Politecnico di Torino

Porto Institutional Repository

[Article] The virtual element method for discrete fracture network simulations

*Original Citation:*
Benedetto, M.F.; Berrone, S.; Pieraccini, S.; Scialò, S. (2014). *The virtual element method for discrete fracture network simulations.* In: COMPUTER METHODS IN APPLIED MECHANICS AND ENGINEERING, vol. 280 n. 1, pp. 135-156. - ISSN 0045-7825

*Availability:*
This version is available at : http://porto.polito.it/2526323/ since: January 2014

*Publisher:*
Elsevier

*Published version:*
DOI:10.1016/j.cma.2014.07.016

*Terms of use:*
This article is made available under terms and conditions applicable to Open Access Policy Article ("Public - All rights reserved") , as described at http://porto.polito.it/terms_and_conditions.html

Porto, the institutional repository of the Politecnico di Torino, is provided by the University Library and the IT-Services. The aim is to enable open access to all the world. Please share with us how this access benefits you. Your story matters.

(Article begins on next page)

# THE VIRTUAL ELEMENT METHOD FOR DISCRETE FRACTURE NETWORK SIMULATIONS *

MATÍAS FERNANDO BENEDETTO†, STEFANO BERRONE†, SANDRA PIERACCINI†, AND STEFANO SCIALÒ†

February 11, 2014

**Abstract.** In this work, an optimization based approach presented in [5, 6, 7] for Discrete Fracture Network simulations is coupled with the Virtual Element Method (VEM) for the space discretization of the underlying Darcy law. The great flexibility of the VEM in handling rather general polygonal elements allows, in a natural way, for an effective description of irregular solutions starting from an arbitrary triangulation, which is built independently of the mesh on other fractures. Only partial conformity is in fact obtained with this approach. Numerical results performed on several DFN configurations confirm the viability and efficiency of the resulting method.

**Key words.** VEM, Fracture flows, Darcy flows, discrete fracture networks, optimization methods for elliptic problems

**AMS subject classifications.** 65N30, 65N15, 65N50, 65J15

**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. Underground 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, 11, 12], 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

*This work was supported by Italian funds MIUR-PRIN-2012 (2012HBLYE4_001), and by INdAM-GNCS (project: *Aspetti emergenti nello studio di strategie adattative per problemi differenziali* ).

†Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. e-mail: {matias.benedetto,stefano.berrone,sandra.pieraccini, stefano.scialo}@polito.it

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. [13, 17], propose a simplification of DFN geometry to better handle the meshing procedure. In other cases, dimensional reduction is explored as in [9] and [10], 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 [15] and [16]).

In the recent paper [5] and follow up works [6] and [7], 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 [5, 6, 7] 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 2, followed by a brief introduction to the application of virtual element method to the problem at hand in Section 3. Formulation and resolution of the discrete problem are sketched in Section 4. Some technical issues concerning VEM implementation in this context as well as numerical results are given in Section 5. We end with some conclusions in Section 6.

**2. Problem description.** In this section we briefly sketch the main ideas of the PDE optimization method for discrete fracture network simulations introduced in

[5, 6, 7].

Let us denote by $\Omega$ 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 $\partial F_i$ 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 $\Gamma_D$ the Dirichlet boundary and $\Gamma_N$ 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 $\partial F_i$ 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 $\Omega$ 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 [5]. Then $H_i$ satisfies, for $i = 1, \dots, I$, the following problem: find $H_i \in V_i^D$ such that $\forall v \in V_i$

$$\begin{aligned} \int_{F_i} \mathbf{K}_i \nabla H_i \nabla v \mathrm{d}\Omega = \int_{F_i} q_i v \mathrm{d}\Omega + \langle G_i^N, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} \\ + \sum_{S \in \mathcal{S}_i} \langle \left[\!\left[ \frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S, v_{|S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(S), \mathrm{H}^{\frac{1}{2}}(S)}, \end{aligned} \tag{2.1}$$

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

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

with $\hat{n}_i$ outward normal to the boundary $\Gamma_{iN}$, and $\left[\!\left[ \frac{\partial H_i}{\partial \hat{\nu}_S^i} \right]\!\right]_S$ denotes the jump of the co-normal derivative along the unique normal $\hat{n}_S^i$ fixed for the trace $S$ on $F_i$, and represents the flux incoming into the fracture $F_i$ through the trace $S$. The equations (2.1) for $i = 1, ..., I$ are coupled with the following matching conditions, ensuring hydraulic head continuity and flux balance across the traces:

$$H_{i|S_m} - H_{j|S_m} = 0, \qquad \text{for } i, j \in I_{S_m}, \ \forall m = 1, \dots, M, \tag{2.2}$$

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

The simultaneous resolution of equations (2.1)-(2.3) might result infeasible for practical applications, as previously discussed. In contrast, the approach developed in [5, 6, 7] 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 $U_i^S = \alpha H_{i|_S} + \left[\!\left[ \frac{\partial H_i}{\partial \vec{\nu}_S^i} \right]\!\right]_S$, where $\alpha$ is a fixed positive parameter, and the quadratic functional

$$J(H,U) = \sum_{m=1}^{M} \Big( \left\| H_{i|_{S_m}} - H_{j|_{S_m}} \right\|^2_{\mathrm{H}^{\frac{1}{2}}(S)} + \left\| U_i^{S_m} + U_j^{S_m} - \alpha \left( H_{i|_{S_m}} + H_{j|_{S_m}} \right) \right\|^2_{\mathrm{H}^{-\frac{1}{2}}(S)} \Big). \tag{2.4}$$

Equations (2.1), prescribed on the fractures, are equivalently restated as:

$$\int_{F_i} \mathbf{K}_i \, \nabla H_i \nabla v \mathrm{d}\Omega + \alpha \sum_{S \in \mathcal{S}_i} \int_S H_{i|_S} v_{|_S} \mathrm{d}\Gamma = \int_{F_i} q_i v \mathrm{d}\Omega + \langle G_i^N, v_{|_S} \rangle_{\mathrm{H}^{-\frac{1}{2}}(\Gamma_{iN}), \mathrm{H}^{\frac{1}{2}}(\Gamma_{iN})} + \sum_{S \in \mathcal{S}_i} \langle U_i^S, v_{|_S} \rangle_{\mathcal{U}^S, \mathcal{U}^{S\prime}}. \tag{2.5}$$

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

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

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

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

According to this functional setting and definitions, problems (2.5) are restated as: $\forall i = 1, ..., I$, find $H_i \in V_i^D$, with $H_i = H_i^0 + \mathcal{R}_i \, H_i^D$ and $H_i^0 \in V_i$, such that

$$A_i H_i^0 = q_i + B_i U_i + B_{iN} G_i^N - A_i^D \, \mathcal{R}_i \, H_i^D, \quad \text{in } F_i, \tag{2.6}$$

where $A_i^D$ is an operator defined similarly to $A_i$, but operating on elements in $\mathrm{H}^1(F_i)$. We remark that, if $\alpha > 0$, for a given $U_i$, the solution $H_i$ to (2.6) exists and is unique for a non isolated fracture even if we set Neumann boundary conditions on the whole $\partial F_i$.

Following the arguments proposed in [7], it can be shown that the unique minimum of functional (2.4) is obtained for values of $H$ and of the control functions $U$

that correspond to the fulfilment of conditions (2.2) and (2.3) on the traces. In other words, the solution of the problem

$$\min J \quad \text{subject to (2.6)} \tag{2.7}$$

corresponds to the solution of the coupled system of equations (2.1)-(2.3).

As shown in previous works (see e.g. [7]) this optimization problem can be tackled with a gradient based method. Even if different approaches could also be employed, gradient-based methods are particularly appealing since they allow to independently solve problems on fractures and can be straightforwardly plugged in a parallel resolution process.

In the continuous setting, the gradient based method is formally devised on the following considerations: the optimal $U \in \mathcal{U}$, solution to (2.7), satisfies the following system of equations, corresponding to the Fréchet derivatives of $J$ with respect to the control variables: $\forall i = 1, \dots, I$

$$B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \left( U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right) - \alpha \prod_{S \in \mathcal{S}_i} \left( C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) = 0, \tag{2.8}$$

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

$$\begin{aligned} A_i^* P_i = C_i{}^* \Lambda_{\mathcal{U}^{\mathcal{S}_i}}^{-1} \Bigg[ & \prod_{S \in \mathcal{S}_i} \left( C_i^S H_i(U_i) - C_j^S H_j(U_j) \right) \\ & + \alpha^2 \prod_{S \in \mathcal{S}_i} \left( C_i^S H_i(U_i) + C_j^S H_j(U_j) \right) \Bigg] - \alpha C_i{}^* \left( U_i + \prod_{S \in \mathcal{S}_i} U_j^S \right), \quad \text{in } F_i, \end{aligned} \tag{2.9}$$

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

$$\nabla J(U_i) = B_i{}^* P_i + \Lambda_{\mathcal{U}^{\mathcal{S}_i}} \prod_{S \in \mathcal{S}_i} (U_i^S + U_j^S - \alpha \Lambda_{\mathcal{U}^S}^{-1} (C_i^S H_i(U_i) + C_j^S H_j(U_j))), \tag{2.10}$$

and

$$\nabla J(U) = \prod_{i=1}^{I} \nabla J(U_i). \tag{2.11}$$

The gradient based algorithm for solving (2.7) is fully described in [7]. Here, we focus on a first-discretize-then-optimize approach, and we move on by introducing, in the next section, the space discretization.

**3. The virtual element method.** The Virtual Element Method [3, 4, 8, 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 (2.6), whose solution can have a discontinuous gradient across the traces. In order to correctly reproduce this irregular behaviour, we can take advantage of the flexibility of virtual elements by transforming, on each fracture, a given triangulation (non conforming to traces) in a more general mesh, conforming to traces, simply obtained by splitting the triangles along traces into more general sub-polygons not crossed by traces. We remark that we do not require conformity between the meshes of the two fractures intersecting at a trace. As a consequence of the meshing process,

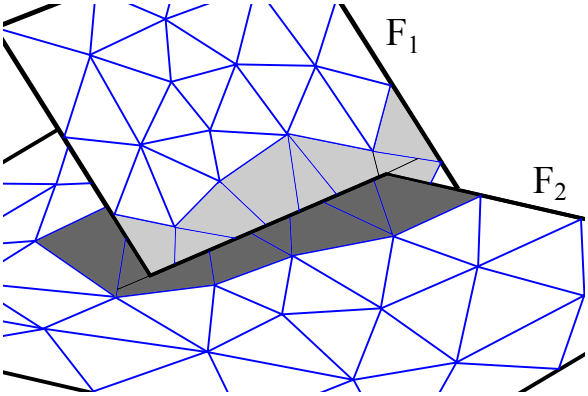


FIGURE 3.1. *Example of the mesh for the VEM: elements shaded have been cut into polygons to match the trace on the two fractures independently*

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 3.1).

Let us now describe the application of the VEM to the problem considered. For the sake of simplicity, we consider in this section homogeneous conditions on the Dirichlet boundary; furthermore, we consider in this work the case of virtual elements of order $k = 1$ and we assume that the fracture transmissivity $\mathbf{K}_i$ is constant on each fracture, but might vary from one fracture to another. We will focus on a generic fracture $F_i \subset \Omega$, since the process is independent on each fracture. Let $\{\mathcal{T}_{i,\delta}\}_\delta$ be a family of meshes on $F_i$, being $\delta$ 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 $\delta$ 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 $\partial E$, or, equivalently, specifying the values at the $n_E$ vertices of the polygon. With this natural choice for the degrees of freedom, the $C^0$ continuity of functions in $V_{i,\delta}$ is easily enforced. The dimension of $V_{i,\delta}$ is $N_i$, and we introduce a Lagrange basis $\{\phi_1, \ldots, \phi_{N_i}\}$, defined by $\phi_j(x_k) = \delta_{jk}$, where $x_k$ is the $k$-th vertex

in the mesh. Functions $\{\phi_j\}$ are in general not explicitly known inside the elements, but only on the boundaries of the elements, and this is a key point of VEM. Further we observe that the space of polynomials $\mathbb{P}_1(E) \subset V_{i,\delta}{}_{|E}$ for each element $E$ in $\mathcal{T}_{i,\delta}$.

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

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

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

$$a^E_{i,\delta}(\phi,\varphi) = (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla \mathcal{P}^E \varphi)_E + \alpha \left(\phi_{|_{\mathcal{S}_i \cap \partial E}}, \varphi_{|_{\mathcal{S}_i \cap \partial E}}\right)_{\mathcal{S}_i \cap \partial E} + S^E(\phi,\varphi), \quad (3.1)$$

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

$$a_{i,\delta}(\phi,\varphi) = \sum_{E \in \mathcal{T}_{i,\delta}} a^E_{i,\delta}(\phi,\varphi). \quad (3.2)$$

In (3.1), the projection operator $\mathcal{P}^E : V_{i,\delta}{}_{|E} \mapsto \mathbb{P}_1(E)$ is defined for any function $\phi \in V_{i,\delta}{}_{|E}$ by

$$\begin{cases} (\mathbf{K}_i \nabla \mathcal{P}^E \phi, \nabla p)_E &= (\mathbf{K}_i \nabla \phi, \nabla p)_E \quad \forall p \in \mathbb{P}_1(E) \\ \sum_{k=1}^{n_E} \mathcal{P}^E \phi(\mathbf{x}_k) &= \sum_{k=1}^{n_E} \phi(\mathbf{x}_k) \end{cases} \quad (3.3)$$

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

REMARK 3.1. *Let us observe that the definition* (3.1) *for the bilinear form and* (3.3) *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* (3.3) *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:*

$$c_0(\mathbf{K}_i \nabla\varphi, \nabla\varphi)_E \le S^E(\varphi,\varphi) \le c_1(\mathbf{K}_i \nabla\varphi, \nabla\varphi)_E, \quad \forall \varphi \in V_{i,\delta}{}_{|E}, \text{ with } \mathcal{P}^E \varphi = 0. \quad (3.4)$$

*On each element* $E$ *of the triangulation we have:*

$$\begin{aligned} a^E_i(\phi,\varphi) = a^E_i(\mathcal{P}^E\phi, \mathcal{P}^E\varphi) + a^E_i(\phi - \mathcal{P}^E\phi, \varphi - \mathcal{P}^E\varphi) \\ + \alpha\left(\phi - \mathcal{P}^E\phi, \mathcal{P}^E\varphi\right)_{\mathcal{S}_i \cap \partial E} + \alpha\left(\varphi - \mathcal{P}^E\varphi, \mathcal{P}^E\phi\right)_{\mathcal{S}_i \cap \partial E} \end{aligned} \quad (3.5)$$

*that replaces equation (4.22) of [3].*

*It is possible to show that the given definition of the bilinear form is consistent and stable. Consistency easily follows from definition* (3.1) *and from* (3.3)*: for all* $E \in \mathcal{T}_{i,\delta}$, $\forall p \in \mathbb{P}_1(E)$, $\forall \phi \in V_{i,\delta}{}_{|E}$ *we have:*

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

*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* (3.4) *and* (3.5).

Assuming basic quality properties for the triangulation, functional $S^E$ can be chosen as in [3], thus satisfying conditions (3.4): for all $\phi, \varphi \in V_{i,\delta|E}$ we set

$$S^E(\phi,\varphi) = \sum_{k=1}^{n_E} \mathbf{K}_i(\phi(\mathbf{x}_k) - (\mathcal{P}^E\,\phi)(\mathbf{x}_k))(\varphi(\mathbf{x}_k) - (\mathcal{P}^E\,\varphi)(\mathbf{x}_k)). \tag{3.6}$$

Concerning the treatment of the source term $q_i$ at right hand side of equation (2.6), it is shown in [4] that convergence rates are preserved by approximating $q_i$ with a piecewise constant function on each element of the mesh.

Given the previous results and definitions, it is possible to use the convergence theorem in [3] to prove that the discrete problems on the fractures are well posed and convergence rates are equal to those of standard finite elements of the same order.

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

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

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

**4. Formulation and resolution of the discrete problem.** As shown in Section 2, the problem has been reformulated as a PDE-constrained optimization problem (see equation (2.7)) in which the quadratic functional $J$ is to be minimized subject to linear constraints. In this section, following a first-discretize-then-optimize approach, we give some details about the discrete formulation of the problem and the numerical approach for computing a solution to the problem. In the following, we will use lower case letters for the finite dimensional approximations of functions $H$ and $U$.

**4.1. Discrete formulation.** As outlined in the previous section, we introduce a finite dimensional basis for each fracture $F_i$, with a total number $N^F = \sum_{i=1}^{I} N_i$ of DOFs on the fractures. Concerning the functional space on the traces, in order to simplify the discussion, we consider the following different numbering for the control functions $u_i^S$, induced by the trace numbering. Being $S = S_m$ a given trace, with $I_{S_m} = \{i, j\}$ and assuming $i < j$, we denote by $u_m^-$ and by $u_m^+$ the control functions related to the $m$-th trace and corresponding to fractures $F_i$ and $F_j$, respectively. By overloading the notation, we use the same symbol for the corresponding vector of DOFs. Let us introduce basis functions $\psi_{m,k}^-$, $k = 1, ..., N_m^-$ and $\psi_{m,k}^+$, $k = 1, ..., N_m^+$ for the space of the control function $u_m^-$ and $u_m^+$, respectively. Note that here we allow to use different spaces on the two "sides" of each trace. Then we have, for $m = 1, ..., M$, $\star = -, +$, $u_m^\star = \sum_{k=1}^{N_m^\star} u_{m,k}^\star \psi_{m,k}^\star$. Setting $N^T = \sum_{m=1}^{M}(N_m^- + N_m^+)$, we define $u \in \mathbb{R}^{N^T}$ concatenating $u_1^-, u_1^+, \ldots, u_M^-, u_M^+$.

Let us consider the functional $J$, whose expression is given in Section 2 by equation (2.4), and let us write the discrete functional in terms of $\mathrm{L}^2$ norms instead of $\mathrm{H}^{-\frac{1}{2}}$

and $\mathrm{H}^{\frac{1}{2}}$ norms on the traces: its discrete counterpart is

$$J = \frac{1}{2}\sum_{i=1}^{I}\sum_{S\in\mathcal{S}_i}\left(\int_S (\sum_{k=1}^{N_i} h_{i,k}\varphi_{i,k_{|S}} - \sum_{k=1}^{N_j} h_{j,k}\varphi_{j,k_{|S}})^2\,\mathrm{d}\gamma + \right. \tag{4.1}$$
$$\left.\int_S (\sum_{k=1}^{N_m^-} u^-_{m,k}\psi^-_{m,k} + \sum_{k=1}^{N_m^+} u^+_{m,k}\psi^+_{m,k} - \alpha\sum_{k=1}^{N_i} h_{i,k}\varphi_{i,k_{|S}} - \alpha\sum_{k=1}^{N_j} h_{j,k}\varphi_{j,k_{|S}})^2\,\mathrm{d}\gamma\right).$$

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

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

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

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

and $B^\star_{i,m} \in \mathbb{R}^{N_i\times N^\star_m}$ and $B^\star_{j,m} \in \mathbb{R}^{N_j\times N^\star_m}$ as

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

The functional $J$ in (4.1) is therefore written, in algebraic form, as

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

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

$$B = \begin{pmatrix} B_1R_1 \\ \vdots \\ B_IR_I \end{pmatrix}.$$

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 $G^u = \mathrm{diag}(\mathcal{C}_m, m = 1, ..., M)$. With these definitions at hand, the functional $J$ is rewritten

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

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

We finally note that, setting

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

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

Constraints (2.6) are written as a unique linear system as follows: For all $i = 1, ..., I$ define the matrix $A_i \in \mathbb{R}^{N_i \times N_i}$ as

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

where the operators $\mathcal{P}^E$ and $S^E$ are defined by (3.3) and (3.6), respectively.

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

The algebraic formulation of the primal equations (2.6) is then

$$A_i h_i = \tilde{q}_i + \mathcal{B}_i \, u'_i, \qquad i = 1, ..., I, \tag{4.2}$$

where $\tilde{q}_i$ accounts for the term $q_i$ in (2.6) and for the boundary conditions on the fracture $F_i$.

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

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

Setting $q = (\tilde{q}_1, \ldots, \tilde{q}_I) \in \mathbb{R}^{N^F}$, constraints (4.2) are then written $Ah - \mathcal{B}\, u = q$.

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

$$\min\ J(h,u) = \frac{1}{2}\left(h^T G^h h - \alpha h^T B u - \alpha u^T B^T h + u^T G^u u\right) \tag{4.3}$$

$$\text{s.t.} \quad Ah - \mathcal{B}\, u = q. \tag{4.4}$$

**4.2. Solving the optimization problem.** The first order optimality conditions for problem (4.3)-(4.4) are the following:

$$\begin{pmatrix} G^h & -\alpha B & A^T \\ -\alpha B^T & G^u & -\mathcal{B}^T \\ A & -\mathcal{B} & 0 \end{pmatrix} \begin{pmatrix} h \\ u \\ -p \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ q \end{pmatrix} \tag{4.5}$$

being $p$ the vector of Lagrange multipliers.

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

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

$$h = A^{-1}(\mathcal{B}\, u + q), \tag{4.6}$$

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 (4.6), 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

$$A^T p = G^h h - \alpha B u. \tag{4.7}$$

With these definitions, we may write

$$\nabla \hat{J}(u) = \mathcal{B}^T p + G^u u - \alpha B^T h. \tag{4.8}$$

Note that setting to zero the previous expression for obtaining stationary points for $\hat{J}(u)$, and collecting such equation together with (4.6) and (4.7), we obtain system (4.5).

Concerning the numerical solution of the optimization problem, we mention here two possible approaches. The first one consists in solving the linear system (4.5). An iterative solver is clearly a recommended choice, and `symmlq` [14] would be a suitable choice; this approach has been used in [6]. 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 (4.6) and (4.7) and $g_0$ by (4.8)
3. Set $d_0 = -g_0$, $k = 0$
4. While $g_k \neq 0$
   - 4.1. Compute $\lambda_k$ 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 $\lambda_k$ in Step 4.1 can be computed via an exact line search. Given a descent direction $d_k$, we compute $\lambda_k$ such that it minimizes the function $\phi(\lambda) := \hat{J}(u_k + \lambda d_k)$. Straightforward computations show that one has

$$\lambda_k = -\frac{d_k^T g_k}{d_k^T \hat{G} d_k}. \tag{4.9}$$

The stepsize $\lambda_k$ 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.

**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 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.

**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 5.1 and 5.2 that represent the two phases of the process described above. In these pictures, coloured elements are the new virtual elements, whereas blank elements are the original triangular elements. Elements with up to 6 edges are introduced in these examples. In the figures, each color corresponds to a different number of edges in the element. The reader might refer to the PDF file to zoom in the pictures for a more detailed view.

The polygonal mesh obtained with the procedure described is possibly improved through the displacement of some nodes. Namely, when a node falls very close to a trace, it can be moved onto the trace itself, and therefore reducing the number of element edges and total degrees of freedom. The mesh improvement process is performed as detailed in the following. The distance of each node of intersected elements from the nearest trace is compared to a given mesh dependent tolerance. If the distance of the node to the closest trace is below the tolerance, then the node is moved to its projection on the trace. Vertices of the fractures always remain fixed and nodes in the border are only moved provided that they remain on the same border in order to avoid changing the 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 5.3.

Since VEM basis functions are not known in the interior of mesh elements in general, we resort to the following mesh-dependent $\mathrm{L}^2$ and $\mathrm{H}^1$ 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.

**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.

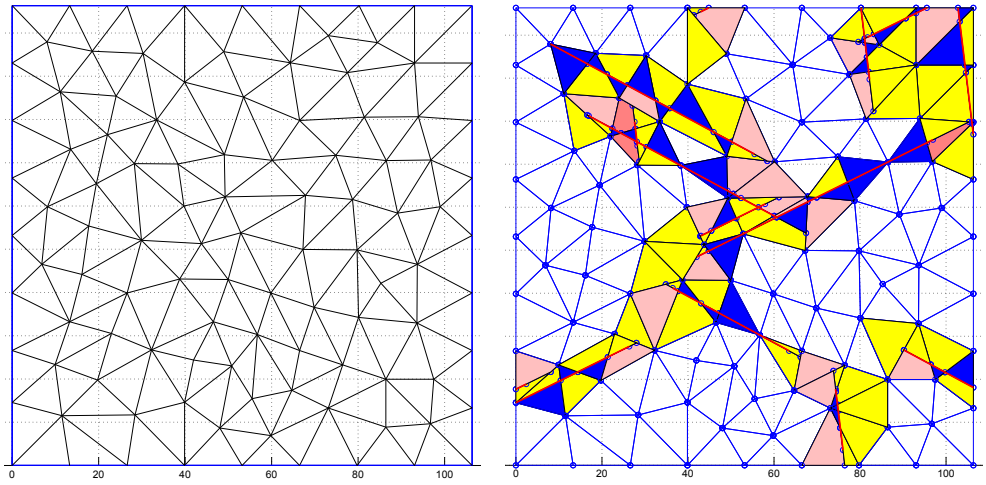


FIGURE 5.1. *Mesh example. Left: original triangulation. Right: mesh for VEM.*

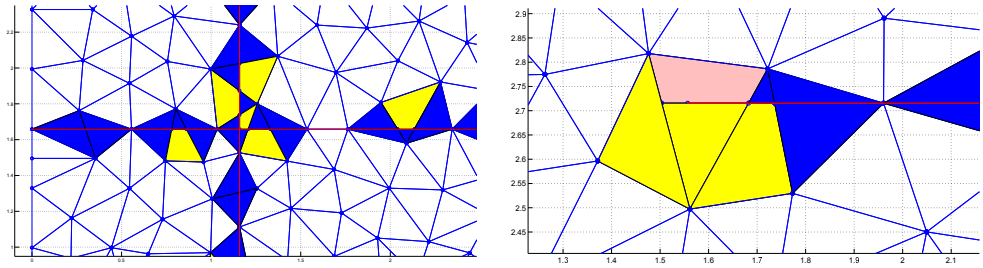


FIGURE 5.2. *Left: detail of a mesh around a trace intersection. Right: detail of a mesh around a trace tip.*

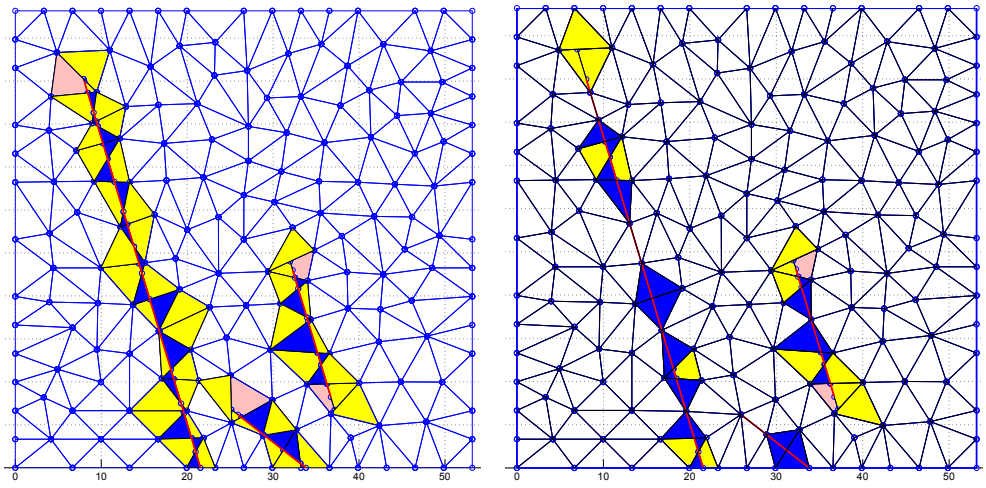


FIGURE 5.3. *Left: example of VEM mesh without modification. Right: Same mesh after modifications.*

In these test cases, a single problem of the form (2.1) is solved, i.e. a single fracture $F$ is considered, assigning $u$ on the traces. In the first case, two intersecting traces are present in $F$, completely crossing the domain, while a single trace ending inside the domain is studied in the second problem. The proposed numerical results show very good approximation capabilities of virtual elements in dealing with these geometrical configurations.

**5.2.1. Problem 1.** The first test problem, labeled P1, displays two traces intersecting each other inside the domain. The domain is a single rectangular fracture $F \subset \mathbb{R}^2$ with two traces $S_1$ and $S_2$ defined by:

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

The domain is shown in Figure 5.4 with a coarse mesh with parameter $\delta_{max} = 0.2$ along with a detail of trace intersection. Here and in the sequel $\delta_{max}$ denotes the square root of the maximum element area for the initial triangulation on each fracture. For this mesh, the original triangular element containing trace intersection is split into four new elements, two triangles and two quadrilaterals.

The problem is set as follows:

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

with

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

where $A_1$, $A_2$, $A_3$ and $A_4$ denote the four regions in which $F$ is divided by the traces, as indicated in Figure 5.4. Values of $f_{S_1}$ and $f_{S_2}$ are

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

and

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

In Figure 5.6, 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

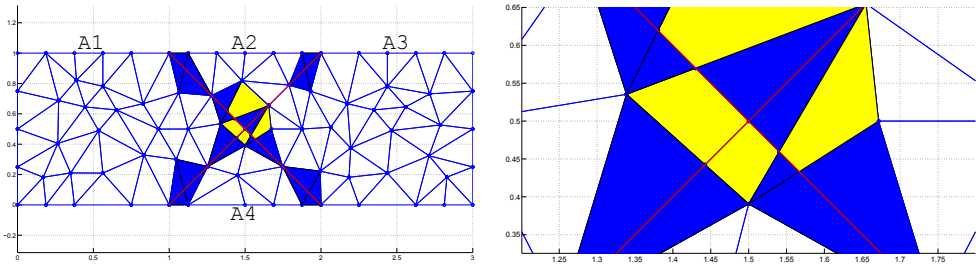


FIGURE 5.4. *Problem P1. Left: Domain with coarse grid $\delta_{max} = 0.2$. Right: a detail of trace intersection.*

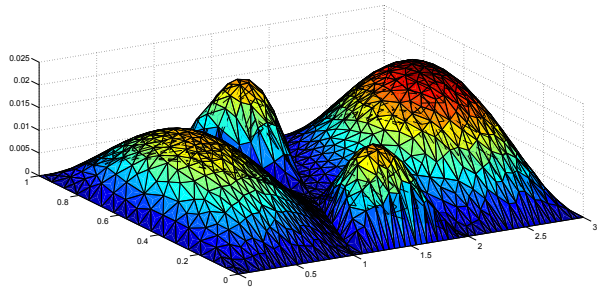

FIGURE 5.5. *Problem P1: approximate solution on a mesh with $\delta_{max} = 0.05$*

the XFEM for the space discretization, as described in [5, 6, 7]. Figure 5.6, 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 [6, 7] and references therein), resulting the two methods in a different number of DOFs, when the same mesh parameter is used. Computed convergence rates are close to the expected ones both in the $L^2$ and the $H^1$ mesh-dependent norms, and both for the VEM and for the XFEM: namely, $L^2$ norm convergence rate is 1.03 for the VEM and 0.99 for the XFEM, whereas the $H^1$ norm convergence rate is 0.49 both for the VEM and for the XFEM. The $L^2$ norm of the error on the restriction of the solution to the traces is also reported (label 'L2H on trace' in the legend), and displays a convergence rate of 1.0 for the VEM and 0.91 for the XFEM. As a whole, the two space discretizations yield a comparable level of accuracy, and the intersection between traces is easily handled by the VEM on a polygonal mesh with very good approximation properties.

**5.2.2. Problem 2.** Let us define the domain $F$ for the second test problem $P2$ as

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

with a single trace $S = \left\{(x, y) \in \mathbb{R}^2 : y = 0 \text{ and } -1 \leq x \leq 0\right\}$ ending in the interior of $F$. This test problem has also been considered in [6]. 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:

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

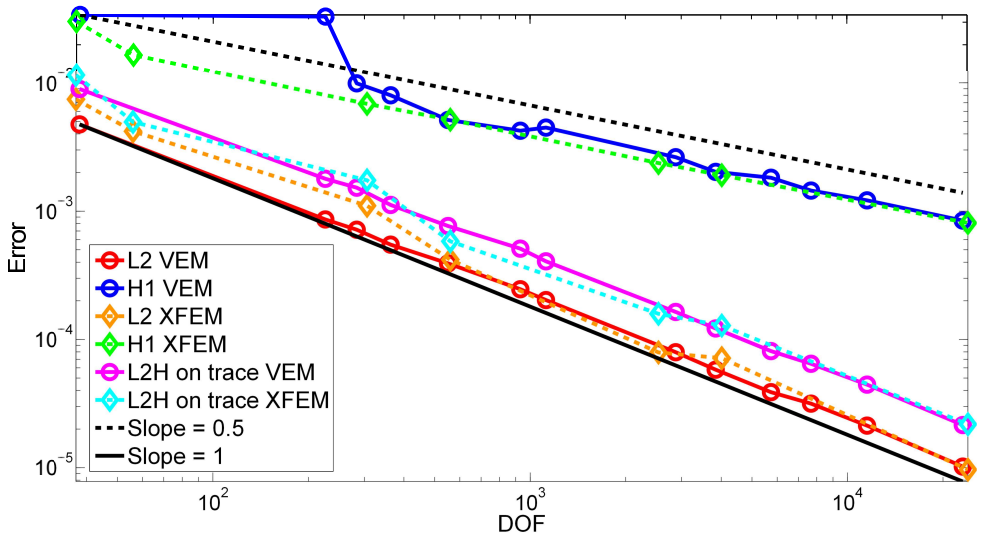


FIGURE 5.6. *Problem P1: error behaviour*

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 problem is defined by the system:

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

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

Figure 5.7 shows the VEM mesh and the resulting elements near the tip. In this implementation of the method, the tip becomes a new node of the triangulation, and three new four-sided elements are generated. Two of them are obtained from the original triangle that contained the trace tip, while the third one appears when the node given by the intersection between the prolonged trace and the opposite mesh element is added to the corresponding neighbouring triangle that becomes a quadrilateral.

The approximate solution is shown in Figure 5.8. In Figure 5.9 we report errors computed both with the $L^2$ and with the $H^1$ mesh dependent norms, both for the VEM and for the XFEM. Computed convergence rates are, also for this test problem, quite similar for the two space discretizations: 1.05 in the $L^2$ norm, and 0.51 in the $H^1$ norm for the VEM; 1.02 in the $L^2$ norm, and 0.47 in the $H^1$ norm for the XFEM. The Figure also reports the errors on the restriction of $H$ to the trace $S$, computed in the $L^2$ norm. Computed convergence rate are in this case 0.85 for the VEM and 0.96 for the XFEM. As for problem P1, the approximation properties of the two space discretizations are therefore quite similar. As a whole, also this geometrical configuration including a trace 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.

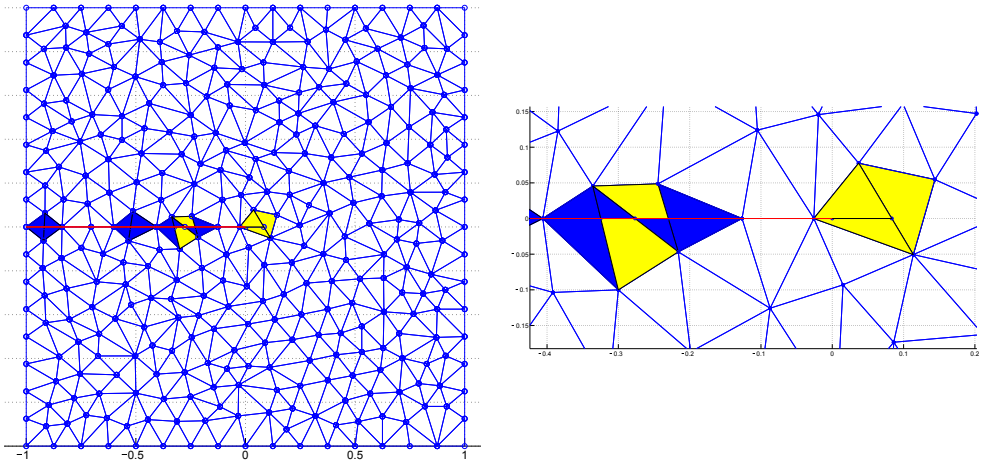

FIGURE 5.7. *Problem P2. Domain meshed with* $\delta_{max} = 0.1$. *Right: a detail of elements near trace tip.*

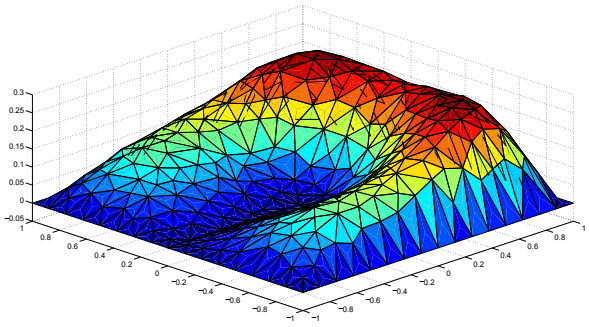

FIGURE 5.8. *Problem P2: approximate solution with VEM obtained with a mesh with* $\delta_{max} = 0.1$

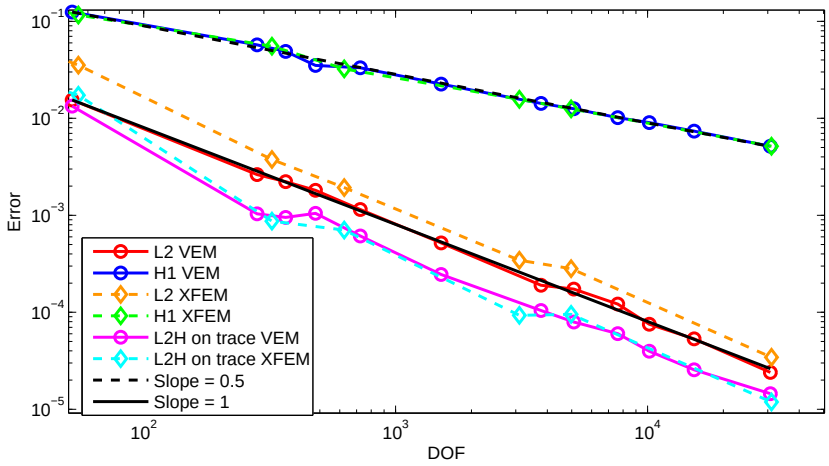


FIGURE 5.9. *Problem P2: error behaviour*

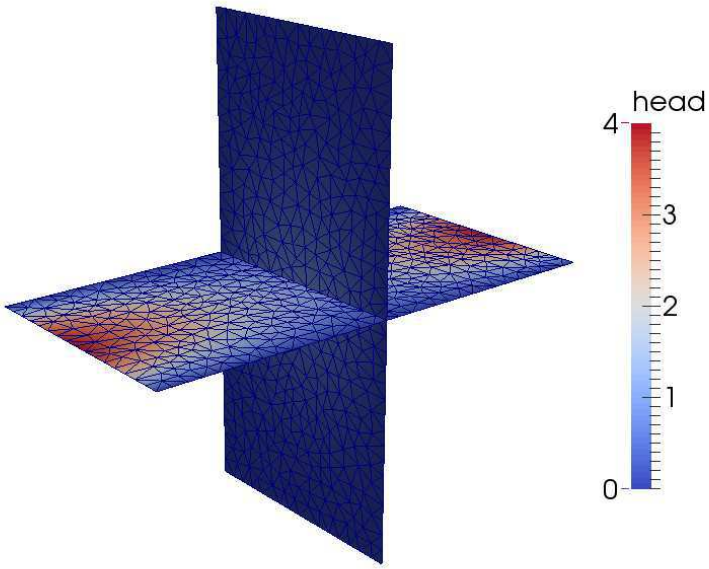


FIGURE 5.10. *DFN2: spatial distribution of fractures and the obtained solution for the hydraulic head.*

**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:

$$\begin{aligned} -\Delta H &= q && \Omega \setminus \mathcal{S}, \\ H_{|\Gamma_D} &= H^D && \text{on } \Gamma_D, \\ \frac{\partial H}{\partial \hat{\nu}} &= G^N && \text{on } \Gamma_N, \end{aligned} \tag{5.1}$$

with reference to the nomenclature introduced in Section 2.

**5.3.1. DFN2.** Here we analyze a very simple DNF consisting of two identical fractures that intersect each other orthogonally, as can be seen in Figure 5.10 where the domain $\Omega$ is depicted.

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

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

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

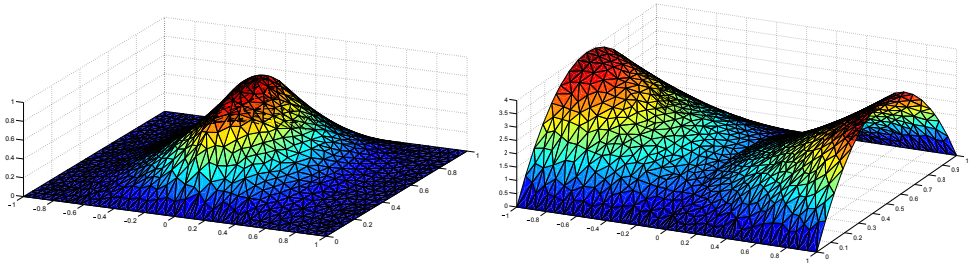

FIGURE 5.11. *DFN2: approximate solution for fracture 1 (left) and fracture 2 (right).*

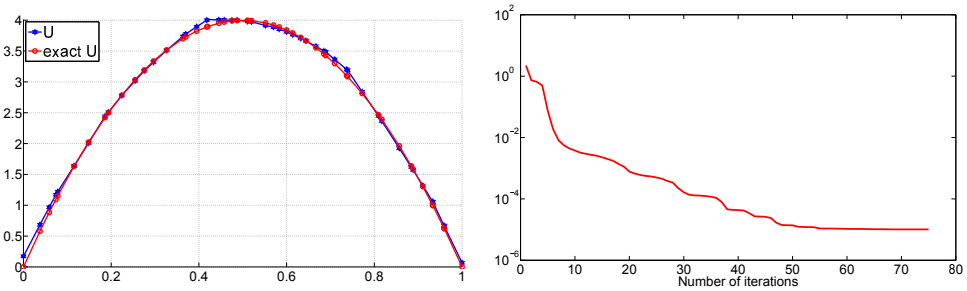


FIGURE 5.12. *DFN2. Left: comparison between exact and approximate flux. Right: values of $J$ versus number of iterations.*

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 5.11 we present the results obtained for the hydraulic head on fracture $F_1$ (left) and $F_2$ (right) using a mesh size $\delta_{max} = \sqrt{0.002}$. Figure 5.12 shows the comparison of the obtained flux with the exact solution and the trend of the minimization of functional $J$ against iteration number. Here, we have performed a number of iterations large enough to let $J$ reach stagnation at its minimum. The computed flux relative to the minimum of the functional approximates the exact solution well.

Error norms are computed for the solution on the fractures in terms of the mesh-dependent $\mathrm{L}^2$ and $\mathrm{H}^1$ norms and are shown in Figure 5.13 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 $\mathrm{L}^2$ and $\mathrm{H}^1$ mesh dependent norms respectively, while a slope of 0.91 is shown for the $\mathrm{L}^2$ error norm relative to the flux and a slope of 0.94 for the $\mathrm{L}^2$ 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.

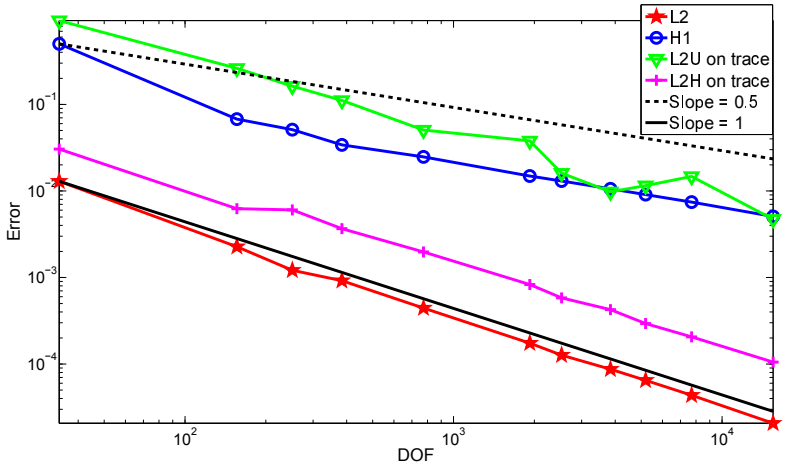


FIGURE 5.13. *DFN2: error behaviour*

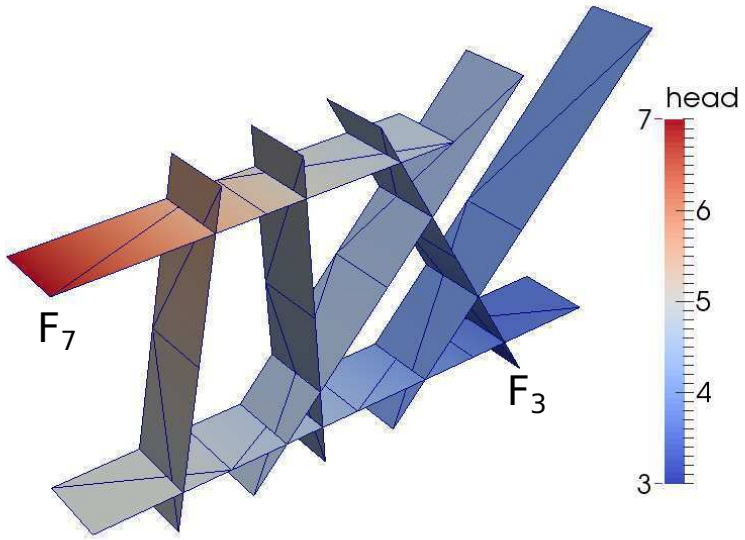


FIGURE 5.14. *DFN7: spatial distribution of fractures and the obtained solution for the hydraulic head.*

**5.3.2. DFN7.** This problem consists of 7 fractures intersecting in 11 traces. The spatial distribution of the fractures can be seen in Figure 5.14. The source term is $q = 0$ in equation (5.1).

The Dirichlet boundary $\Gamma_D$ 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 5.14) and $H^D = 7$ is set on one edge of fracture $F_7$. On all the remaining boundaries of the network we set homogeneous Neumann conditions.

Due to the disposition of the fractures and the boundary conditions, the exact solution to this DFN problem is piecewise affine and displays a slope change at each trace (the jump in the slope corresponding to flux exchange). In this problem we show

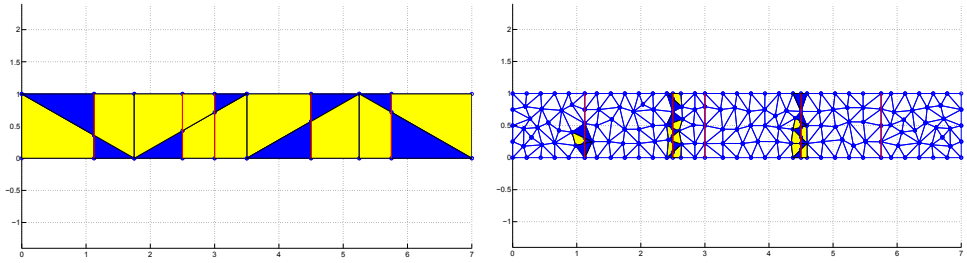

FIGURE 5.15. *DFN7: mesh on $F_6$ with parameter $\delta_{max} = 1.2$ (left) and finer mesh with $\delta_{max} = 0.2$ (right).*

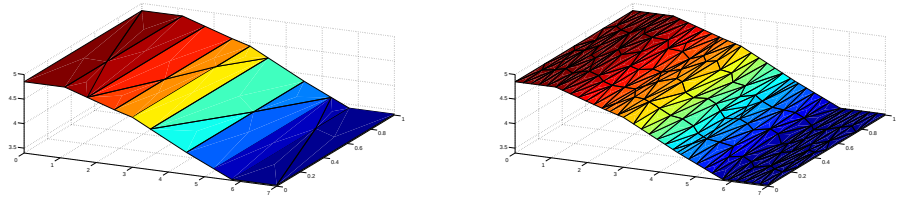

FIGURE 5.16. *DFN7: solutions obtained for fracture 6 with coarse (left) and fine (right) mesh.*

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 5.15 for a detail of the meshes for fracture 3.

Table 5.1 details the flux exchange in fractures and traces for the solution on the finer mesh. Rows correspond to traces and columns to fractures. The last row contains the sum of all the incoming and outgoing flow for each fracture, while the last column shows the balance in flux exchange between the two fractures that share a trace. An almost perfect balancing of the fluxes can be seen, both within fractures and in trace exchanges. Fracture $F_7$ acts as a source that provides 0.7505 of flux to the system (negative values represent flux leaving the fracture), which leaves the system at fracture $F_3$ with an approximately 0 unbalance reported in the bottom-right cell of the table. All other fractures show a quasi non-existent net flow, which agrees with the homogeneous Neumann boundary condition.

**5.3.3. DFN36.** We end the section with a realistic (though rather small) DFN consisting of 36 fractures intersecting in 65 traces. The spatial distribution of the fractures can be seen in Figure 5.17. Assuming meters as unit of length, fracture size spans from $2.8 \times 10^3 \mathrm{m}^2$ to $1.2 \times 10^4 \mathrm{m}^2$.

The Dirichlet boundary is composed by two edges of two fractures, namely $\Gamma_D$ is composed by the borders of fracture $F_1$ and $F_2$ indicated in Figure 5.17, prescribing constant value Dirichlet conditions, $H_1^D = 100$ and $H_2^D = 0$. Homogeneous Neumann boundary conditions are set on all the remaining boundaries. With these boundary conditions fracture $F_1$ is a source of hydraulic head, $F_2$ is a sink fracture and all other fractures are insulated. Also in this case we set $q = 0$ in (5.1).

The problem is solved on several meshes, with $2\mathrm{m}^2 < \delta^2_{\max} < 50\mathrm{m}^2$. In Figure 5.18 the detail of a mesh with $\delta^2_{\max} = 30\mathrm{m}^2$ on a selected fracture and the corresponding obtained solution are shown.

TABLE 5.1

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

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

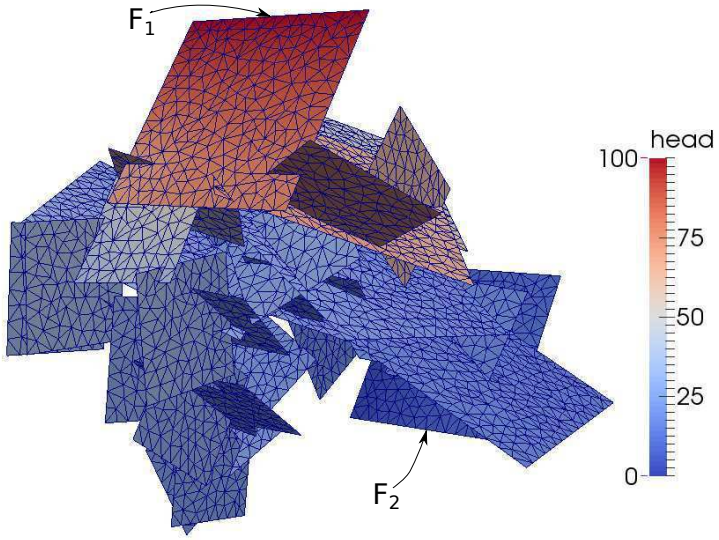


FIGURE 5.17. *DFN36: Spatial distribution of fractures and the obtained solution for the hydraulic head.*

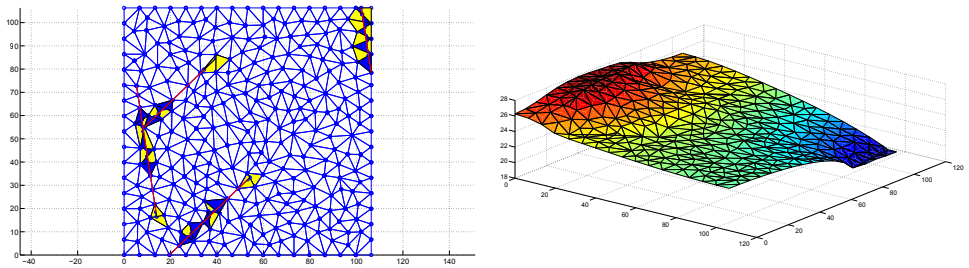

FIGURE 5.18. *DFN36: Left: Mesh with maximum element size of* $30\text{m}^2$ *on a selected fracture. Right: Solution on the same grid.*

TABLE 5.2
*DFN36: $\Delta_{\text{cont}}$ and $\Delta_{\text{flux}}$ for various mesh sizes.*

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

The 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 5.2 for different mesh sizes. Namely, we report values obtained with both the VEM and the XFEM based space discretizations. The table also reports the number of degrees of freedom in the two cases, corresponding to each mesh parameter. We remark that the number of DOFs for $u$ is the same in the two cases, as we use on the traces a finite element discretization which is induced by the intersection points among the initial triangular mesh element edges (the same for the two approaches) and the trace itself. On the other hand, the number of DOFs for $h$ is different for the two approaches here adopted, and is in general smaller for the VEM. This is due to the fact that the XFEM deals with totally non-conforming meshes through the introduction of suitable enrichment functions in triangles close to the traces, thus yielding a bit larger number of DOFs. Note that this larger number of DOFs for the XFEM is required for handling a total non-conforming mesh, but it does not yield more accurate mismatch errors with respect to the VEM approach. As a whole, a good accuracy is obtained with both approaches, and the mismatch errors reduce with mesh refinement.

**6. Conclusions.** The very recent Virtual Element Method is coupled with the optimization based algorithm presented in [5, 6, 7] 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.

REFERENCES


[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. Scialò, *A PDE-constrained optimization formulation for discrete fracture network flows*, SIAM J. Sci. Comput., 35 (2013), pp. B487–B510.

[6] S. Berrone, S. Pieraccini, and S. Scialò, *On simulations of discrete fracture network flows with an optimization-based extended finite element method*, SIAM J. Sci. Comput., 35 (2013), pp. A908–A935.

[7] S. Berrone, S. Pieraccini, and S. Scialò, *An optimization approach for large scale simulations of discrete fracture network flows*, J. Comput. Phys., 256 (2014), pp. 838–853.

[8] F. Brezzi and L. D. Marini, *Virtual element methods for plate bending problems*, Comput. Methods Appl. Mech. Engrg., 253 (2013), pp. 455–462.

[9] 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.

[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. 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] 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.

[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] 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.

[15] 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. 1629–1643.

[16] ———, *A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[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.