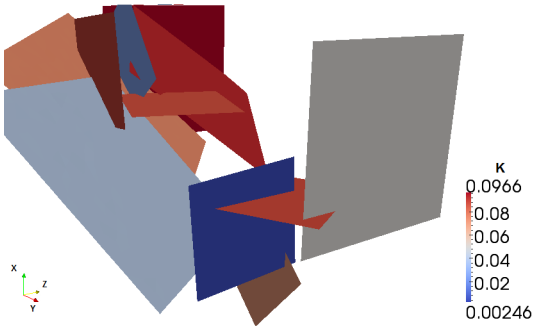


Figure 5.26: Problem 11F: fracture transmissivity $\mathbf{K}$ distribution

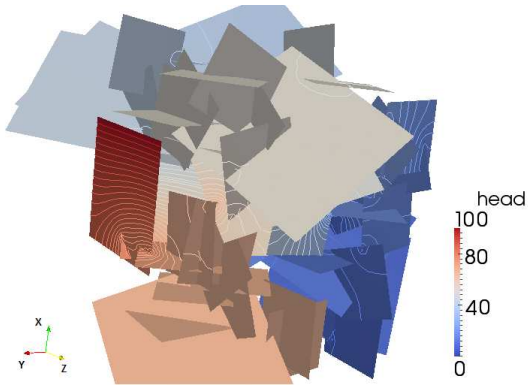


Figure 5.27: Problem 68F: solution with iso-h lines on the intermediate grid

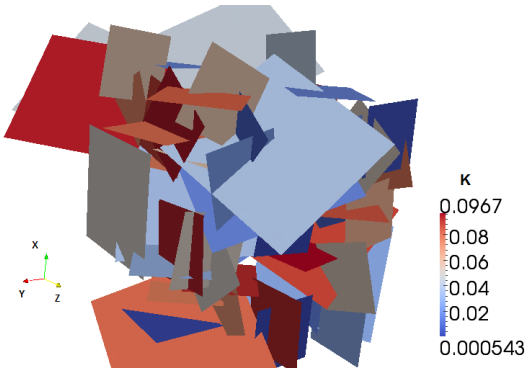


Figure 5.28: Problem 68F: fracture transmissivity $\mathbf{K}$ distribution

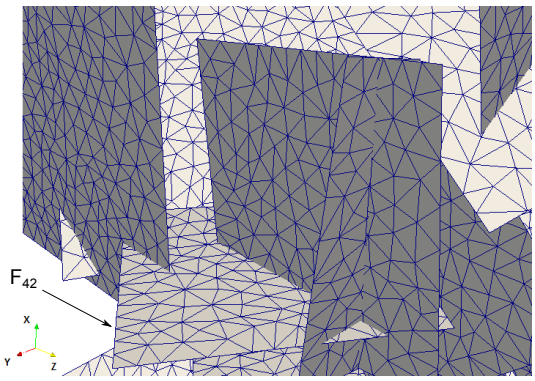


Figure 5.29: Problem 68F: detail of the intermediate grid

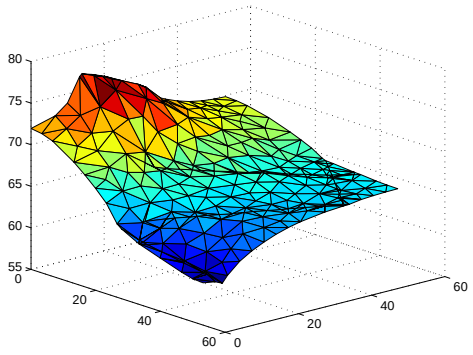


Figure 5.30: Problem 68F: Solution on fracture F_{42}. Intermediate grid.

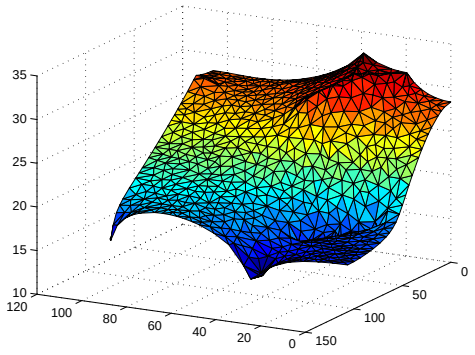


Figure 5.31: Problem 68F: solution on a selected fracture. Intermediate grid

Table 5.7: Flux unbalance for approximate solution

	grid 30			grid 15			grid 7		
DFN	in	out	diff	in	out	diff	in	out	diff
11F	0.16	-0.16	4.4e-4	0.16	-0.16	1e-5	0.16	-0.16	2e-5
27F	0.42	-0.42	2.9e-4	4.19	-4.19	1.4e-4	0.42	-0.42	2e-5
37F	1.10	-1.10	1.2e-4	1.09	-1.09	2.0e-4	1.08	-1.08	8e-5
55F	1.45	-1.45	6.3e-4	1.44	-1.44	3.3e-4	1.43	-1.43	9e-5
68F	1.12	-1.12	9.4e-4	1.11	-1.11	3.5e-4	1.10	-1.10	1e-5

5.4.1 Convergence study

Let us introduce for each problem and grid a reference solution h_{ref}, corresponding to the stagnation of the functional J around its minimum. For an approximate solution obtained at a given number of iterations, h_{curr} we define a relative distance from the reference solution as the H^1-norm of the difference between current approximation and reference solution divided by the H^1-norm of the reference solution: $\|h_{curr} - h_{ref}\|_{H^1}/\|h_{ref}\|_{H^1}$. As a reasonable choice we measure the complexity of each problem with the number of traces in the system. In Figure 5.32 the relative distance of solution at various number of iterations against the ratio of iteration and number of traces is displayed for the 27F, 37F and 68F DFNs on the coarse and fine grid. A similar plot is in Figure 5.33 for all the problems considered on the intermediate grid. In both Figures the global trend is plotted on the left side, showing that the curves are well clustered and show an initial steep descent path, after which the slope reduces. On the right there is a zoom at low values of iterations over the number of traces. After a small number of iterations compared to the number of traces, the current approximation is close to the reference solution, with variations lower than 1%. In the simulations performed this occurs typically in a range of iterations between two and four times the number of traces, independently of the problem and grid considered. A similar behaviour is also documented in Chapter 3, showing that the algorithm can provide a good solution with a cost that increases linearly with problem complexity.

We end the presentation of numerical results providing some stopping criteria for the discrete algorithm. Two possible criteria are discussed here and, summarized in Table 5.8: 1) algorithm stops when the difference between subsequent iterations is small,

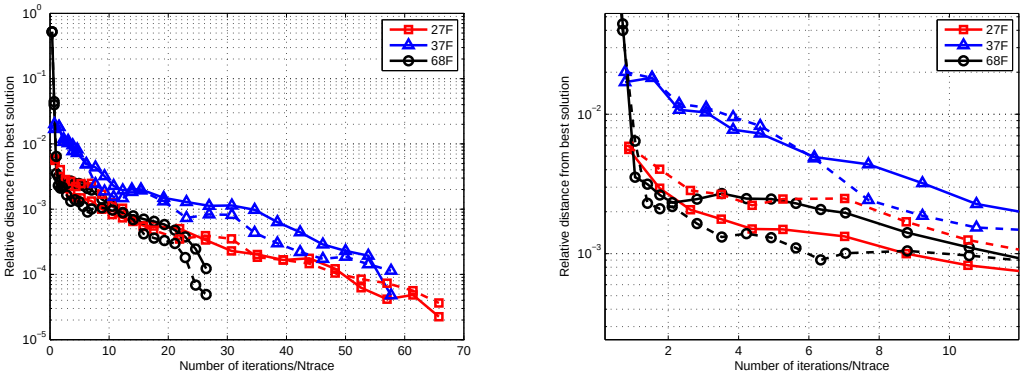


Figure 5.32: Relative distance in H^1 norm of solution at various number of iterations for selected problems. Coarse grid in dashed lines, fine grid in solid lines. Full picture on the left, zoom on the right.

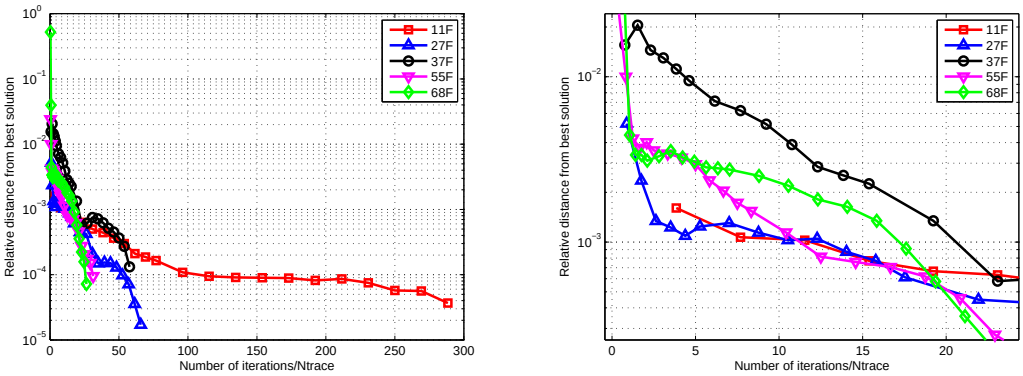


Figure 5.33: Relative distance in H^1 norm of solution at various number of iterations for selected problems on the intermediate grid. Full picture on the left, zoom on the right.

Table 5.8: Exit criteria used in simulations

Label	Criterion
t_1	$\mathcal{R}_1 = J^k - J^{k-1} < \text{Tol}_1$
t_2	$\mathcal{R}_3 = J^k(J^k - J^{k-1}) < \text{Tol}_2$

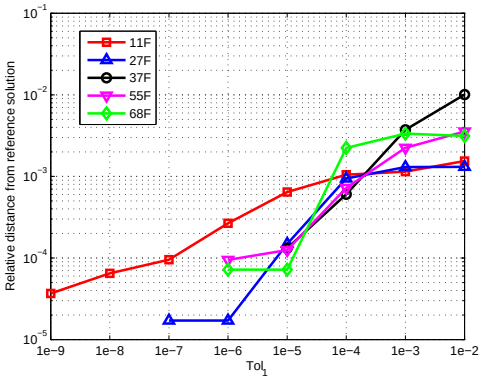


Figure 5.34: Distance of solution from reference solution versus different values of Tol_1. Intermediate grid.

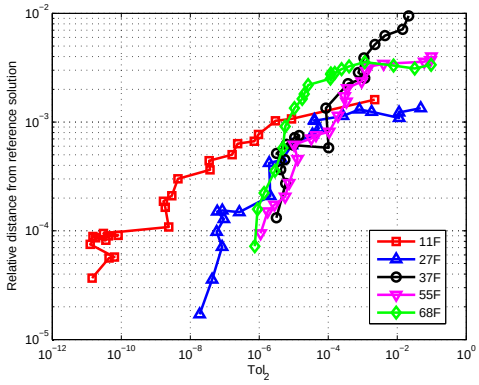


Figure 5.35: Distance of solution from reference solution versus different values of Tol_2. Intermediate grid.

i.e. $\mathcal{R}_1 < \text{Tol}_1$ six subsequent times; 2) stop occurs when th difference between subsequent iterations scaled with functional value is small, i.e. $\mathcal{R}_2 < \text{Tol}_2$. Both conditions seek functional stagnation, differing in the fact that condition t_2 also takes into account functional absolute value, see also Chapter 3. The criterion on u introduced in Chapter 3 has been removed, since we actually solve the rescaled problem in which the effects of the control variable u are amplified. The behaviour of the suggested criteria is shown in Figures 5.34-5.35. A value of 10^{-3} appears to be a suitable choice for both criteria for all the problems.

5.5 Scalability

We end this Chapter with a preliminary analysis on the scalability performances of the proposed approach for discrete fracture network simulations on large scales. As mentioned the method allows an independent meshing process on each fracture of the

network, and the resolution of the constrained optimality problem with a gradient based method can be performed in parallel with a very limited exchange of data related to the traces.

The implementation of the parallel version of the method is performed using an MPI package for *Octave*, [3], called *openMPI_ ext*, [4], that provides a subset of the standard MPI library for the C programming language.

Since dealing with an implementation oriented to computer architectures with non-shared memory, the parallel version of the algorithm is structured in order to limit the amount of communications. To this end a hierarchical organization is envisaged, with *Master* processes managing groups of *Slave* processes. The DFN is subdivided into smaller subsets of fractures, each managed by one *Slave* process. The *Slave* processes refer to a *Master* process for the communication phase, such that all the information shared by the *Slave* processes transit through the *Master* . For very large DFN configurations this basic structure can be repeated, introducing a hierarchy in *Master* processes with higher level *Masters* managing groups of lower level *Master* processes, down to the *Slave* processes managing groups of fractures. We remark that this configuration would not be optimal for shared memory computer architectures, such as GPU based machines, where a more efficient implementation would consist in assigning each fracture to a different process. Investigation of the parallel approach on shared memory architectures is postponed to a future work.

5.5.1 Partitioning the DFN

The first task that the parallel implementation of the proposed method has to accomplish consists in determining the subsets of fractures that will be associated to each *Slave* process. To this end, the DFN can be suitably represented by a non-directed graph $G(V, E)$, with fractures representing the vertices V of the graph and traces the edges, E. The objective is to minimize the number of edge cuts, i.e. the amount of communication between processes, balancing the workload among processes at the same time. Let us assume that k represents the number of *Slave* processes, and I is the total number of vertices (i.e. fractures) in the graph, than we want to determine a subdivision of $G(V, E)$ such that the weight (i.e. the computational cost) of each part is lower than $\nu\frac{I}{k}$, where ν is a parameter close to one, and the capacity (i.e. the amount of data shared) of edge cuts is minimized. This problem is well known in graph theory

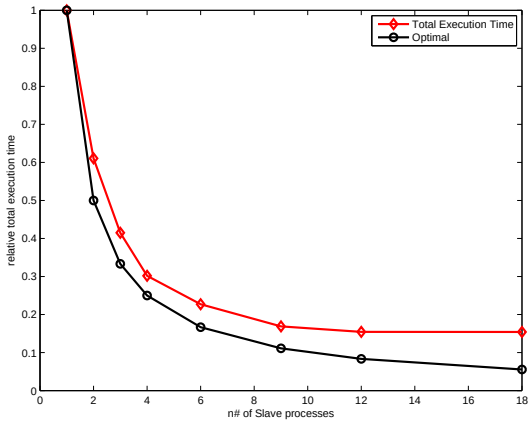


Figure 5.36: Scalability analysis for the DFN 36F

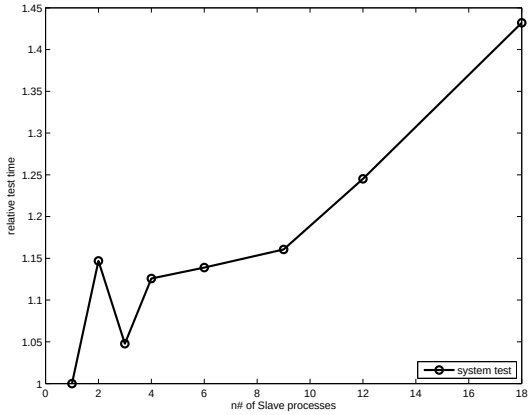


Figure 5.37: Process independence test

as (k, ν)-*balanced graph partitioning*, see e.g. [1] . For this preliminary investigation a simple unweighted graph partitioning procedure is implemented, ensuring that all the vertices of the graphs, (i.e. the fractures of the DFN) have a similar computational cost. This condition is quite restrictive, but is appropriate for the current preliminary investigation, since it reduces the complexity of the graph partitioning procedure and can be easily achieved by prescribing a similar number of degrees of freedom on all the fractures in the DFN.

5.5.2 The message passing process implementation

As mentioned, all the information are shared by *Slave* processes through *Master* processes and each *Slave* process sends to and receives from the *Master* process only the portion of data related to those traces in common with other processes. Since the DFN is partitioned in a way that minimizes the number of traces shared by different processes, the communication phase is minimized. In any case only arrays of small size compared to the size of the problems on the fractures need to be shared. The *openMPI_ext* package does not allow for non-blocking communication routines and this is a severe limitation for this application. Indeed non-blocking send routines would allow to partially hide the overhead for communications, allowing each process to send the information required by other processes and continue computing on other fractures whose data is not required by other processes. This kind of limitations will be removed in future implementations of the method based on the C language.

5.5.3 Scalability results

We now show the scalability results obtained on the DFN 36F, using the XFEM for the discretization of the hydraulic head on the fractures and induced nodes for the control variables. The mesh parameter is different on each fracture in order to obtain a number of DOFs similar for all the fracture around 3500 DOFs, thus allowing for an unweighted graph partitioning for determining the workload for the *Slave* processes.

Simulations are performed on a computer with two six-core processors, for a total of twelve physical cores and twenty-four virtual cores. The machine has a shared memory architecture but is treated as a non-shared memory machine.

The scalability results for the 36F DFN are shown in Figure 5.36 in terms of execution time relative to the execution time in serial mode. It can be noted that scalability performances are good and quite close to the ideal ones when using up to 9 *Slave* processes. When using more than 10 *Slave* processes the slope of the curve reduces with respect to the ideal one, and there is no further reduction of execution time using more than 12 *Slave* processes. This is partly due to the overhead in communication and partly to the bottleneck of memory access due to the architecture of the computer used. The ideal curve considers that none part of the algorithm is strictly serial.

An analysis is performed to measure the level of independence among the virtual cores and to highlight conflicts in memory access observed during the simulations. A large size (6400×6400) sparse linear system with about 3×10^4 non zero elements is solved 10 times in serial mode by an increasing number of processes running in parallel, such that each process performs exactly the same operations and no communication occurs. The average execution time across the 10 repeated resolutions, t_{10}^j, is stored for each process j. The mean value $t_{av}^k = k^{-1} \sum_{j=1}^{k} t_{10}^j$ among the k different processes, relative to execution time with a single process, is reported in Figure 5.37 for different values of k (number of processes) ranging from 1 to 19. It is possible to note that, even with a small number of processes running in parallel, the execution time increases between $5 - 15\%$, due to conflicts in accessing the memory. When using more than ten processes the degradation of performances becomes severe.

Concluding, this preliminary investigation on the scalability performances of the proposed algorithm for DFN simulations shows a very high potential, despite the limitation of the MPI library used and of the parallel computer available. Implementation improvements can have the potential of further reducing the gap with the ideal scalabil-

ity performances and of extending the scalability range to a higher number of parallel processes.

Bibliography

[1] K. Andreev and H. Räcke, *Balanced graph partitioning*, in Gibbons and Adler [2], pp. 120–124.

[2] P. B. Gibbons and M. Adler, eds., *SPAA 2004: Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures, June 27-30, 2004, Barcelona, Spain*, ACM, 2004.

[3] *GNU Octave*. `http://www.gnu.org/software/octave/`, February 2014.

[4] *octave-openmpi-ext1.1.0-1 package*. `http://octave.sourceforge.net/mpi/`, February 2014.

Chapter 6

Preconditioning

In this Chapter a preliminary investigation of a possible strategy for preconditioning the conjugate gradient method for DFN problems with the proposed approach is investigated, aiming at a reduction in the overall computational cost.

Let us recall the optimization algorithm described in Section 4.4:

Conjugate gradient method

1. Choose an initial guess u^0
2. Compute h_0 and p_0 solving (7.23) and (7.24) and g_0 by (7.25)
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. $k = k + 1$

where g_k indicates the gradient $\nabla \hat{J}(u_k)$ at step k and d_k the direction of movement. Let us denote by d_k^{ex} the descent direction at iteration k that vanishes the residual g_k, i.e. $g_k + \hat{G} d_k^{ex} = 0$. The idea for preconditioning consists in evaluating an approximation

d_k^p of d_k^{ex} to be used in place of the conjugate gradient direction. In order to make the computation of direction d_k^p affordable, it can be determined on a coarser grid with respect to the computational grid. In this respect this kind of preconditioning borrows the structure of multi-grid (MG) preconditioners, but relying on a different concept. In fact MG preconditioning is based on a correlation between error frequencies and computational grid typical of problems with an elliptic structure with solvers that have certain smoothing properties, [1]. Even if on each fracture of a DFN elliptic problems are solved, $\hat{G}$ does not share the spectral properties expected for efficient application of MG preconditioning. As a consequence, the speed up will not be given by the reduction of lower error frequencies on the coarser grids as in multi-grid preconditioners, but rather it depends on the quality of direction d_k^p in approximating d_k^{ex}.

Let us introduce a two grid framework with a finer computational grid for the resolution of the DFN problem and a coarser grid for preconditioning purposes, and let us denote by $\mathcal{U}_{\delta,f}$ and $\mathcal{U}_{\delta,c}$ the discrete spaces for the control variable on the fine and coarse grid respectively. Let then $\hat{G}_f$ be the matrix $\hat{G}$ on the fine grid and $\hat{G}_c$ the corresponding matrix on the coarse grid. We define a prolongation operator $I_c^f : \mathcal{U}_{\delta,c} \to \mathcal{U}_{\delta,f}$ and a restriction operator $I_f^c : \mathcal{U}_{\delta,f} \to \mathcal{U}_{\delta,c}$, such that $\left(I_f^c u, v\right)_{\mathcal{U}_{\delta,c}} = \left(u, I_c^f v\right)_{\mathcal{U}_{\delta,f}}$, for all $u \in \mathcal{U}_{\delta,f}$ and for all $v \in \mathcal{U}_{\delta,c}$, (see [1]). Given the gradient direction $g_{k,f}$ at iteration k on the fine grid provided by the conjugate gradient algorithm, the preconditioned descent direction can be written as:

$$d_k^p = I_c^f(d_{k,c}), \qquad \hat{G}_c d_{k,c} = -\left(I_f^c(g_{k,f})\right) \tag{6.1}$$

The resolution of the linear system in (6.1) for $d_{k,c}$ on the coarse grid does not necessarily require to form matrix $\hat{G}_c$. It is possible to rewrite it as a minimization problem on the coarse grid as follows:

$$\begin{aligned} &\hat{G}_c d_{k,c} + \left(I_f^c(g_{k,f})\right) = \hat{G}_c\left(u_c - I_f^c(u_{k,f})\right) + \left(I_f^c(g_{k,f})\right) = 0 \\ &\Leftrightarrow \min_{u_c} u_c^T \hat{G}_c u_c + \left(I_f^c(g_{k,f}) - \hat{G}_c I_f^c(u_{k,f})\right)^T u_c \end{aligned}$$

thus having the same structure of the problem on the finer grid, and, as such, solved with an optimization (iterative) method. As in multi-grid preconditioning, more levels with successive coarsening of the grids could be used and, depending on the size of the coarsest grid, the computational cost for assembling $\hat{G}_c$ could be comparable or even less than that of solving (6.1) as a minimization problem. We remark that matrix $\hat{G}_c$ can be assembled working independently on each fracture of the DFN.

As in a multi-grid scheme the correction on the coarse grid can be evaluated after a given number of iterations of conjugate gradient, according to the value of a parameter n_{CG}. The preconditioned algorithm can be then written as:

Algorithm 6.1. Preconditioned conjugate gradient method

1. Choose an initial guess u^0
2. Compute h_0 and p_0 solving (7.23) and (7.24) and g_0 by (7.25)
3. Set $d_0 = 0$, $\beta_0 = 0$, $k = 0$, $k_{CG} = 0$
4. While $g_k \neq 0$
5. if $k_{CG} < n_{CG}$ (Conjugate Gradient scheme)
   - 5.1. Compute $d_k = -g_k + \beta_k d_k$
   - 5.2. Compute λ_k with a line search along d_k
   - 5.3. Compute $u_{k+1} = u_k + \lambda_k d_k$
   - 5.4. Update $g_{k+1} = g_k + \lambda_k \hat{G} d_k$
   - 5.5. Compute $\beta_{k+1} = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$
   - 5.6. $k = k + 1$, $k_{CG} = k_{CG} + 1$
6. else (Preconditioned scheme)
   - 6.1. Compute d_k^p according to (6.1)
   - 6.2. Compute $u_{k+1} = u_k + d_k^p$
   - 6.3. Update $g_{k+1} = g_k + \hat{G} d_k^p$
   - 6.4. $k = k + 1$, $k_{CG} = 0$
7. end (if)

Some numerical results on this preconditioning technique are now discussed. All the simulations are performed solving system (6.1) exactly on the coarse grid. The XFEM is used for the discretization of the solution h on the fractures, while a node strategy ED is chosen for the control variables. The fine grid has maximum elements area of 7m^2, while the coarse grid of 30m^2. The ED discretization for the control variables on the

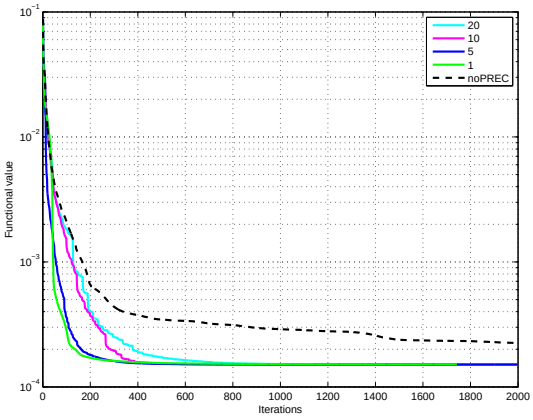


Figure 6.1: Functional value vs iterations for different values of n_{CG}

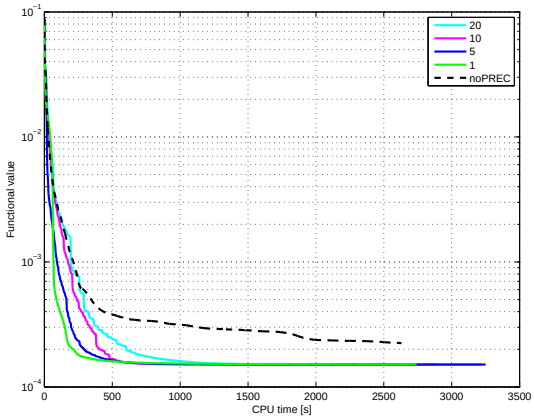


Figure 6.2: Functional value vs cpu time for different values of n_{CG}

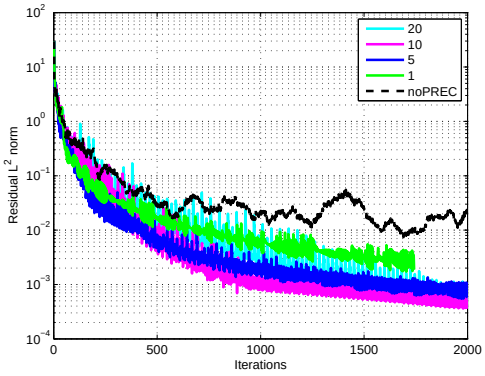


Figure 6.3: System residual value vs iterations for different values of n_{CG}

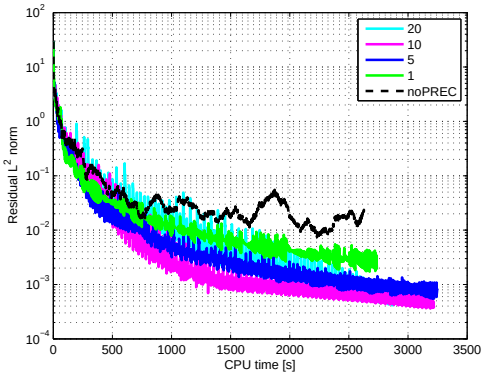


Figure 6.4: System residual value vs cpu time for different values of n_{CG}

fine and coarse grids are nested, in order to ease the generation of the restriction and prolongation operators. Numerical result are shown for the DFN 120F. A maximum number of 2000 iterations is prescribed for all the simulations. The results relative to simpler DFN configurations, in fact, are not significant for the analysis performed here, since the computational cost of a single iteration of the non preconditioned scheme (in serial) might be significantly more expensive that the evaluation of the direction d_k^p, while this is not the case for more complex configurations, where the use of preconditioning is of interest.

The quality of the solution is evaluated in terms of functional final value and of the L^2-norm of the residual g_k at iteration exit. In both cases lower values are desirable.

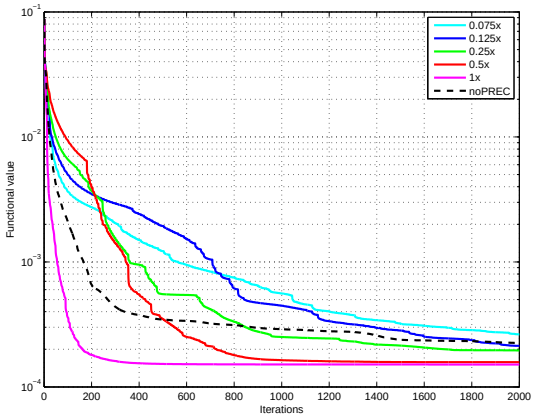


Figure 6.5: Functional value vs iterations for different values of n_U

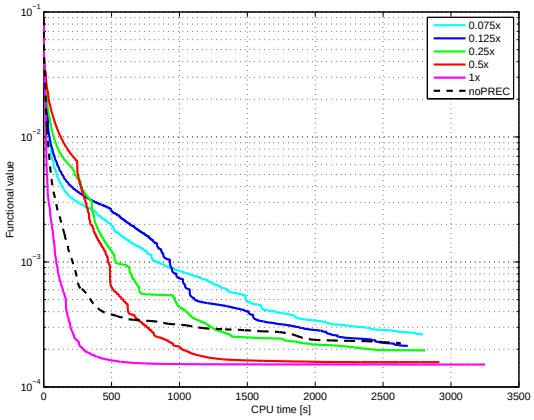


Figure 6.6: Functional value vs cpu time for different values of node factor n_U

Looking at Table 6.1, Panel A, it is possible to notice that the use preconditioning allows to reach a much lower residual than the non-preconditioned case in the same number of iterations. The computational cost in terms of cpu time required to perform the maximum number of iterations allowed is higher for the preconditioned case, but observing Figures 6.1-6.4 we can see that at the same time the preconditioned scheme reaches a better solution in terms of residual norm. The minimum for the residual is obtained for a value of $n_{CG} = 10$. The results of Table 6.1, Panel B show the performances of the preconditioner when the number of nodes for the control variables are reduced on the fine and coarse grid of the same factor n_U. Decreasing the number of nodes leads to a reduction of the computational cost in terms of cpu time, but also the benefits of preconditioning vanish, and if $n_U < 0.25$ there is no advantage in the use of preconditioning, as can be noticed observing Figures 6.5-6.8.

Figures 6.9-6.12 show the effectiveness of preconditioning when the maximum area of the coarse grid elements is increased, imposing the same number and disposition of nodes for the control variables on the fine and coarse grids (i.e. $I_c^f = I_f^c = I$, identity matrix). The results obtained highlight that increasing coarse grid area is not a viable option to reduce the cost of this preconditioning technique.

Concluding, the presented preconditioning technique has a good potential in reducing the computational cost of the optimization algorithm, both in terms of number of iterations and cpu time, but further investigations on more complex configurations are required. Also the efficiency of different resolution strategies for the resolution of

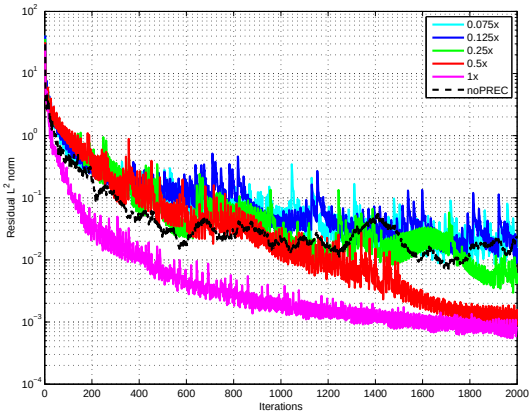


Figure 6.7: Residual value vs iterations for different values of n_U

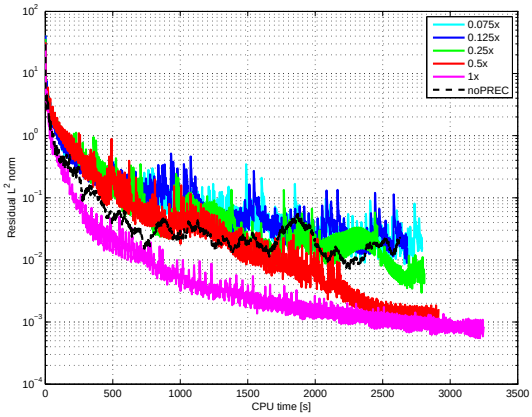


Figure 6.8: Residual value vs cpu time for different values of n_U

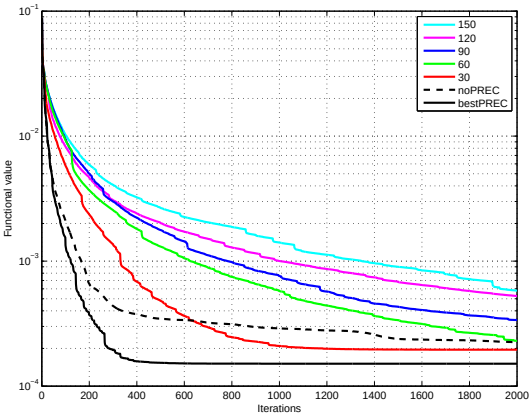


Figure 6.9: Functional value vs iterations for different values of coarse grid area, discretization of control variables on coarse grid equal to fine grid (ED).

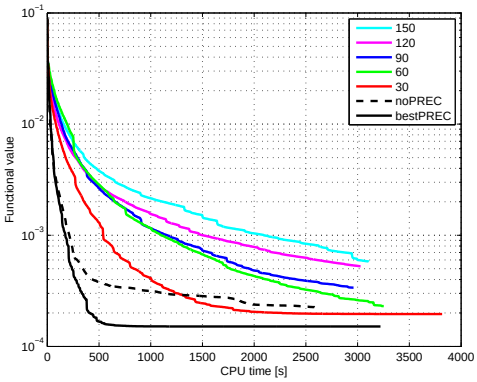


Figure 6.10: Functional value vs cpu time for different values of coarse grid area, discretization of control variables on coarse grid equal to fine grid (ED).

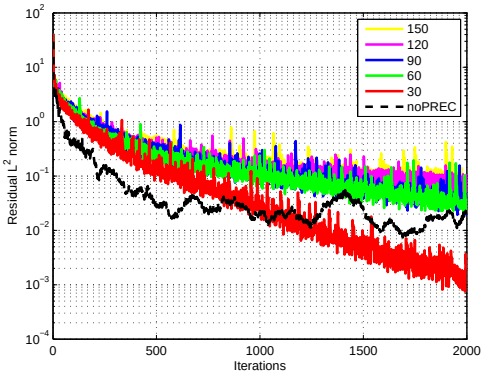


Figure 6.11: System residual value vs iterations for different values of coarse grid area, discretization of control variables on coarse grid equal to fine grid (ED).

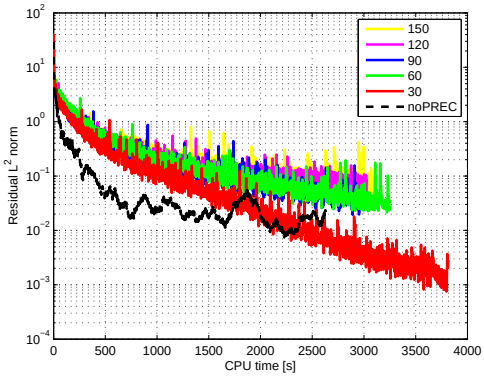


Figure 6.12: System residual value vs cpu time for different values of coarse grid area, discretization of control variables on coarse grid equal to fine grid (ED).

system 6.1 need to be evaluated. The obtained speed up is not comparable to the exponential convergence velocity achievable with proper multi-grid preconditioners, and a spectral analysis of the method is advisable to design preconditioning techniques capable of providing exponential convergence rates.

Table 6.1: Preconditioner behaviour for the DFN 120F, node strategy: ED

Panel A: $n_U = 1$, varying n_{CG}

Grid fine - Coarse	**CGsteps**	**Iter**	Iter cpu time [s]	Residual	J
	Ref.	2000	2628.72	0.000524984	0.000223922
	1	1743	2737.17	3.56557e-06	0.000151742
7 - 30	5	2000	3248.51	7.08703e-07	0.000151179
	10	2000	3219.46	1.64967e-07	0.000151127
	20	2000	2887.91	2.76264e-07	0.000151125

Panel B: $n_{\mathrm{CG}} = 5$, varying n_U

Grid fine - Coarse	**n_U**	**Iter**	Iter cpu time [s]	Residual	J
	Ref.	2000	2628.72	0.000524984	0.000223922
	0.075	2000	2791.64	0.000255202	0.000263584
7 - 30	0.125	2000	2681.03	0.000238779	0.000213198
	0.250	2000	2808.57	3.7036e-05	0.000196008
	0.500	2000	2912.5	2.61896e-06	0.00015778
	1.000	2000	3248.51	7.08703e-07	0.000151179

Bibliography

[1] A. Borzì and S. Volker, *Computational Optimization of Systems Governed by Partial Differential Equations*, Society for Industrial and Applied Mathematics, Philadephia, PA, 2011, ch. 5 - Multigrid Methods, pp. 67–176.

Part III

Virtual elements for DFN simulations: a preliminary investigation

Chapter 7

The Virtual Element Method for Discrete Fracture Network simulations

Abstract The present work discusses the application of the Virtual Element Method (VEM) to the simulation of discrete fracture network flows, with the optimization approach developed in [5, 6, 8]. The VEM is a newly developed technique for solving partial differential equation problems with meshes constituted of polygonal elements with an arbitrary number of edges. The generation of a conforming mesh is a demanding task for DFN simulations given the intricate geometry of realistic network configurations. The possibility of handling elements of arbitrary polygonal shape eases the process of mesh generation, still giving a mesh conforming to the trace on a given fracture, but non-conforming to the discretization of the intersecting fractures. The non-conformities are easily handled by the optimization approach used. The implementation of the VEM in the context of DFN simulations is fully described, and a panel of test problems and some numerical results on complex networks are provided to show the effectiveness of the method.

7.1 Introduction

Subsurface fluid flow has applications in a wide range of fields, including e.g. oil/gas recovery, gas storage, pollutant percolation, water resources monitoring, etc. Under-

ground fluid flow is a complex heterogeneous multi-scale phenomenon that involves complicated geological configurations. Discrete Fracture Networks (DFNs) are complex sets of planar polygonal fractures used to model subsurface fluid flow in fractured (porous) rocks. 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 [1, 13, 14], 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 focus 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 the Darcy law. 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. Matching conditions need to be added in order to preserve continuity along traces and flux balance at fracture intersections. 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. If this kind of conformity is required, the meshing process for each fracture is not independent of the others, leading in practice to a demanding computational effort for the mesh generation. 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.

Strategies are proposed in literature to ease the process of mesh generation and resolution for DFNs of large size. Some authors, see e.g. [15, 19], propose a simplification of DFN geometry to better handle the meshing procedure. In other cases, dimensional reduction is explored as in [11] and [12], 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 [17] and [18]).

In the recent paper [9] and follow up works [7] and [8], 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 neighbourhood of the traces. In [9, 7, 8] the XFEM is used in order to improve the solution near traces. 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, several 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 Mortar approach. This possibility is currently under investigation by the authors. 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 work is organized as follows: a description of the general problem is provided in Section 7.2, followed by a brief introduction to the application of virtual element method to the problem at hand in Section 7.3. Formulation and resolution of the discrete problem are sketched in Section 7.4. Some technical issues concerning VEM

implementation in this context as well as numerical results are given in Section 7.5. We end with some conclusions in Section 7.6.

7.2 Problem description

In this section we briefly sketch the main ideas of the PDE optimization method for discrete fracture network simulations introduced in [9, 7, 8].

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:

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

and

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

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 [9]. Then H_i satisfies, for

$i = 1, \ldots, I$, the following problem: find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\begin{aligned} \int_{F_i} \mathbf{K}_i \nabla H_i \nabla v d\Omega &= \int_{F_i} q_i v d\Omega + \langle G_i^N, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}),\mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} \\ &+ \sum_{S \in \mathcal{S}_i} \langle \left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(S),\mathrm{H}^{\frac{1}{2}}(S)}, \end{aligned} \tag{7.1}$$

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 (7.1) for $i = 1, ..., I$ are coupled with the following matching conditions, ensuring hydraulic head continuity and flux balance across the traces:

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i, j \in I_{S_m},\ \forall m = 1, \ldots, M, \tag{7.2}$$

$$\left[\!\left[\frac{\partial H_i}{\partial \hat{\nu}_{S_m}^i} \right]\!\right]_{S_m} + \left[\!\left[\frac{\partial H_j}{\partial \hat{\nu}_{S_m}^j} \right]\!\right]_{S_m} = 0, \qquad \text{for } i, j \in I_{S_m}. \tag{7.3}$$

The simultaneous resolution of equations (7.1)-(7.3) might result infeasible for practical applications, as previously discussed. In contrast, the approach developed in [9, 7, 8] 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

$$\begin{aligned} J(H, U) &= \sum_{m=1}^{M} \left(\left\| H_{i|_{S_m}} - H_{j|_{S_m}} \right\|^2_{\mathrm{H}^{\frac{1}{2}}(S)} \right. \\ &+ \left. \left\| U_i^{S_m} + U_j^{S_m} - \alpha \left(H_{i|_{S_m}} + H_{j|_{S_m}} \right) \right\|^2_{\mathrm{H}^{-\frac{1}{2}}(S)} \right). \end{aligned} \tag{7.4}$$

Equations (7.1), prescribed on the fractures, are equivalently restated as:

$$\begin{aligned} &\int_{F_i} \mathbf{K}_i \nabla H_i \nabla v d\Omega + \alpha \sum_{S \in \mathcal{S}_i} \int_S H_{i|_S} v_{|_S} d\Gamma = \\ &\int_{F_i} q_i v d\Omega + \langle G_i^N, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}),\mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} + \sum_{S \in \mathcal{S}_i} \langle U_i^S, v_{|S} \rangle_{\mathcal{U}^S, \mathcal{U}^{S\prime}}. \end{aligned} \tag{7.5}$$

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 (7.5) 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

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_i^N - A_i^D \mathcal{R}_i H_i^D, \quad in\ F_i, \tag{7.6}$$

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 (7.6) 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 [8], it can be shown that the unique minimum of functional (7.4) is obtained for values of H and of the control functions U that correspond to the fulfilment of conditions (7.2) and (7.3) on the traces. In other words, the solution of the problem

$$\min J \quad \text{subject to (7.6)} \tag{7.7}$$

corresponds to the solution of the coupled system of equations (7.1)-(7.3).

As shown in previous works (see e.g. [8]) 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 (7.7), satisfies the following system of equations, corresponding to the Fréchet derivatives of J with respect to the control variables: $\forall i = 1, \dots, I$

$$B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \left(U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right) - \alpha \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) = 0, \tag{7.8}$$

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

$$\begin{aligned} A_i^* P_i \;=\; & C_i{}^* \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} \left[\prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) - C_j^S H_j(U_j) \right) \right. \\ & \left. + \alpha^2 \prod_{S \in \mathcal{S}_i} \left(C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) \right] - \alpha C_i{}^* \left(U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right), \qquad \text{in } F_i, \end{aligned} \tag{7.9}$$

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

$$\nabla J(U_i) = B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S \in \mathcal{S}_i} (U_i^S + U_j^S - \alpha \Lambda_{\mathcal{U}^S}^{-1} (C_i^S H_i(U_i) + C_j^S H_j(U_j))), \tag{7.10}$$

and

$$\nabla J(U) = \prod_{i=1}^{I} \nabla J(U_i). \tag{7.11}$$

The gradient based algorithm for solving (7.7) is fully described in [8]. Here, we focus on a first-discretize-then-optimize approach, and we move on by introducing, in the next section, the space discretization.

7.3 The virtual element method

The Virtual Element Method [3, 4, 10, 2] 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 (7.6), 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

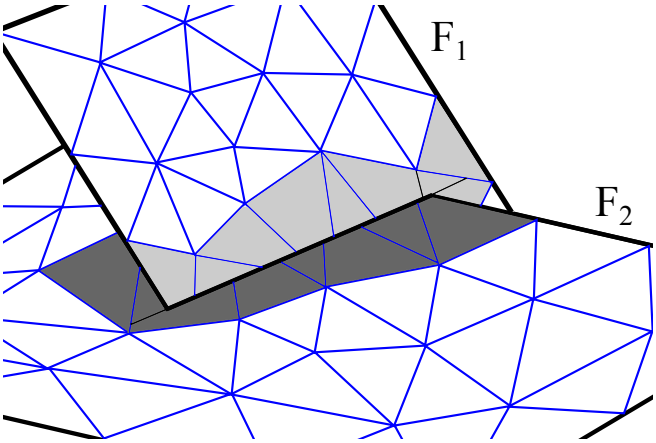


Figure 7.1: Example of the mesh for the VEM: elements shaded have been cut into polygons to match the trace on the two fractures independently

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 7.1).

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 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 [3]. 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 [3], 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 known explicitly 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 $a^E_{i,\delta}(\cdot,\cdot) : V_{i,\delta_{|E}} \times V_{i,\delta_{|E}} \mapsto \mathbb{R}$:

$$a^E_{i,\delta}(\phi, \varphi) = (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla \mathcal{P}^E \varphi)_E + \alpha \left(\phi_{|_{\mathcal{S}_i \cap \partial E}}, \varphi_{|_{\mathcal{S}_i \cap \partial E}} \right)_{\mathcal{S}_i \cap \partial E} + S^E(\phi, \varphi), \quad (7.12)$$

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

$$a_{i,\delta}(\phi, \varphi) = \sum_{E \in \mathcal{T}_{i,\delta}} a^E_{i,\delta}(\phi, \varphi). \quad (7.13)$$

In (7.12), 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

$$\begin{cases} (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla p)_E &= (\mathbf{K}_i \nabla \phi, \nabla p)_E \quad \forall p \in \mathbb{P}_1(E) \\ \sum_{k=1}^{n_E} \mathcal{P}^E \phi(\mathbf{x}_k) &= \sum_{k=1}^{n_E} \phi(\mathbf{x}_k) \end{cases} \quad (7.14)$$

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 7.1. Let us observe that the definition (7.12) for the bilinear form and (7.14) for the projection operator slightly differ from the definitions introduced in [3]. 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 (7.14) or in the definition of the stability operator S^E. According to [3] 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:

$$c_0(\mathbf{K}_i\,\nabla\varphi,\nabla\varphi)_E \le S^E(\varphi,\varphi) \le c_1(\mathbf{K}_i\,\nabla\varphi,\nabla\varphi)_E, \quad \forall\varphi\in V_{i,\delta|E},\ \text{with}\ \mathcal{P}^E\,\varphi = 0. \tag{7.15}$$

On each element E of the triangulation we have:

$$\begin{aligned}
a_i^E(\phi,\varphi) &= a_i^E(\phi-\mathcal{P}^E\,\phi+\mathcal{P}^E\,\phi,\varphi-\mathcal{P}^E\,\varphi+\mathcal{P}^E\,\varphi)\\
&= a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi)+a_i^E(\phi-\mathcal{P}^E\,\phi,\varphi-\mathcal{P}^E\,\varphi)\\
&\quad +a_i^E(\phi-\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi)+a_i^E(\mathcal{P}^E\,\phi,\varphi-\mathcal{P}^E\,\varphi)\\
&= a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi)+a_i^E(\phi-\mathcal{P}^E\,\phi,\varphi-\mathcal{P}^E\,\varphi)\\
&\quad +\alpha\left(\phi-\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi\right)_{\mathcal{S}_i\cap\partial E}+\alpha\left(\varphi-\mathcal{P}^E\,\varphi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}\\
&\quad +\left(\mathbf{K}_i\,\nabla(\phi-\mathcal{P}^E\,\phi),\nabla(\mathcal{P}^E\,\varphi)\right)_E+\left(\mathbf{K}_i\,\nabla(\varphi-\mathcal{P}^E\,\varphi),\nabla(\mathcal{P}^E\,\phi)\right)_E\\
&= a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi)+a_i^E(\phi-\mathcal{P}^E\,\phi,\varphi-\mathcal{P}^E\,\varphi)\\
&\quad +\alpha\left(\phi-\mathcal{P}^E\,\phi,\mathcal{P}^E\,\varphi\right)_{\mathcal{S}_i\cap\partial E}+\alpha\left(\varphi-\mathcal{P}^E\,\varphi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}
\end{aligned} \tag{7.16}$$

where the orthogonality condition (7.14) has been used for the last equality.

It is possible to show that the given definition of the bilinear form is consistent and stable. Consistency easily follows from definition (7.12) and from (7.14): for all $E\in\mathcal{T}_{i,\delta}$, $\forall p\in\mathbb{P}_1(E),\ \forall\phi\in\ V_{i,\delta|E}$ we have:

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

being $a_i^E(\cdot,\cdot)$ the restriction to a mesh element of the continuous bilinear form. Stability can be proved similarly to [3], using (7.12) and (7.16), as there exist two positive constants $\underline{a}$ and $\overline{a}$ independent from the element E and from δ such that $\forall\phi\in$

$V_{i,\delta_{|E}}$, $\quad \underline{a}\, a_i^E(\phi,\phi) \le a_{i,\delta}^E(\phi,\phi) \le \overline{a}\, a_i^E(\phi,\phi)$. For all $\phi \in V_{i,\delta_{|E}}$ we have:

$$\begin{aligned}
a_{i,\delta}^E(\phi,\phi) &= \left(\mathbf{K}_i\,\nabla(\mathcal{P}^E\,\phi),\nabla(\mathcal{P}^E\,\phi)\right)_E + \alpha\,(\phi,\phi)_{\mathcal{S}_i\cap\partial E} + S^E(\phi-\mathcal{P}^E\,\phi,\phi-\mathcal{P}^E\,\phi)\\
&= \left(\mathbf{K}_i\,\nabla(\mathcal{P}^E\,\phi),\nabla(\mathcal{P}^E\,\phi)\right)_E + \alpha\left(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}\\
&\quad -\alpha\left(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E} + \alpha\,(\phi,\phi)_{\mathcal{S}_i\cap\partial E} + S^E(\phi-\mathcal{P}^E\,\phi,\phi-\mathcal{P}^E\,\phi)\\
&\le a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi) + \alpha\,(\phi,\phi)_{\mathcal{S}_i\cap\partial E} - \alpha\left(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}\\
&\quad +c_1\left(\mathbf{K}_i\,\nabla(\phi-\mathcal{P}^E\,\phi),\nabla(\phi-\mathcal{P}^E\,\phi)\right)_E\\
&\le \max\{1,c_1\}\Big(a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi) + \left(\mathbf{K}_i\,\nabla(\phi-\mathcal{P}^E\,\phi),\nabla(\phi-\mathcal{P}^E\,\phi)\right)_E\\
&\quad +\alpha\left(\phi-\mathcal{P}^E\,\phi,\phi-\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E} + 2\alpha\left(\phi-\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}\Big)\\
&= \overline{a}\, a_i^E(\phi,\phi),
\end{aligned}$$

and in a similar fashion:

$$\begin{aligned}
a_{i,\delta}^E(\phi,\phi) &\ge \min\{1,c_0\}\Big(a_i^E(\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi) + \left(\mathbf{K}_i\,\nabla(\phi-\mathcal{P}^E\,\phi),\nabla(\phi-\mathcal{P}^E\,\phi)\right)_E\\
&\quad +\alpha\left(\phi-\mathcal{P}^E\,\phi,\phi-\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E} + 2\alpha\left(\phi-\mathcal{P}^E\,\phi,\mathcal{P}^E\,\phi\right)_{\mathcal{S}_i\cap\partial E}\Big)\\
&= \underline{a}\, a_i^E(\phi,\phi).
\end{aligned}$$

Assuming basic quality properties for the triangulation, functional S^E can be chosen as in [3] to satisfy conditions (7.15), thus having for all $\phi,\varphi \in V_{i,\delta_{|E}}$:

$$S^E(\phi,\varphi) = \sum_{k=1}^{n_E} \mathbf{K}_i(\phi(\mathbf{x}_k) - (\mathcal{P}^E\,\phi)(\mathbf{x}_k))(\varphi(\mathbf{x}_k) - (\mathcal{P}^E\,\varphi)(\mathbf{x}_k)). \tag{7.17}$$

Concerning the treatment of the source term q_i at right hand side of equation (7.6), it is shown in [4] that convergence rates are preserved approximating q_i with a piecewise constant function on each element of the triangulation.

Given the previous results and definitions it is possible to use the convergence theorem in [3] to prove that the discrete problems on the fractures are well posed and enjoy the convergence rates of standard finite elements of the same order.

Even if functions in $V_{i,\delta}$ are only known on the edges of triangulation 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:

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

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

7.4 Formulation and resolution of the discrete problem

As shown in Section 7.2, the problem has been reformulated as a PDE-constrained optimization problem (see equation (7.7)) 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.

7.4.1 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 7.2 by equation (7.4), 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

$$
\begin{aligned}
J \;=\; & \frac{1}{2} \sum_{i=1}^{I} \sum_{S \in \mathcal{S}_i} \left(\int_S \Big(\sum_{k=1}^{N_i} h_{i,k} \phi_{i,k|S} - \sum_{k=1}^{N_j} h_{j,k} \phi_{j,k|S} \Big)^2 \mathrm{d}\gamma + \right. \\
& \left. \int_S \Big(\sum_{k=1}^{N_m^-} u_{m,k}^- \psi_{m,k}^- + \sum_{k=1}^{N_m^+} u_{m,k}^+ \psi_{m,k}^+ - \alpha \sum_{k=1}^{N_i} h_{i,k} \phi_{i,k|S} - \alpha \sum_{k=1}^{N_j} h_{j,k} \phi_{j,k|S} \Big)^2 \mathrm{d}\gamma \right).
\end{aligned} \tag{7.18}
$$

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

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

The functional J in (7.18) 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

$$G_{ii}^h = (1+\alpha^2) C_{i,i}, \qquad G_{ij}^h = (\alpha^2 - 1) C_{i,j}^S \text{ if } j \in J_i \text{ (0 elsewhere)},$$

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 $(G_{ij}^h)^T = G_{ji}^h$, 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 (7.6) 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

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

where the operators $\mathcal{P}^E$ and S^E are defined by (7.14) and (7.17), 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 (7.6) is then

$$A_i h_i = \tilde{q}_i + \mathcal{B}_i \, u'_i, \qquad i = 1, ..., I, \tag{7.19}$$

where $\tilde{q}_i$ accounts for the term q_i in (7.6) 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 (7.19) are then written $Ah - \mathcal{B}\, u = q$.

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

$$\begin{aligned} \min\ J(h,u) &= \frac{1}{2}\left(h^T G^h h - \alpha h^T B u - \alpha u^T B^T h + u^T G^u u\right) && (7.20) \\ \text{s.t.} \quad & Ah - \mathcal{B}\, u = q. && (7.21) \end{aligned}$$

7.4.2 Solving the optimization problem

The first order optimality conditions for problem (7.20)-(7.21) are the following:

$$\begin{pmatrix} G^h & -\alpha B & A^T \\ -\alpha B^T & G^u & -\mathcal{B}^T \\ A & -\mathcal{B} & 0 \end{pmatrix} \begin{pmatrix} h \\ u \\ -p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix} \qquad (7.22)$$

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

$$h = A^{-1}(\mathcal{B}\, u + q), \qquad (7.23)$$

we obtain the following equivalent unconstrained problem :

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

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

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

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

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

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

and recalling (7.23), one has

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

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

$$A^T p = G^h h - \alpha B u. \tag{7.24}$$

With these definitions, we may write

$$\nabla \hat{J}(u) = \mathcal{B}^T\, p + G^u u - \alpha B^T h. \tag{7.25}$$

Note that setting to zero the previous expression for obtaining stationary points for $\hat{J}(u)$, and collecting such equation together with (7.23) and (7.24), we obtain system (7.22).

Concerning the numerical solution of the optimization problem, we mention here two possible approaches. The first one consists in solving the linear system (7.22). An iterative solver is clearly a recommended choice, and `symmlq` [16] would be a suitable choice; this approach has been used in [7]. 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 (7.23) and (7.24) and g_0 by (7.25)
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

$$\lambda_k = -\frac{d_k^T g_k}{d_k^T \hat{G} d_k}. \tag{7.26}$$

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.

7.5 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 7.4. 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.

7.5.1 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 in their own right, as shown in Figures 7.2 and 7.3 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

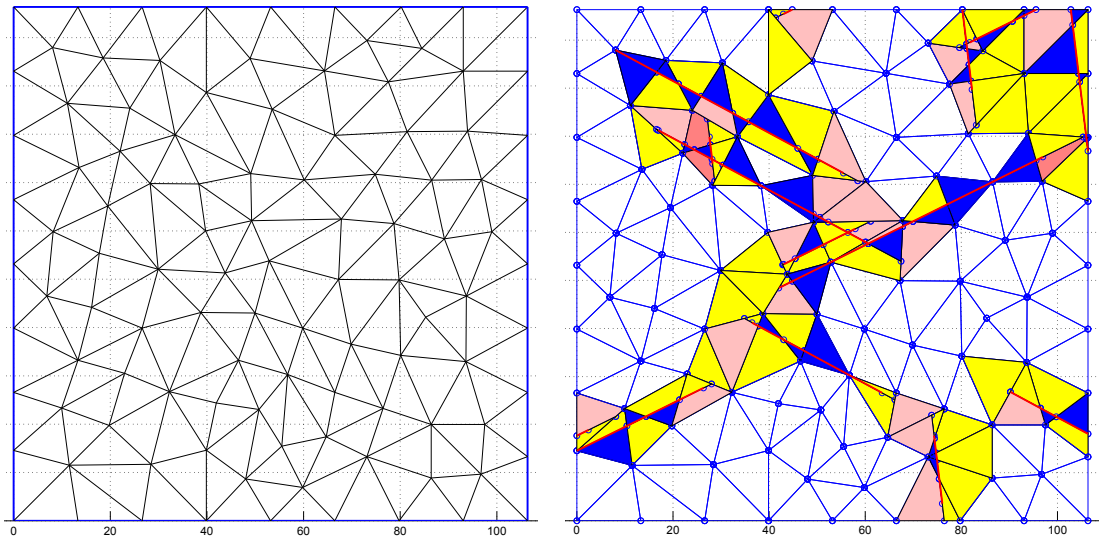


Figure 7.2: Mesh example. Left: original triangulation. Right: mesh for VEM.

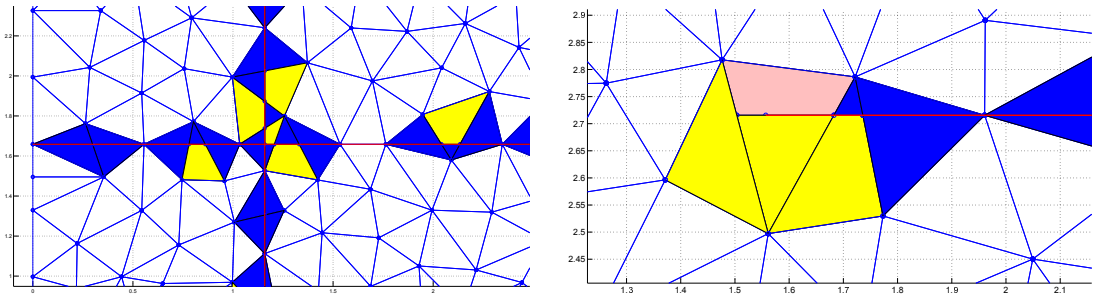


Figure 7.3: Left: detail of a mesh around a trace intersection. Right: detail of a mesh around a trace tip.

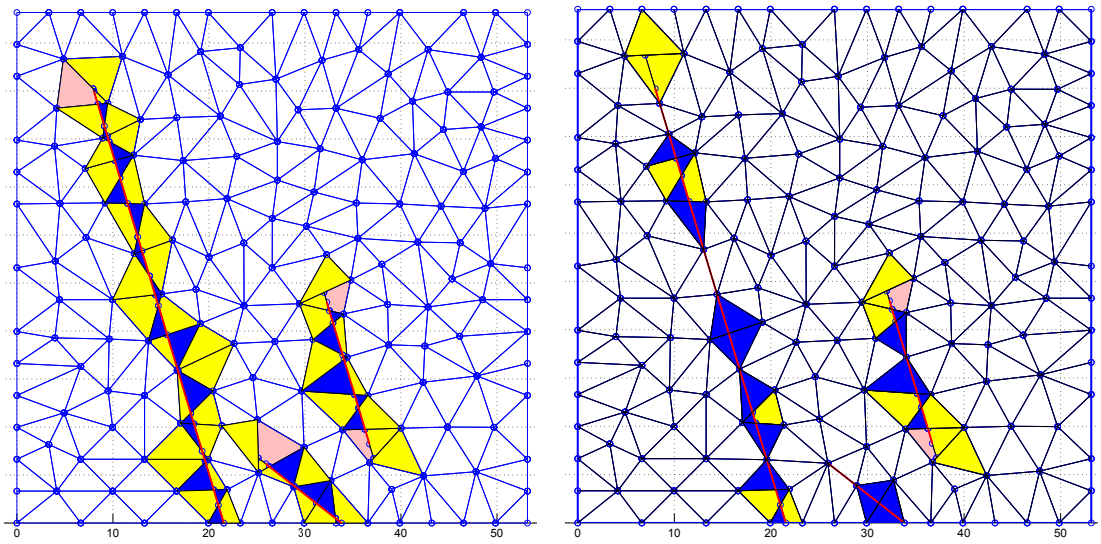


Figure 7.4: Left: example of VEM mesh without modification. Right: Same mesh after modifications.

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 7.4.

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 $i = 1, \ldots, I$, 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.

7.5.2 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 (7.1) 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.

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 7.5 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:

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

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,

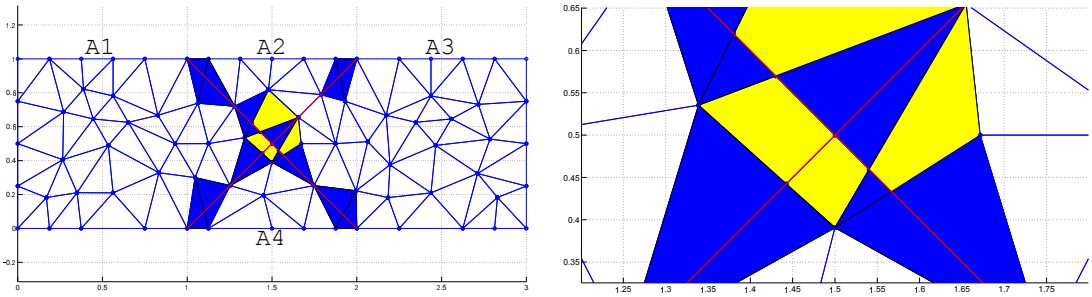


Figure 7.5: Problem P1. Left: Domain with coarse grid $\delta_{max} = 0.2$. Right: a detail of trace intersection.

as indicated in Figure 7.5. Values of f_{S_1} and f_{S_2} are

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

and

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

In Figure 7.7, left, 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 [9, 7, 8]. Figure 7.7, right, 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 [7, 8] 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

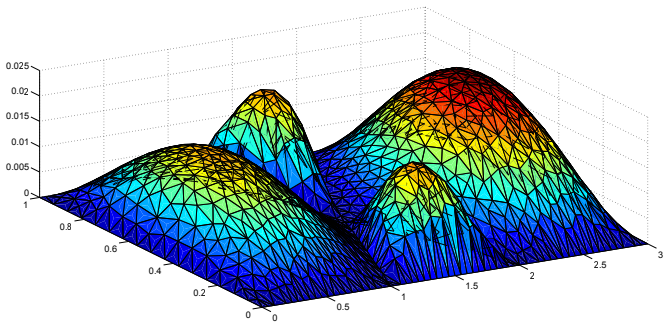


Figure 7.6: Problem P1: approximate solution on a mesh with $\delta_{max} = 0.05$

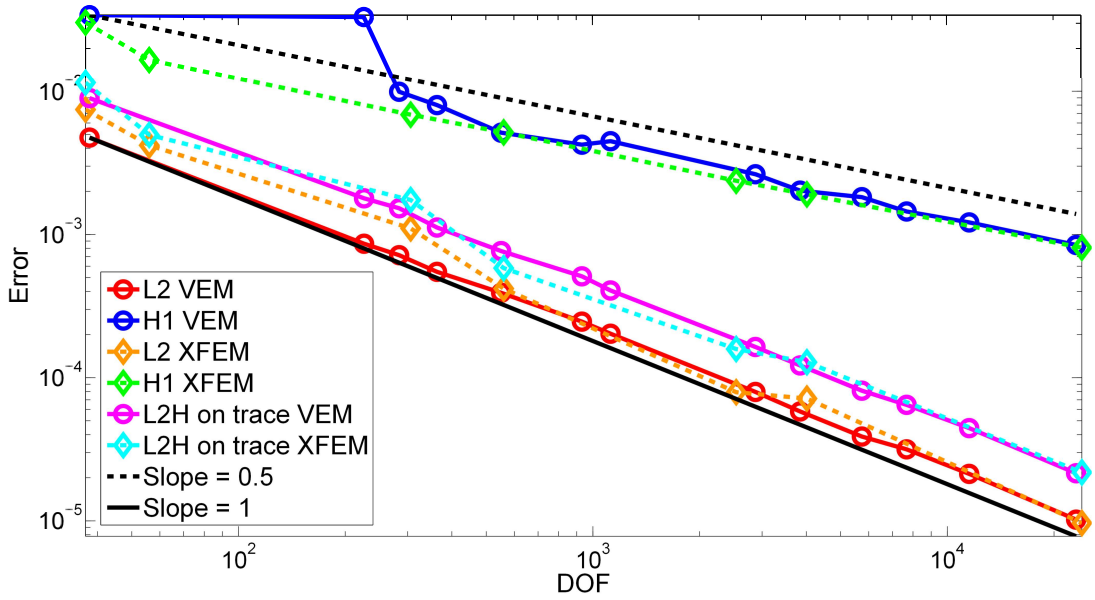


Figure 7.7: Problem P1: error behaviour

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.

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 $S = \{(x,y) \in \mathbb{R}^2 : y = 0 \text{ and } -1 \leq x \leq 0\}$ ending in the interior of F. This test problem has also been considered in [7]. 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 $\text{arctan2}(x,y)$ 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 $\arctan(\cdot)$ 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 7.8 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 7.9. In Figure 7.10 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 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.

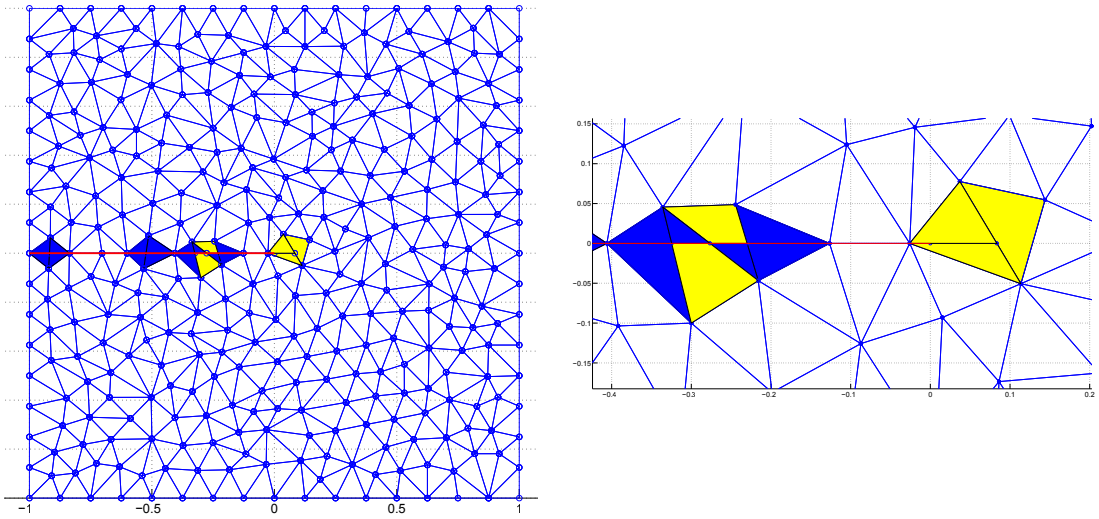


Figure 7.8: Problem P2. Domain meshed with $\delta_{max} = 0.1$. Right: a detail of elements near trace tip.

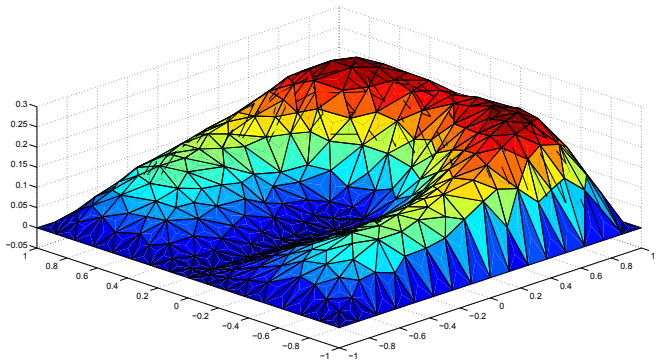


Figure 7.9: Problem P2: approximate solution with VEM obtained with a mesh with $\delta_{max} = 0.1$

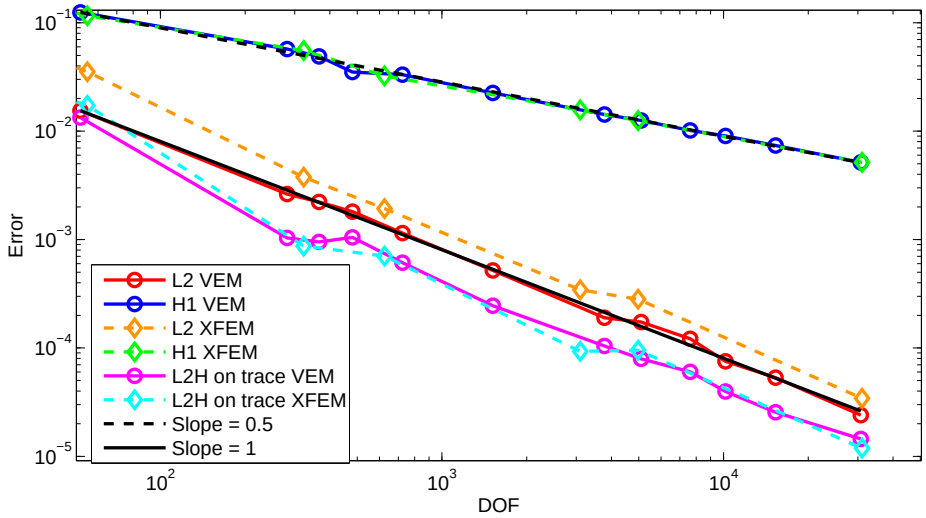


Figure 7.10: Problem P2: error behaviour

7.5.3 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 discretization. The general DFN problem is set as follows:

$$\begin{aligned} -\Delta H &= q && \Omega \setminus \mathcal{S}, \\ H_{|\Gamma_D} &= H^D && on\ \Gamma_D, \\ \frac{\partial H}{\partial \hat{\nu}} &= G^N && on\ \Gamma_N, \end{aligned} \tag{7.27}$$

with reference to the nomenclature introduced in Section 7.2.

DFN2

Here we analyze a very simple DNF consisting of two identical fractures that intersect each other orthogonally, as can be seen in Figure 7.11 where the domain Ω is depicted.

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

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

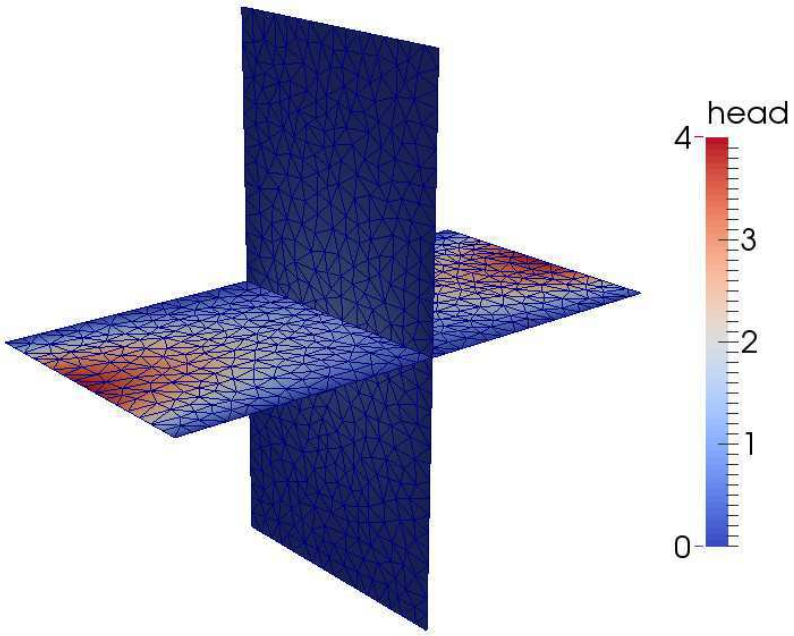


Figure 7.11: DFN2: spatial distribution of fractures and the obtained solution for the hydraulic head.

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 we set homogeneous Neumann conditions for fracture F_1, and a non-constant Neumann boundary condition for fracture F_2 given by $G^N = 16y(1-y)^2 on\ \Gamma_N$. 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 7.12 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 7.13 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

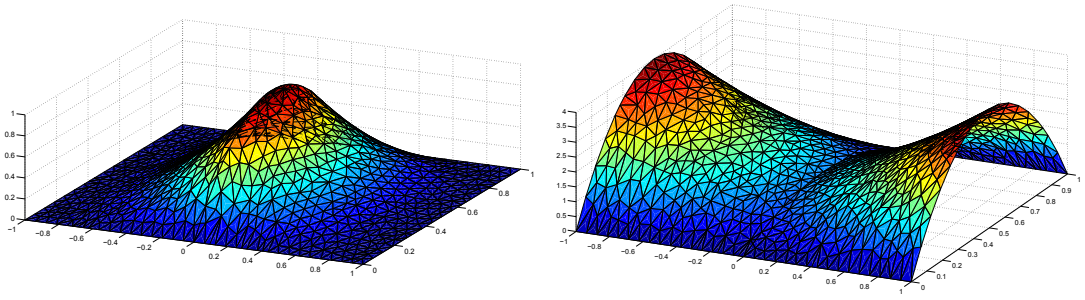


Figure 7.12: DFN2: approximate solution for fracture 1 (left) and fracture 2 (right).

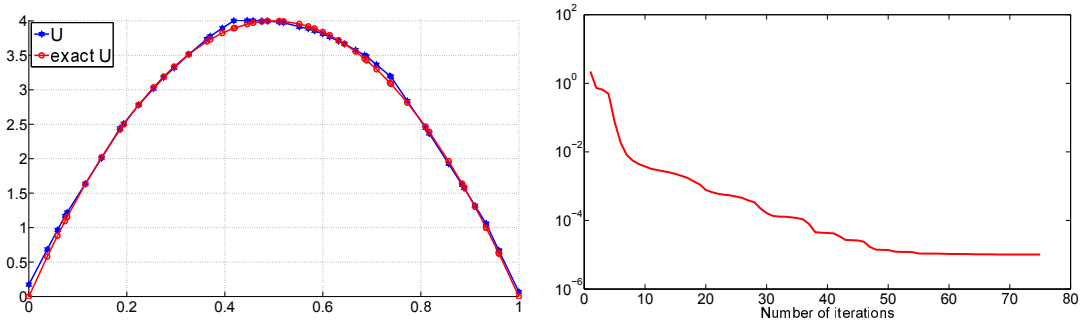


Figure 7.13: DFN2. Left: comparison between exact and approximate flux. Right: values of J versus number of iterations.

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 7.14 against the number of degrees 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.

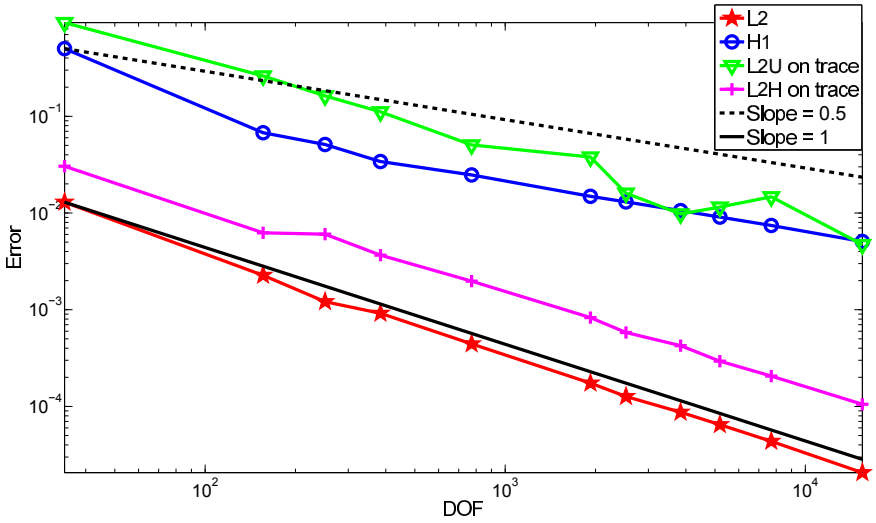


Figure 7.14: DFN2: error behaviour

DFN7

This problem consists of 7 fractures intersecting in 11 traces. The spatial distribution of the fractures can be seen in Figure 7.15. The source term is $q = 0$ in equation (7.27).

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 7.15) 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.

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 7.16 for a detail of the meshes for fracture 3.

Table 7.1 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 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

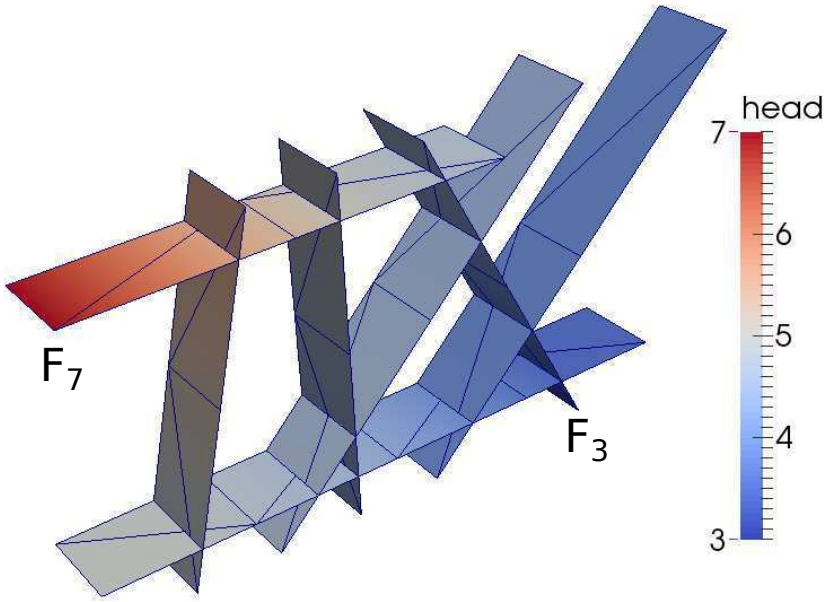


Figure 7.15: DFN7: spatial distribution of fractures and the obtained solution for the hydraulic head.

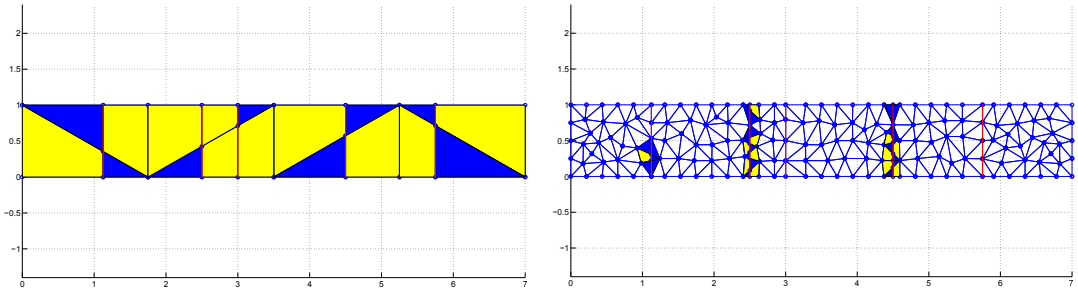


Figure 7.16: DFN7: mesh on F_6 with parameter $\delta_{max} = 1.2$ (left) and finer mesh with $\delta_{max} = 0.2$ (right).

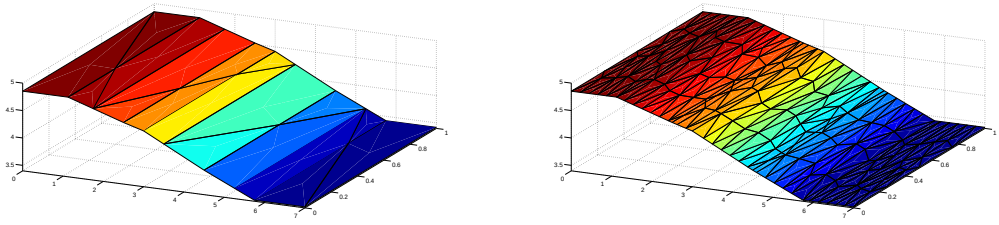


Figure 7.17: DFN7: solutions obtained for fracture 6 with coarse (left) and fine (right) mesh.

Table 7.1: 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

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.

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 7.18. 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 7.18, 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 (7.27).

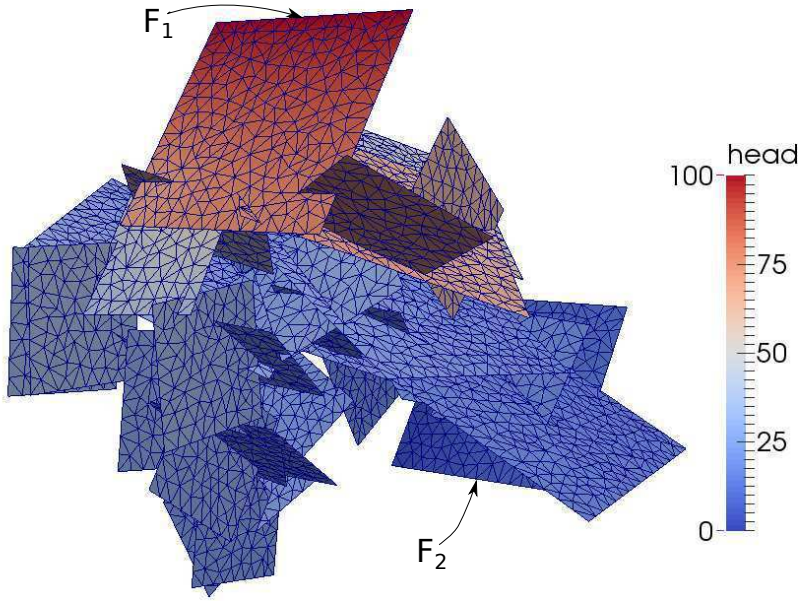


Figure 7.18: DFN36: Spatial distribution of fractures and the obtained solution for the hydraulic head.

The problem is solved on several meshes, with $2\text{m}^2 < \delta^2_{\max} < 50\text{m}^2$. In Figure 7.19 the detail of a mesh with $\delta^2_{\max} = 30\text{m}^2$ on a selected fracture and the corresponding obtained solution are shown.

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 7.2 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 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

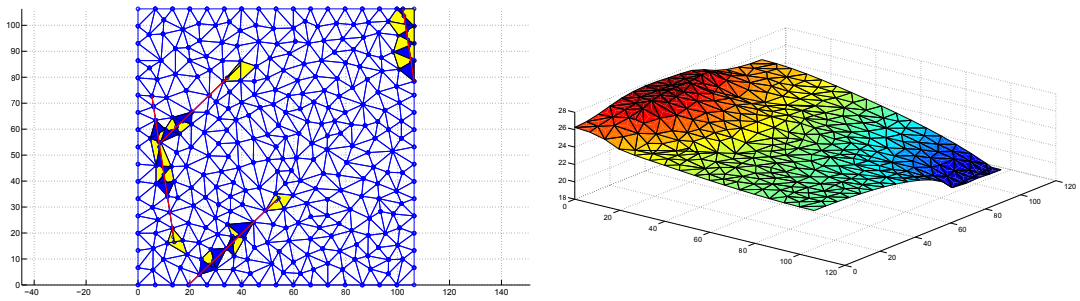


Figure 7.19: DFN36: Left: Mesh with maximum element size of 30m^2 on a selected fracture. Right: Solution on the same grid.

Table 7.2: 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

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.

7.6 Conclusions

The very recent Virtual Element Method is coupled with the optimization based algorithm presented in [9, 7, 8] 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.

Bibliography

[1] P. M. Adler, *Fractures and Fracture Networks*, Kluwer Academic, Dordrecht, 1999.

[2] B. Ahmad, A. Alsaedi, F. Brezzi, L. D. Marini, and A. Russo, *Equivalent projectors for virtual element methods*, Comput. Math. Appl., 66 (2013), pp. 376–391.

[3] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo, *Basic principles of virtual element methods*, Math. Models Methods Appl. Sci., 23 (2013), pp. 199–214.

[4] L. Beirão da Veiga, F. Brezzi, and L. D. Marini, *Virtual elements for linear elasticity problems*, SIAM J. Numer. Anal., 51 (2013), pp. 794–812.

[5] S. Berrone, S. Pieraccini, and S. Scialó, *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM Journal on Scientific Computing, 35 (2013), pp. B487–B510.

[6] ——, *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM Journal on Scientific Computing, 35 (2013), pp. A908–A935.

[7] ——, *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM Journal on Scientific Computing, (2013). To appear. Preprint available at http://porto.polito.it/id/eprint/2499005.

[8] S. Berrone, S. Pieraccini, and S. Scialò, *An optimization approach for large scale simulations of discrete fracture network flows*, SIAM J. Sci. Comput., J. Comput. Phys. (2013).

[9] S. Berrone, S. Pieraccini, and S. Scialó, *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM Journal on Scientific Computing, (2013). To appear. Preprint available at http://porto.polito.it/id/eprint/2495599.

[10] F. Brezzi and L. D. Marini, *Virtual element methods for plate bending problems*, Comput. Methods Appl. Mech. Engrg., 253 (2013), pp. 455–462.

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

[12] W. S. Dershowitz and C. Fidelibus, *Derivation of equivalent pipe networks analogues for three-dimensional discrete fracture networks by the boundary element method*, Water Resource Res., 35 (1999), pp. 2685–2691.

[13] J. R. D. Dreuzy, P. Davy, and O. Bour, *Hydraulic properties of two-dimensional random fracture networks following a power law length distribution: 2., permeability of networks based on log-normal distribution of apertures*, Water Resour. Res., 37 (2001), pp. 2079–2095.

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

[15] T. Kalbacher, R. Mettier, C. McDermott, W. Wang, G. Kosakowski, T. Taniguchi, and O. Kolditz, *Geometric modelling and object-oriented software concepts applied to a heterogeneous fractured network from the grimsel rock laboratory*, Comput. Geosci., 11 (2007), pp. 9–26.

[16] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629.

[17] G. Pichot, J. Erhel, and J. de Dreuzy, *A mixed hybrid mortar method for solving flow in discrete fracture networks*, Applicable Analysis, 89 (2010), pp. 1629 – 643.

[18] ——, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86 – B105.

[19] M. Vohralík, J. Maryška, and O. Severýn, *Mixed and nonconforming finite element methods on a system of polygons*, Applied Numerical Mathematics, 51 (2007), pp. 176–193.