Chapter 4

A hybrid mortar method

The contents of this chapter follow closely those found in the article "A hybrid mortar virtual element method for discrete fracture network simulations" published in the *Journal of Computational Physics*, Volume 306, 1 February 2016, Pages 148–166.

4.1 Introduction

In the present chapter, the use of VEM in the DFN framework proposed in chapter 2 is coupled with the well established Mortar Method [18]. A major advantage of this new coupling with respect to previous works that also make use of the primal formulation of the problem is that the flux entering/exiting each fracture from its intersections is directly obtained as part of the solution of the discrete problem and not through a post-processing of the results.

The chapter is organized as follows: in section 4.2 we state the problem setting; in section 4.3 we briefly recall the main features of the VEM needed for the description of our method; section 4.4 is devoted to the description of the hybrid method obtained from coupling the VEM with the mortar method; section 4.5 addresses some implementation issues related to the generation of the locally conforming mesh; finally, section 4.6 reports some numerical results assessing the behavior of the method.

4.2 Problem formulation

A DFN, Ω, is a set of N open planar polygons F_i, $i = 1, \ldots, N$, representing the fractures in the medium. In the sequel, we will identify the fractures with the polygons. Fractures intersect each other along segments called *traces*. We assume throughout the

paper that traces are given by the intersection of exactly two fractures. Whenever two traces intersect each other, we split both traces into two sub-traces. The set of all traces and sub-traces will be denoted by $\mathcal{S}$. For the sake of simplicity, in the sequel we will refer to all elements in $\mathcal{S}$ as traces. For each $S \in \mathcal{S}$, it is convenient to identify the set $\mathcal{I}_S = \{i, j\}$ of the indices of the two fractures intersecting at S. For any function or set defined on the whole DFN, its restriction to fracture F_i will be denoted using the subscript i.

On the domain Ω, we consider the Darcy's law as a model for the equilibrium of the hydraulic head $H = P + \zeta$, where $P = p/(\rho\, g)$ is the fluid pressure, g is the gravitational constant, ρ the fluid density and ζ its elevation. We introduce on each fracture the transmissivity K_i, which is assumed, for the sake of simplicity, to be a scalar function of the local tangential coordinates system on F_i. Let Γ^D be a non-empty portion of $\partial\Omega$ on which the Dirichlet boundary condition H^D is imposed, and let us set $\Gamma_i^D = \Gamma^D \cap F_i$. Note that Γ_i^D is allowed to be empty for some i. Let us assume that $H_i^D \in \mathrm{H}^{\frac{1}{2}}(\Gamma_i^D)$ for all $i \in \{1, \dots, N\}$. Furthermore, let $\Gamma_i^N = \partial F_i \setminus \Gamma_i^D$ be the local Neumann boundary and let $H_i^N \in \mathrm{H}^{-\frac{1}{2}}(\Gamma_i^N)$ be the Neumann boundary condition imposed therein.

Let us define the following functional spaces:

$$
\begin{aligned}
V_i &= \left\{ v \in \mathrm{H}^1(F_i) \colon\ \gamma_{\Gamma_i^D}(v) = 0 \right\} \quad \forall i = 1, \dots, N\,, \\
V_i^D &= \left\{ v \in \mathrm{H}^1(F_i) \colon\ \gamma_{\Gamma_i^D}(v) = H_i^D \right\} \quad \forall i = 1, \dots, N\,, \\
V^D &= \prod_{i=1}^N V_i^D\,, \qquad V = \prod_{i=1}^N V_i\,,
\end{aligned}
$$

where $\gamma_{\Gamma_i^D} : \mathrm{H}^1(F_i) \mapsto \mathrm{H}^{\frac{1}{2}}(\Gamma_i^D)$ is the trace operator on Γ_i^D.

The problem of interest is to find $H \in V^D$ such that $H = H_0 + R^D$ where $R_i^D = R^D\big|_{F_i}$ is a lifting of H_i^D on $\mathrm{H}^1(F_i)$ and $H_0 \in V$ satisfies, for any given $v \in V$ and any $i = 1, \dots, N$,

$$
\begin{aligned}
(K_i \nabla H_{0,i}, \nabla v_i)_{F_i} - \sum_{S \in \mathcal{S}} \left\langle \left[\!\left[K_i \frac{\partial H_i}{\partial \mathbf{n}_S^i} \right]\!\right]_S, v_i \right\rangle_{\pm\frac{1}{2}, S} &= (f_i, v_i)_{F_i} + \left\langle H_i^N, v_i \right\rangle_{\pm\frac{1}{2}, \Gamma_i^N} \\
&\quad - \left(K_i \nabla R_i^D, \nabla v_i\right)_{F_i} \qquad (4.2.1)
\end{aligned}
$$

where $\langle \cdot, \cdot \rangle_{\pm\alpha,\omega}$ is the duality product between $\mathrm{H}^{-\alpha}(\omega)$ and $\mathrm{H}^{\alpha}(\omega)$, $\mathbf{n}_S^i$ is the unit vector normal to trace S on fracture F_i, and the symbol $\left[\!\left[K_i \frac{\partial H_i}{\partial \mathbf{n}_S^i} \right]\!\right]_S$ denotes the jump of the co-normal derivative of H_i across S on F_i. The equations on each fracture are coupled

by the balance of fluxes on traces:

$$\forall S \in \mathcal{S}, \text{ if } \mathcal{I}_S = \{i,j\}, \quad \left[\!\left[K_i \frac{\partial H_i}{\partial \mathbf{n}_S^i} \right]\!\right]_S + \left[\!\left[K_j \frac{\partial H_j}{\partial \mathbf{n}_S^j} \right]\!\right]_S = 0\,, \tag{4.2.2}$$

and by the continuity of the solution across traces, that can be written as:

$$\forall S \in \mathcal{S}, \forall \psi \in \mathrm{H}^{-\frac{1}{2}}(S) = \left(\mathrm{H}_{00}^{\frac{1}{2}}(S)\right)', \langle [\![H]\!]_S, \psi \rangle_{\pm\frac{1}{2},S} = 0\,. \tag{4.2.3}$$

We introduce

$$\forall S \in \mathcal{S}, \forall \psi \in \mathrm{H}^{-\frac{1}{2}}(S), \quad b_S(v,\psi) = \langle [\![v]\!]_S, \psi \rangle_{\pm\frac{1}{2},S}\,, \tag{4.2.4}$$

and rewrite (4.2.3) as

$$\forall S \in \mathcal{S}, \forall \psi \in \mathrm{H}^{-\frac{1}{2}}(S), \quad b_S(H,\psi) = 0\,. \tag{4.2.5}$$

For the sake of convenience in rewriting the jump of a function on a trace S, let us fix the following sign convention: for each $S \in \mathcal{S}$, with $\mathcal{I}_S = \{i,j\}$, let us introduce the function $m_S : \mathcal{I}_S \to \{0,1\}$ as follows:

$$m_S(k) = \begin{cases} 1 & \text{if} \quad k = \min\{i,j\} \\ 0 & \text{otherwhise} \end{cases}.$$

Hence, we may write

$$\forall v \in V, \forall S \in \mathcal{S}, \; [\![v]\!]_S = \sum_{i \in \mathcal{I}_S} (-1)^{m_S(i)}\, \gamma_s(v_i)\ . \tag{4.2.6}$$

We introduce the space $M = \prod_{S\in\mathcal{S}} \mathrm{H}^{-\frac{1}{2}}(S)$ and set

$$\forall v_i \in V_i, \forall \psi \in M, \quad b_i(v_i,\psi) = \sum_{S \in \mathcal{S}_i} (-1)^{m_S(i)} \langle \gamma_S(v_i), \psi_S \rangle_{\pm\frac{1}{2},S}\,. \tag{4.2.7}$$

With these definitions at hand, we define $\Lambda \in M$ such that, $\forall S \in \mathcal{S}$,

$$\Lambda_S = \left[\!\left[K_i \frac{\partial H_i}{\partial \mathbf{n}_S^i} \right]\!\right]_S\,,$$

where i is such that $m_S(i) = 1$. Then, defining $a_i : V_i \times V_i \mapsto \mathbb{R}$ as

$$a_i(u_i,v_i) = (K_i \nabla u_i, \nabla v_i)_{F_i} \quad \forall i = 1,\ldots,N\,, \tag{4.2.8}$$

equation (4.2.1) can be written as

$$a_i\left(H_{0i}, v_i\right) + b_i\left(v_i, \Lambda\right) = \left(f_i, v_i\right)_{F_i} + \left\langle H_i^N, v_i\right\rangle_{\pm\frac{1}{2},\Gamma_i^N} - a_i\left(R_i^D, v_i\right) \quad \forall v \in V\,. \tag{4.2.9}$$

In view of a global formulation of the problem, we define

$$a\left(u, v\right) = \sum_{i=1}^{N} a_i\left(u_i, v_i\right) \quad \forall u, v \in V\,, \tag{4.2.10}$$

$$b\left(v, \psi\right) = \sum_{i=1}^{N} b_i\left(v_i, \psi\right) \quad \forall v \in V,\ \psi \in M\,. \tag{4.2.11}$$

Note that due to (4.2.4), (4.2.7) and (4.2.5) we have $b\left(H, \psi\right) = \sum_{S\in\mathcal{S}} b_S\left(H, \psi\right) = 0$. Summing up (4.2.9) over all the fractures we obtain:

$$\begin{cases} a\left(H_0, v\right) + b\left(v, \Lambda\right) = \left(f, v\right) + \left\langle H^N, v\right\rangle_{\pm\frac{1}{2},\Gamma^N} - a\left(R^D, v\right) & \forall v \in V, \\ b\left(H_0, \psi\right) = -b\left(R^D, \psi\right) & \forall \psi \in M. \end{cases} \tag{4.2.12}$$

Let us endow V^D and V with the norm

$$\| v \|_V = \left(\sum_{i=1}^{N} \| v_i \|^2_{\mathrm{L}^2(F_i)} + a_i\left(v_i, v_i\right)\right)^{\frac{1}{2}}. \tag{4.2.13}$$

Well-posedness of problem (4.2.12) follows observing that, introducing the Hilbert space

$$W = \left\{v \in V : \forall S \in \mathcal{S},\, \forall \psi \in \mathrm{H}^{-\frac{1}{2}}(S),\, \langle [\![v]\!]_S, \psi\rangle_{\pm\frac{1}{2},S} = 0\right\} = \ker(b)\,,$$

problem (4.2.12) is equivalent to: *find* $H_0 \in W$ *such that*

$$a\left(H_0, v\right) = \left(f, v\right) + \left\langle H^N, v\right\rangle_{\pm\frac{1}{2},\Gamma^N} - a\left(R^D, v\right) \quad \forall v \in W\,.$$

4.3 The Virtual Element Method

In this section we briefly recall the main features of the conforming VEM which are useful for the description of the approach proposed in the following sections. The reader is referred to the seminal papers [10, 12] for a thorough description and to [2, 6, 14, 15, 32] for further developments of the method.

The VEM is a generalization of the standard finite element method to polygonal meshes, and it includes some of the ideas present in the mimetic difference method [9, 55]. The peculiarity of the method is that the discrete functional space contains more general functions in addition to standard piecewise polynomials, namely, it contains functions

whose restrictions to element edges are polynomials, whereas in the interior only information of the function for certain degrees of freedom is known. When computing the stiffness matrix or the right hand side of the problem, the integrals will be computed exactly only if at least one of the two factors is a polynomial, whereas in other cases they will be substituted by operations on the degrees of freedom suitably defined to maintain the right order of convergence.

Let us consider a given fracture $F_i \subset \mathbb{R}^2$, a mesh $\tau_{\delta,i}$ on F_i with mesh parameter δ, representing the maximum element size, and consisting of a finite number of polygons E, convex in the following, with an arbitrary number of edges. We denote by k the desired order of accuracy of the method and by $\mathbb{P}_k$ the space of the polynomials of maximum order k, with $\mathbb{P}_{-1} = \{0\}$. The local virtual element space $V_{k,\delta}(E)$ is defined as

$$V_{k,\delta}(E) = \big\{ v_\delta \in \mathrm{H}^1(E) : v_{\delta|\partial E} \in C^0(\partial E),\, v_{\delta|e} \in \mathbb{P}_k(e),\, \forall e \subset \partial E, \\ \Delta v_\delta \in \mathbb{P}_{k-2}(E) \big\}$$

where ∂E is the boundary of E, and e is an edge. Note that from the definition it clearly follows that $\mathbb{P}_k(E) \subseteq V_{k,\delta}(E)$; the latter set may also include other non-polynomial functions.

Following [10], for each element E the following set of DOFs is introduced:

- the value of v_δ at the vertices of E;
- the value of v_δ at $k-1$ internal points on each edge of E;
- the scaled moments $\frac{1}{|E|}\int_E v_\delta m_\alpha$ for $|\alpha| \leq k-2$,

where m_α, with $\alpha = (\alpha_1, \alpha_2)$, denotes the scaled monomial

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

being (x_c, y_c) and h_E the centroid and the diameter of the element E, respectively. Edge moments can also be chosen as degrees of freedom instead of internal edge point values; in general, any set of DOFs that completely defines $v_{\delta|e}$ for all edges of the element is a valid choice. Note that for $k = 1$ the set of DOFs is given by the values of v_δ at the vertices of E. The selected set of degrees of freedom is unisolvent [10] and therefore, given an element E with n_v vertices, we have that the dimension of $V_{k,\delta}(E)$ is

$$\dim V_{k,\delta}(E) = n_v k + k(k-1)/2\,.$$

We can define basis functions ϕ_ℓ with $\ell = 1, ..., \dim V_{k,\delta}(E)$ in $V_{k,\delta}(E)$ in such a way that $\mathrm{dof}_\ell(\phi_m) = \delta_{\ell m}$ where $\mathrm{dof}_\ell(v) :=$ *value of* v *at* ℓ*-th degree of freedom.* The global virtual element space on F_i is:

$$V_{k,\delta}(F_i) = \{v_\delta \colon v_\delta|_E \in V_{k,\delta}(E) \text{ for all } E \in \tau_{\delta,i}\} \subset \mathrm{H}^1(F_i) \ .$$

Inclusion in $\mathrm{H}^1(F_i)$ is a consequence of the choice of edges and vertices as DOFs, that guarantees continuity of any function $v_\delta \in V_{k,\delta}(F_i)$ on internal edges of the mesh.

Let us assume for the sake of simplicity that the fracture transmissivity K_i, $i = 1, \ldots, N$, is constant on F_i, and let us introduce, on each element E of F_i, the bilinear form

$$a^E(u, v) = K_i(\nabla u, \nabla v)_E \quad \forall u, v \in V_{k,\delta}(E) \, .$$

For $k \geq 1$, let us introduce a projection operator on E:

$$\Pi^\nabla_{E,k} : V_{k,\delta}(E) \longrightarrow \mathbb{P}_k(E) \, ,$$

defined by

$$\int_E \nabla p_k \cdot (\nabla v_\delta - \nabla \Pi^\nabla_{E,k} v_\delta) = 0 \quad \forall p_k \in \mathbb{P}_k(E) \, , \tag{4.3.1}$$

$$\begin{cases} \displaystyle\int_E \Pi^\nabla_{E,k} v_\delta = \int_E v_\delta & k > 1, \\ \displaystyle\sum_{i=1}^{n_v} \Pi^\nabla_{E,k} v_\delta(\mathcal{V}_i) = \sum_{i=1}^{n_v} v_\delta(\mathcal{V}_i) & k = 1, \end{cases} \tag{4.3.2}$$

where $\mathcal{V}_i$ are the vertices of the element.

Note that

$$\Pi^\nabla_{E,k} p_k = p_k \text{ for all } p_k \in \mathbb{P}_k(E) \, .$$

The projection represents an orthogonality condition in the scalar product induced by the bilinear form a. Thanks to integration by parts, the computation of $\Pi^\nabla_{E,k} v_\delta$ can be performed just by exploiting the knowledge of v_δ in the degrees of freedom [12]. Equation (4.3.1) completely determines the gradient of the projection, while (4.3.2) takes care of the constant part. Other options for (4.3.2) exist [10, 14].

Remark 4.1. The assumption of K_i being constant on each fracture is made here for the ease of description. In case of problems presenting non-constant coefficients or a more general second order differential equation including lower order terms, other projectors have to be used in order to retain optimal convergence [14].

Let us now introduce the discrete bilinear form on the element E as:

$$a_\delta^E(u_\delta, v_\delta) = a^E\left(\Pi_{E,k}^\nabla u_\delta, \Pi_{E,k}^\nabla v_\delta\right) + S_\delta^E\left(u_\delta - \Pi_{E,k}^\nabla u_\delta, v_\delta - \Pi_{E,k}^\nabla v_\delta\right), \quad \forall u_\delta, v_\delta \in V_{k,\delta}(E), \tag{4.3.3}$$

where S_δ^E is any symmetric, positive definite bilinear form that verifies

$$C_0 a^E(v_\delta, v_\delta) \leq S_\delta^E(v_\delta, v_\delta) \leq C_1 a^E(v_\delta, v_\delta) \quad \forall v_\delta \in \ker(\Pi_{E,k}^\nabla),$$

for constants $C_0, C_1 > 0$ independent of E. This means that $S_\delta^E(v, v)$ scales like $a^E(v, v)$ on the kernel of $\Pi_{E,k}^\nabla$. A possible choice for S_δ^E is the Euclidean product in $\mathbb{R}^{\#V_{k,\delta}^E \times \#V_{k,\delta}^E}$ between vectors whose components are the values of the functions at the degrees of freedom. Note that the first term of (4.3.3) ensures the consistency of the form, and the second one has in charge its stability. In particular, we have

$$a_\delta^E(v_\delta, p_k) = a^E(v_\delta, p_k) \quad \forall v_\delta \in V_\delta, \forall p_k \in \mathbb{P}_k(E). \tag{4.3.4}$$

Going back to the whole DFN, the global discrete bilinear form is defined as

$$a_\delta(h_\delta, v_\delta) = \sum_{i=1}^N \sum_{E \in \tau_{\delta,i}} a_\delta^E(h_\delta, v_\delta) \quad \forall h_\delta, v_\delta \in V_{k,\delta} = \prod_{i=1}^N V_{k,\delta}(F_i). \tag{4.3.5}$$

Under proper regularity assumptions, it can be proved [10] that a_δ is equivalent to a, i.e. that there exist two positive constants α^* and α_*, independent of δ and N, such that

$$\alpha_* \, a(v_\delta, v_\delta) \leq a_\delta(v_\delta, v_\delta) \leq \alpha^* \, a(v_\delta, v_\delta) \quad \forall v_\delta \in V_{k,\delta}. \tag{4.3.6}$$

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

$$(f, v_\delta)_\delta = \sum_{E \in \tau_\delta} \int_E f \Pi_{E,k}^0 v_\delta \tag{4.3.7}$$

where $\forall E \in \mathcal{T}_h$, $\Pi_{E,k}^0$ is computed as in [12, sections 5.3–6.1], i.e. it is the polynomial function such that, $\forall v \in V_{k,\delta}(E)$,

$$\left(\Pi_{E,k}^0 v, p\right)_E = (v, p)_E \quad \forall p \in \mathbb{P}_{k-2}(E),$$
$$\left(\Pi_{E,k}^0 v, p\right)_E = \left(\Pi_{E,k}^\nabla v, p\right)_E \quad \forall p \in \mathbb{P}_k(E) \setminus \mathbb{P}_{k-2}(E).$$

4.4 Mortar formulation of the problem

In this section we introduce a Mortar formulation for problem (4.2.12), in conjunction with the VEM for the finite dimensional approximation of V.

From now on, let us consider the VEM of order k and let us introduce the spaces

$$V_\delta = \left\{ v_\delta \in V_{k,\delta} \colon \gamma_{\Gamma_i^D}(v_\delta) = 0 \quad \forall i = 1, \dots, N \right\}, \tag{4.4.1}$$

$$V_\delta^D = \left\{ v_\delta \in V_{k,\delta} \colon \gamma_{\Gamma_i^D}(v_\delta) = \Pi^0_{k,\Gamma^D}(H^D) \quad \forall i = 1, \dots, N \right\}, \tag{4.4.2}$$

where Π^0_{k,Γ^D} is the piecewise $\mathrm{L}^2(e)$ projection on polynomials of degree $\leq k$ for all edges e such that $e \cap \Gamma^D \neq \emptyset$. We equip both spaces with the same norm as V. The Mortar Method [18] consists in weakening the continuity of the solution on each trace $S \in \mathcal{S}$, replacing it by suitable orthogonality conditions with respect to a proper finite dimensional subspace of $\mathrm{H}^{-\frac{1}{2}}(S)$, which will be denoted by $M_{\delta,S} \subset \mathrm{L}^2(S)$. Going back to equation (4.2.9), let h and λ denote the discrete counterpart of H and Λ, respectively; the discrete version of (4.2.9) is written as: find $h \in V_\delta^D$ such that, for $i = 1, \dots, N$

$$a_{\delta i}(h_i, v_{\delta i}) + b_i(v_{\delta i}, \lambda) = (f_i, v_{\delta i})_\delta + \left(H^N, v_{\delta i}\right)_{\Gamma_i^N} \quad \forall v_\delta \in V_\delta \,. \tag{4.4.3}$$

Following the mortar terminology, for each $S \in \mathcal{S}$, $S = \bar{F}_i \cap \bar{F}_j$, we call *mortar fracture* the one whose index i is such that $m_S(i) = 0$, while the other fracture intersecting at S will be denoted as *non-mortar*. In the present context, $\lambda_S = \lambda|_S$ will approximate the jump of the co-normal derivative of the solution on the non-mortar fracture, while the jump on the mortar fracture will be approximated by $-\lambda_S$.

Considering again a lifting R_δ^D of $\Pi^0_{k,\Gamma^D}(H^D)$ and summing up (4.4.3) over all fractures, the global form a_δ defined by (4.3.5) arises and the problem can be rewritten as: find $h = h_0 + R_\delta^D$, with $h_0 \in V_\delta$, and $\lambda \in M_\delta$ such that

$$\begin{cases} a_\delta(h_0, v_\delta) + b(v_\delta, \lambda) = (f, v_\delta)_\delta + \left(H^N, v_\delta\right)_{\Gamma^N} - a_\delta\left(R_\delta^D, v_\delta\right) & \forall v_\delta \in V_\delta, \\ b(h_0, \psi_\delta) = -b\left(R_\delta^D, \psi_\delta\right) & \forall \psi_\delta \in M_\delta, \end{cases} \tag{4.4.4}$$

being

$$M_\delta = \prod_{S \in \mathcal{S}} M_{\delta,S}, \tag{4.4.5}$$

and $b(v_\delta, \psi_\delta)$ computed as an integral in $\mathrm{L}^2(S)$.

4.4.1 Well-posedness of the discrete problem

Following [25, Corollary 2.1], the well-posedness of problem (4.4.4) is guaranteed if a_δ is coercive on

$$W_\delta = \{v_\delta \in V_\delta \colon b(v_\delta, \psi_\delta) = 0 \quad \forall \psi_\delta \in M_\delta\} \,, \tag{4.4.6}$$

and an *inf-sup* condition holds:

$$\exists \beta > 0 \colon \inf_{\psi_\delta \in M_\delta} \sup_{v_\delta \in V_\delta} \frac{b\left(v_\delta, \psi_\delta\right)}{\| v_\delta \|_V \| \psi_\delta \|_M} \geq \beta \,. \tag{4.4.7}$$

The existence of a constant β independent of δ satisfying (4.4.7) was proved in [16] making use of [68, Lemma 10] in the case of a polynomial Finite Element approximation on a regular triangulation. The same proof applies here under the following assumption.

Assumption 1. There exists a constant $\sigma > 0$ independent of δ such that, for each $E \in \tau_{\delta,i}$, for $i = 1, \ldots, N$, the distance between any two vertices of E is larger then or equal to σh_E, where h_E is the diameter of E.

Under this assumption, consider a trace S and a segment e belonging to the discretization of S. Let E be one of the two polygons sharing e. By Assumption 1, we can construct in the interior of E a triangle $T_{e,E}$ having e as one of its edges and having a shape regularity which depends uniquely on σ (for example, for convex elements, by connecting the extrema of e with the barycenter of E). The area of such a triangle scales as the area of E divided by the number of edges of E. We are thus led to make the following assumption.

Assumption 2. The number of edges of the elements of τ_δ is limited independently of δ.

With this last assumption, the area of $T_{e,E}$ scales like the area of E and thus, the norm of any function belonging to the finite dimensional space on $T_{e,E}$ is equivalent to the one on E. From [68, Lemma 10], we obtain the existence of an inf-sup constant independent of δ for $T_{e,E}$ and thus prove the existence of such a constant for E by the equivalence of the norms.

To prove the coercivity of a_δ on W_δ, we first prove the coercivity of a on such space and then use the equivalence (4.3.6). The key result needed is the following.

Proposition 4.4.1. Assume that M_δ contains the functions which are constant on each trace. Then, the functional $v_\delta \mapsto |\!|\!| v_\delta |\!|\!|$ is a norm over W_δ.

Proof. It is enough to verify that $|\!|\!| v_\delta |\!|\!| = 0$ only if $v_\delta = 0$. Let $v_\delta \in W_\delta$ be such that $|\!|\!| v_\delta |\!|\!| = 0$. Then it must be constant on each fracture, since its gradient on each fracture

is null. Furthermore, v_δ clearly vanishes on all fractures such that $\Gamma_i^D \neq \emptyset$. It is now easy to prove that v_δ vanishes on all fractures. Indeed, let S be a trace shared by fractures F_i and F_j, with $\gamma_S(v_{\delta i}) = 0$; thanks to the mortar condition one has

$$(\llbracket v_\delta \rrbracket_S, 1)_S = |S| \llbracket v_\delta \rrbracket_S = 0 \Rightarrow \gamma_S(v_{\delta j}) = \gamma_S(v_{\delta i}) = 0$$

and since $v_{\delta i}$ and $v_{\delta j}$ are constant, it follows that $v_{\delta j} = 0$. Thanks to the network connectivity, this ensures that v_δ vanishes on all the fractures. □

From now on, M_δ is required to satisfy the assumption of Proposition 4.4.1. It follows that a is coercive with coercivity constant 1 on W_δ. By (4.3.6), a_δ is coercive with coercivity constant α_*.

4.4.2 A priori error estimates

We are now able to derive an *a priori* error estimate. To this aim, we introduce the operators $\mathscr{F}, \mathscr{F}_\delta \in V'$ defined such that

$$\langle \mathscr{F}, v \rangle_{\pm 1,\Omega} = (f, v)_\Omega, \qquad \langle \mathscr{F}_\delta, v \rangle_{\pm 1,\Omega} = (f, v)_\delta.$$

Furthermore, define

$$W_\delta^D = \left\{ v \in V_\delta^D : b(v, \psi) = 0, \quad \forall \psi \in M_\delta \right\}, \tag{4.4.8}$$

$$\mathbb{P}_k^D(\Omega) = \left\{ p \in V_\delta^D : \, p \in \mathbb{P}_k(E), \quad \forall E \in \tau_\delta \right\}. \tag{4.4.9}$$

The main result concerning the a priori error estimate is stated in the following Theorem. The proof is reported in 4.8.

Theorem 4.2. *Let V_δ, M_δ, W_δ, W_δ^D and $\mathbb{P}_k^D(\Omega)$ be defined as in (4.4.1), (4.4.5), (4.4.6), (4.4.8) and (4.4.9), respectively. Then, the solution (h, λ) to problem (4.4.4) and the solution (H, Λ) to problem (4.2.9) satisfy*

$$\begin{aligned} \| H - h \| \leq & \left(1 + \frac{\alpha^*}{\alpha_*}\right) \inf_{v_\delta \in W_\delta^D} \| H - v_\delta \| + \frac{1 + \alpha^*}{\alpha_*} \inf_{p_k \in \mathbb{P}_k^D(\Omega)} \| H - p_k \| \\ & + \frac{1}{\alpha_*} \left(\inf_{\psi_\delta \in M_\delta} \sup_{v_\delta \in W_\delta} \frac{b(v_\delta, \Lambda - \psi_\delta)}{\| v_\delta \|} \right) + \frac{1 + C_\Omega}{\alpha_*} \| \mathscr{F} - \mathscr{F}_\delta \|_{V'}. \end{aligned} \tag{4.4.10}$$

Moreover, assume (4.4.7) *is satisfied. Then,*

$$\begin{aligned} \| \Lambda - \lambda \|_M \leq \left(1 + \frac{1}{\beta}\right) \inf_{\psi_\delta \in M_\delta} \| \Lambda - \psi_\delta \|_M + \frac{\sqrt{\alpha^*}}{\beta} \| \! | H - h | \! \| \\ + \frac{1 + \sqrt{\alpha^*}}{\beta} \inf_{p_k \in \mathbb{P}_k^D(\Omega)} \| \! | H - p_k | \! \| + \frac{1}{\beta} \| \mathscr{F} - \mathscr{F}_\delta \|_{V'} \, . \end{aligned} \tag{4.4.11}$$

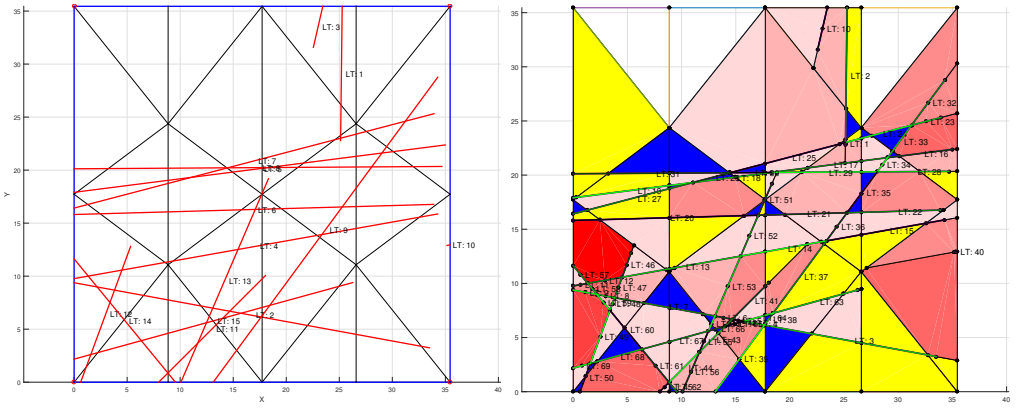
4.5 Implementation

We describe in this section some details concerning the practical implementation of the method.

4.5.1 Mesh generation and trace management

Following closely the ideas in [17], we start by independently introducing a good quality triangular mesh on each fracture, disregarding trace positions. Such triangulation will be called *base mesh*. On each fracture, the base mesh is then modified in such a way that a new polygonal mesh is obtained, that is locally conforming to the traces of the fractures. This means that traces will be covered by edges of the new polygonal elements, though we remark that elements on meshes from different fractures induce a different discretization of the same trace. This new mesh will be suitable for the application of the method described in the previous sections and it will be called *VEM mesh*. The procedure for obtaining the VEM mesh is the following. Whenever a trace intersects an edge of the triangulation, a new node is created at the intersection. Each trace tip defines a new node and the trace segment is prolonged up to the nearest edge of the triangulation, thereby creating a new edge and a new node. When two traces intersect each other, they are split into two sub-traces and in their intersection a new node is created. Whenever an element of the mesh is cut by a (possibly prolonged) trace segment, it is split into two parts which become new elements of the polygonal mesh in their own right. Finally, traces without internal nodes receive the addition of a new node in its midpoint, which is necessary to define the discrete Mortar space for the trace. The overall procedure thus results in a polygonal mesh whose elements are convex polygons made of an arbitrary number of edges.

Figure 4.5.1 is illustrative for such procedure. Focusing on a single fracture, we depict on the left the base mesh introduced, and the local traces present on the fracture, denoted by LT and with a fracture-local numbering from 1 to 15. On the right, the VEM mesh obtained is represented. Note that new traces are introduced by splitting the original traces into sub-traces. Note, as well, the generation of new nodes and



(a) (b)

FIGURE 4.5.1: Mesh examples. Left: base mesh; right: VEM mesh

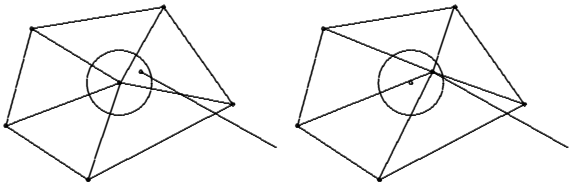
elements obtained via trace segment prolongation and the addition of one internal node (see, e.g., the original local trace 3 on the top of the fracture). To better highlight the number of edges in the elements, a different coloring is used for elements with a different number of edges.

Remark 4.3. In order to verify Assumption 1, a mesh smoothing process can be designed, in order to improve the quality of the VEM mesh, reduce the number of DOFs and prevent irregular elements in the discretization. Let us introduce for each vertex a quantity r_{m} called *moving radius*, defined as a fixed rate of the smallest edge connected to that vertex. Correspondingly, we define a *moving ball* as a ball with center the vertex and radius r_{m}. Then:

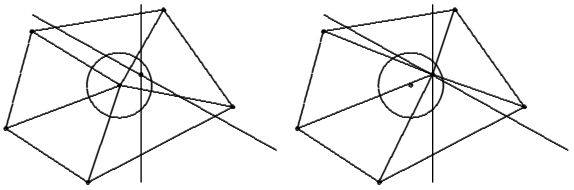
1. if a trace tip lies within a moving ball of a vertex, the vertex is moved on the tip (see Figure 4.2(a));
2. if the intersection between two traces is within the moving ball of a vertex not previously moved to a tip, the latter is moved on the intersection (see Figure 4.2(b));
3. if a vertex not previously moved is closer to a trace than the moving radius, it is moved orthogonally onto the trace (see Figure 4.2(c)).

This procedure does not cover the case in which two traces intersect each other with a very small angle or very small traces, but from the numerical results (see, in particular, section 4.6.2) we can say that the method is sufficiently robust to deal with this kind of issues.

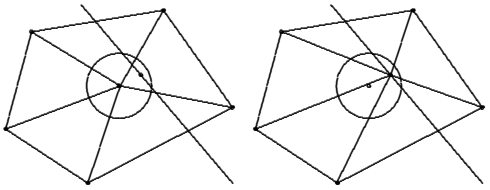
Remark 4.4. Assumption 2 is satisfied by the VEM mesh. Indeed, the triangles of the base mesh are only split when a trace cuts them. Thus, the number of edges of the



(a) Trace tip close to a vertex



(b) Traces intersecting close to a vertex



(c) Trace very close to a vertex

FIGURE 4.5.2: Mesh smoothing process. Left: before mesh smoothing; right: after mesh smoothing

new polygonal elements is limited by the number of traces cutting the element (that is bounded by the number of traces on the fracture), plus 3.

4.5.2 Matrix Formulation of the problem

On the discretization of S induced by the triangulation on the non-mortar fracture, we introduce a finite dimensional subspace of dimension N_S, containing the constant functions (this is required for well-posedness, see Proposition 4.4.1). Let N_h and N_λ be the total number of degrees of freedom for h and λ, respectively, and set $N_{\text{dof}} = N_h + N_\lambda$; let us denote by ϕ_k, $k = 1, \dots, N_h$, and ψ_l, $l = 1, \dots, N_\lambda$, the basis functions for h and λ, respectively. Finally, let N^D be the number of basis functions ϕ_j^D used to define the lifting R^D of the Dirichlet boundary condition. Then, problem (4.4.4) can be written as

$$\sum_{j=1}^{N_h} a_\delta(\phi_j, \phi_k) h_j + \sum_{l=1}^{N_\lambda} b(\phi_k, \psi_l) \lambda_l = (f, \phi_k)_\delta + \left(H^N, \phi_k\right)_{\Gamma^N} - \sum_{j=1}^{N_D} a_\delta\left(\phi_j^D, \phi_k\right) h_j^D$$

$$\sum_{j=1}^{N_h} b(\phi_j, \psi_m) h_j = -\sum_{j=1}^{N_D} b\left(\phi_j^D, \psi_m\right) h_j^D$$

$\forall k = 1, \dots, N_h$ and $\forall m = 1, \dots, N_\lambda$, where h_j^D is the value of $\Pi^0_{k,\Gamma^D}(H^D)$ at the boundary node corresponding to ϕ_j^D. Summarizing, we have to solve the system

$$\begin{pmatrix} A \in \mathbb{R}^{N_h,N_h} & B \in \mathbb{R}^{N_h,N_\lambda} \\ B^\intercal \in \mathbb{R}^{N_\lambda,N_h} & O \in \mathbb{R}^{N_\lambda,N_\lambda} \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \boldsymbol{\Psi} \end{pmatrix}, \tag{4.5.1}$$

where

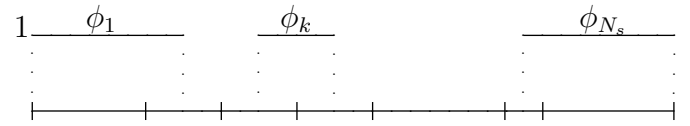
$$A_{kj} = a_\delta(\phi_k, \phi_j), \qquad B_{jl} = b(\phi_j, \psi_l)$$
$$F_k = (f, \phi_k)_\delta + (H^N, \phi_k)_{\Gamma^N} - \sum_{j=1}^{N_D} a_\delta(\phi_j^D, \phi_k) h_j^D, \qquad \Psi_m = -\sum_{j=1}^{N_D} b(\phi_j^D, \psi_m) h_j^D.$$

For the practical construction of the VEM stiffness matrix and right hand side vector, we refer the reader to [12]. We remark that the construction of the matrix B can be done by standard quadrature formulas, since the analytical expression of the basis functions on the edges of each element is known.

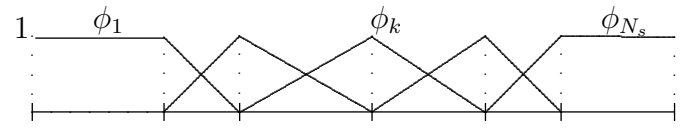
4.5.3 Bases for the discrete Lagrange multipliers

In this subsection we give details about the choice adopted for the space $M_{\delta,S}$, for each $S \in \mathcal{S}$. For a thorough description of the possible choices of Mortar bases, we refer the reader to [73].

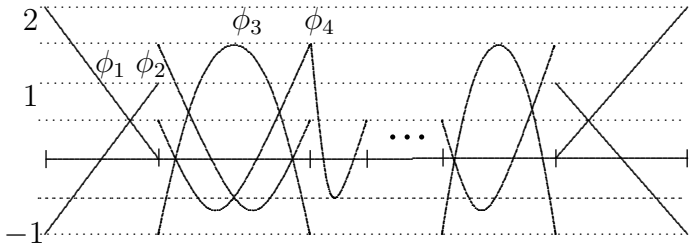
In this work we have used three bases: the basis M_0, composed by piecewise constant functions; the basis M_1, given by continuous piecewise linear functions, except for the first and last intervals on which the functions are taken constant; the basis M_2, given by discontinuous piecewise quadratic functions, except for the first and last interval where the functions are linear. These bases are depicted in Figure 4.5.3.



(a) M_0 basis



(b) M_1 basis



(c) M_2 basis

FIGURE 4.5.3: Lagrange multiplier basis

4.6 Numerical results

We present in this section some numerical results aimed at assessing the practical behavior of the method. The results are obtained on two classes of problems: firstly, we present a benchmark problem for which the exact solution is known, with some convergence results; secondly, we analyse the performance of the method on larger DFNs that introduce several geometrical complexities. All the numerical results here reported are obtained without any kind of mesh smoothing (see Remark 4.3), in order to test the robustness of the method.

4.6.1 Benchmark problem

The benchmark DFN consists of 3 fractures as shown in Figure 4.6.1. Despite being a simple network, it presents two geometrical features (a trace intersection and a trace tip) which make it worthwhile to analyse the behavior of the method at tackling them. The computational domain $\Omega = F_1 \cup F_2 \cup F_3$ is defined by

$$
\begin{aligned}
F_1 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \le x \le 1/2,\ -1 \le y \le 1,\ z = 0\right\},\\
F_2 &= \left\{(x,y,z) \in \mathbb{R}^3 : -1 \le x \le 0,\ y = 0,\ -1 \le z \le 1\right\},\\
F_3 &= \left\{(x,y,z) \in \mathbb{R}^3 : x = -1/2,\ -1 \le y \le 1,\ -1 \le z \le 1\right\},
\end{aligned}
$$

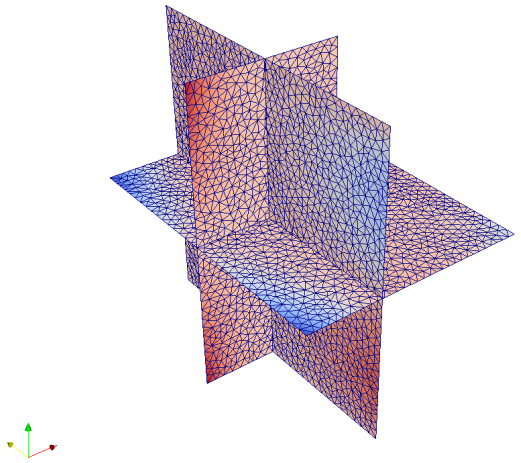


FIGURE 4.6.1: Benchmark problem: geometry of the network

with traces

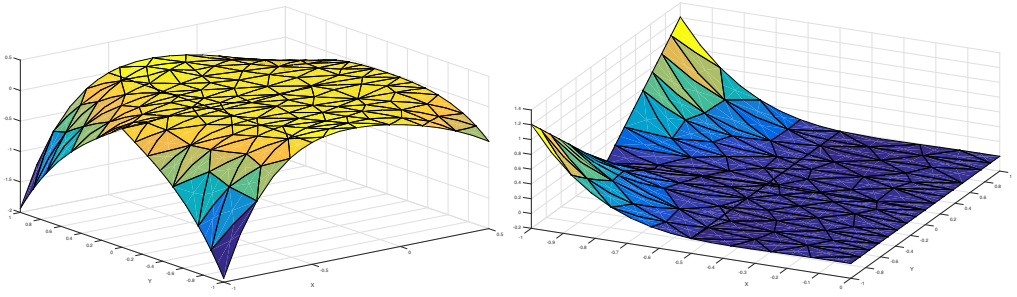
$$
\begin{aligned}
S_1 &= F_1 \cap F_2 = \left\{(x,y,z) \in \mathbb{R}^3 : -1 \le x \le 1/2,\ y = 0,\ z = 0\right\},\\
S_2 &= F_1 \cap F_3 = \left\{(x,y,z) \in \mathbb{R}^3 : x = -1/2,\ -1 \le y \le 1,\ z = 0\right\},\\
S_3 &= F_2 \cap F_3 = \left\{(x,y,z) \in \mathbb{R}^3 : x = -1/2,\ y = 0,\ -1 \le z \le 1\right\}.
\end{aligned}
$$

The problem is defined setting non-homogeneous Dirichlet boundary conditions on the whole boundary $\partial\Omega$, and a load term on each fracture in such a way that the exact solution is given by:

$$
\begin{aligned}
H_1(x,y) &= \frac{1}{10}\left(-x - \frac{1}{2}\right)\left(8xy\left(x^2+y^2\right)\operatorname{arctan2}(y,x) + x^3\right),\\
H_2(x,z) &= \frac{1}{10}\left(-x - \frac{1}{2}\right)x^3 - \frac{4}{5}\pi\left(-x - \frac{1}{2}\right)x^3\,|z|,\\
H_3(y,z) &= (y-1)y(y+1)(z-1)z,
\end{aligned}
$$

where $\operatorname{arctan2}(y,x)$ is the four quadrant inverse tangent function with 2 arguments, that returns the appropriate quadrant of the computed angle y/x. Note that since $H_1, H_2 \notin C^1$, a net flux is expected between F_1 and F_2.

The computed solutions obtained for the hydraulic head on such fractures are shown in Figure 4.6.2. Fluxes exchanged between F_1 and F_2, computed with all three considered choices for the mortar bases are shown in Figure 4.6.3, where they are compared with the exact one.



(a) M_0 basis (b) M_0 basis

FIGURE 4.6.2: Benchmark problem: computed hydraulic head on fractures F_1 (left) and F_2 (right)

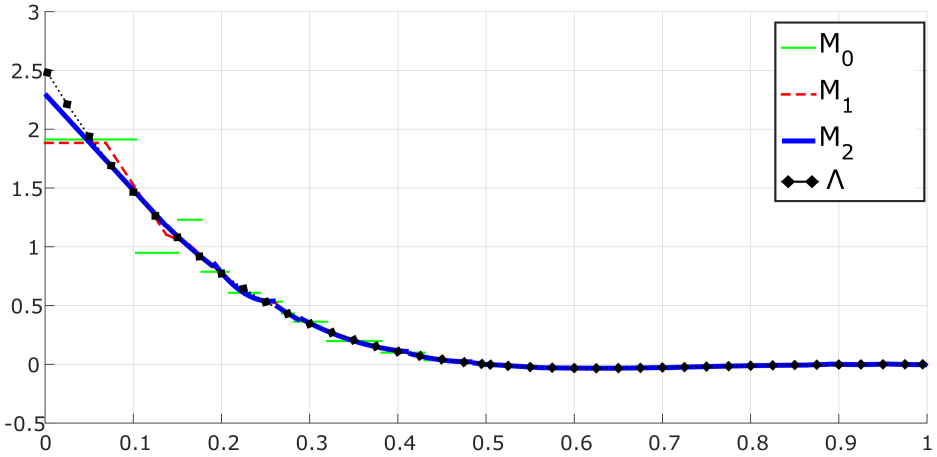


FIGURE 4.6.3: Benchmark problem: computed and exact fluxes

In order to present convergence results, we remark that since the values of the discrete solution are not explicitly known inside the elements but only on the set of DOFs, the errors are computed by projecting the discrete solution on the space of polynomials of degree k, as is the usual procedure with the VEM [14]:

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

where $\Pi^\nabla_{E,k}$ is the projection operator of order k as defined in section 4.3, H is the exact solution and h_E is the discrete solution restricted to element E. Regarding the errors of approximation of Λ, we measure them on each trace both in $\mathrm{L}^2(S)$ and $\mathrm{H}^{-\frac{1}{2}}(S)$ norm; for practical computational issues, we approximate this latter norm with a weighted

		h		λ on S_1	
VEM order	Mortar basis	L^2 Norm	H^1 Norm	L^2 Norm	$H^{-\frac{1}{2}}$ Norm
1	M_0	1.00 (1)	0.50 (0.5)	1.19	1.79
1	M_1	1.00 (1)	0.50 (0.5)	1.26	1.87
2	M_0	1.38 (1.5)	0.91 (1)	0.98	1.54
2	M_1	1.50 (1.5)	1.01 (1)	1.54	2.05
2	M_2	1.51 (1.5)	1.01 (1)	2.45	3.02

TABLE 4.6.1: Benchmark problem: convergence rates for several VEM orders and Mortar bases. The numbers in parentheses indicate the expected rates

$L^2(S)$ norm:

$$\left(Err^{\Lambda}_{L^2}\right)^2 = \sum_{S\in\mathcal{S}}\sum_{e\subset S} \|\Lambda-\lambda\|^2_e\,,$$

$$\left(Err^{\Lambda}_{H^{-\frac{1}{2}}}\right)^2 = \sum_{S\in\mathcal{S}}\sum_{e\subset S} |e|\,\|\Lambda-\lambda\|^2_e\,.$$

In Figure 4.6.4, focusing on fracture F_1, we present the convergence curves for different combinations of the order k for the VEM space and of the type of Mortar basis. Namely, in the left column we report the behavior of the errors $Err^H_{L^2}$ and $Err^H_{H^1}$ (labeled by L^2 and H^1, respectively); the errors are plot versus the total number of h-DOFs on the fracture. In the right column we report the errors $Err^{\Lambda}_{L^2}$ and $Err^{\Lambda}_{H^{-\frac{1}{2}}}$ (labeled by L^2 and $H^{-1/2}$, respectively); here, the errors are plot versus the number of λ-DOFs on the traces of F_1.

Finally, Table 4.6.1 reports, for all the analysed cases, the computed convergence rates with respect to the number of DOFs. Namely, we report the computed rates of convergence for h with respect to the h-DOFs (the expected values being reported in parentheses); note the very good agreement between the computed and the expected rates, except for the case $k = 2$ and M_0, in which the low order of the mortar basis slows down the rate of convergence for the hydraulic head. Focusing on trace S_1, we also report the computed rates of convergence for λ with respect to the number of λ-DOFs. The rates of convergence for the λ-errors with respect to the number of h-DOFs, not listed here, are approximately one half of the reported values; this is in agreement with the fact that the number of λ-DOFs scales as the square root of the number of h-DOFs.

4.6.2 Complex networks

In this section we present results obtained on more complex networks. The first one, DFN36, consists of 36 fractures. The geometry of the DFN is depicted in Figure 4.6.6, from which the geometrical complexity of the domain can be seen. A non-homogeneous constant Neumann boundary condition ($H^N = 100$) has been set on one fracture (called

source fracture), and a homogeneous Dirichlet boundary condition has been set on another fracture (sink fracture). Homogeneous Neumann boundary conditions on the remaining part of the boundary isolate all the other fractures from the surrounding medium.

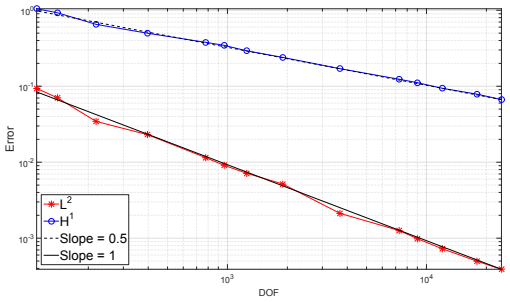
The plots in Figure 4.6.5 report the computed total net fluxes exchanged by the source and sink fracture versus the number of DOFs on traces (logaritmic x-scale), for VEM of order $k = 1, 2$ and 3, and mortar bases M_0, M_1 and M_2. The value Δ reported is the difference between the two curves and is an indication of the global conservation state of the method in the whole DFN. Results show the tendency to approximate the expected values and we note that, interestingly, almost no difference in flux values is appreciated for different choices of mortar bases.

As a further quality indicator for the obtained solution, we introduce a measure of the error of the jump of the hydraulic head on traces. Namely, we set

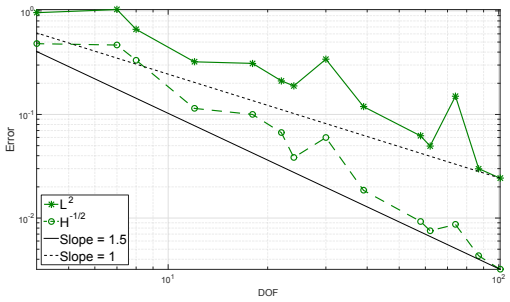
$$E_h = \sum_{S \in \mathcal{S}} \| [\![h]\!]_S \|^2_{\mathrm{L}^2(S)} .$$

The computed values are shown in Figure 4.6.7 for VEM of order $k = 1, 2$ and 3, using the basis M_1. For all orders, a decrease in this parameter was observed with increasing number of DOFs as expected, but interestingly, with a similar rate. Since the defined quantity does not constitute a norm, no further conclusions about convergence can be drawn.

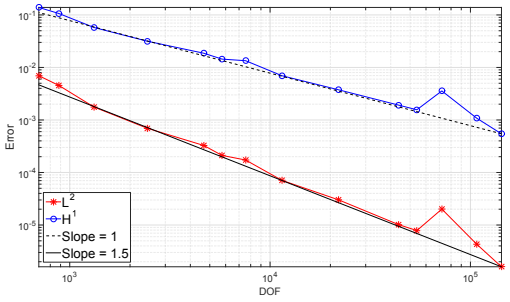
As a second example, a 134 fracture network is proposed (DFN134, Figure 4.6.8). As far as geometrical complexities are concerned, this DFN is far more challenging than DFN36, as it exhibits several critical features: very small angles at trace intersections (thus challenging the shape regularity of the elements stated by Assumption 1 and discussed in Remark 4.3), almost parallel traces, large variation of trace lengths and fracture sizes. Three fractures were chosen as source fractures by imposing non-homogeneous Neumann boundary conditions. A fourth fracture was set as sink fracture, and on one of its edges a homogeneous Dirichlet boundary conditions was set. Homogeneous Neumann conditions were imposed on all the remaining components of the boundary. In Figure 4.6.9 we report some data for a particularly intricate fracture, where the problem has been solved using VEM of order $k = 2$ and the M_1 basis. The VEM mesh is presented, as well as the affine interpolation of the computed hydraulic head solution and the corresponding velocity field obtained from the gradients of the computed hydraulic head. From the detail reported in the bottom right figure, it can be seen how elements of order 2 allow for a better representation of the change in slope between close traces thanks to the added DOFs in the midpoints of each of the edges.



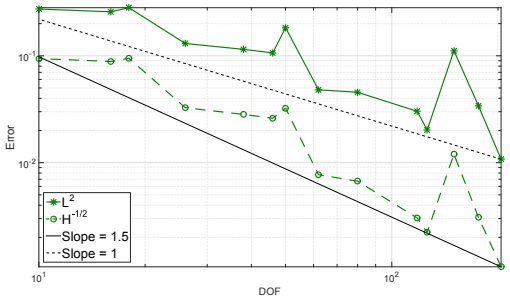
(a) Order 1 - M_0 - Error in h



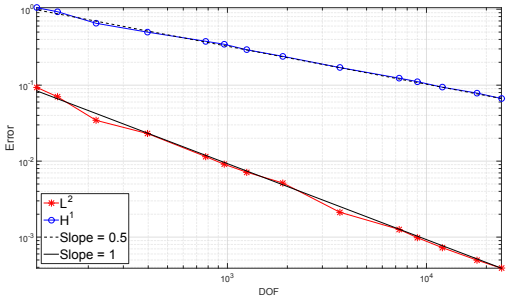
(b) $k = 1$ - Basis M_0 - Error in λ



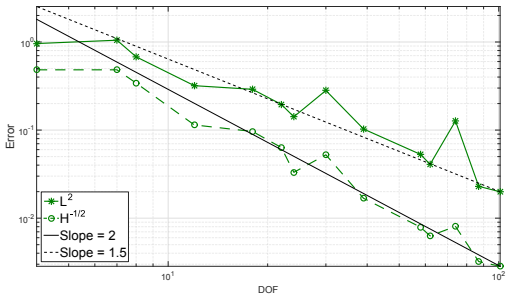
(c) $k = 2$ - Basis M_0 - Error in h



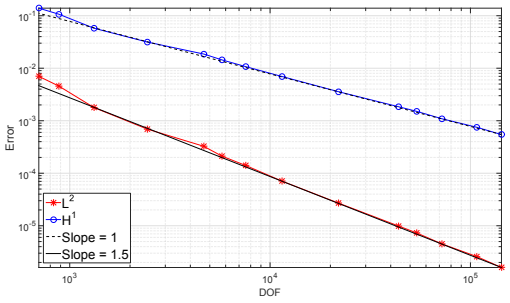
(d) $k = 2$ - Basis M_0 - Error in λ



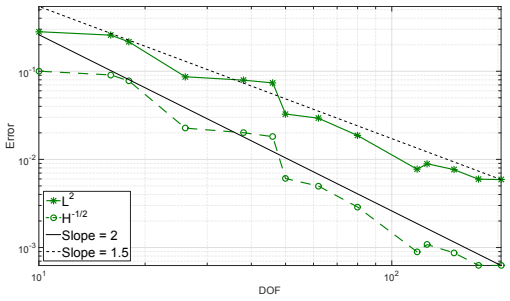
(e) $k = 1$ - Basis M_1 - Error in h



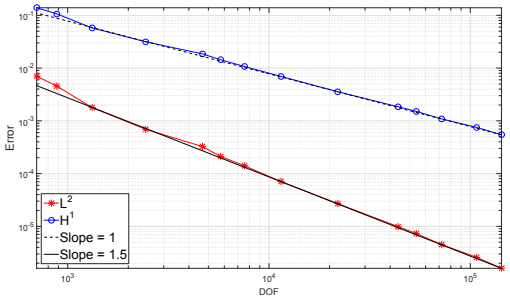
(f) $k = 1$ - Basis M_1 - Error in λ



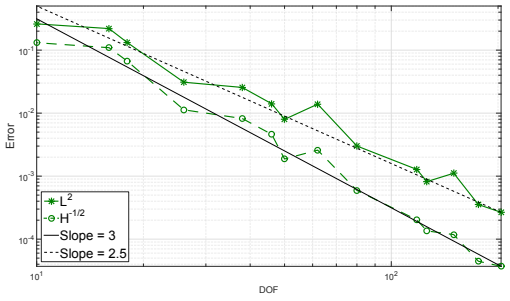
(g) $k = 2$ - Basis M_1 - Error in h



(h) $k = 2$ - Basis M_1 - Error in λ

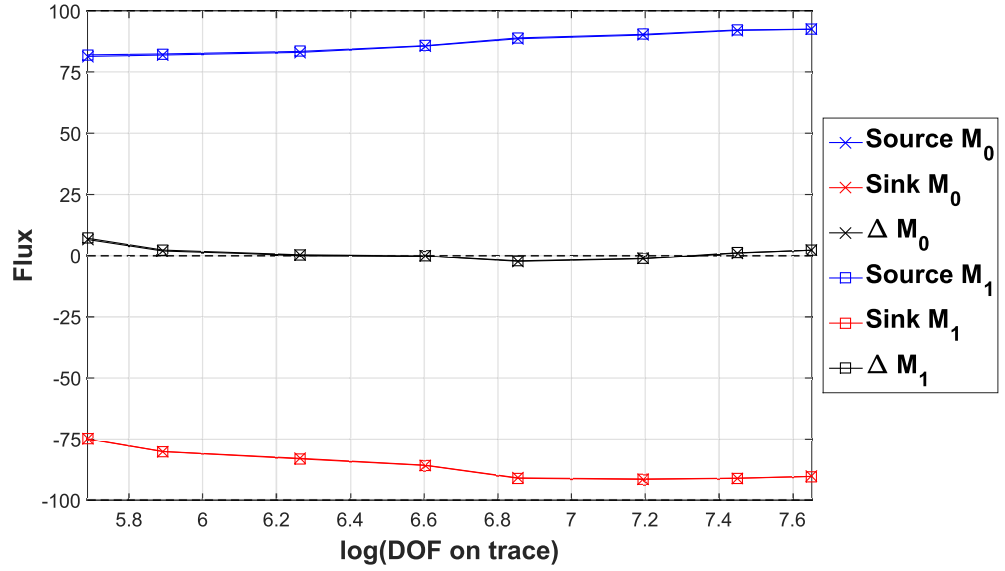


(i) $k = 2$ - Basis M_2 - Error in h

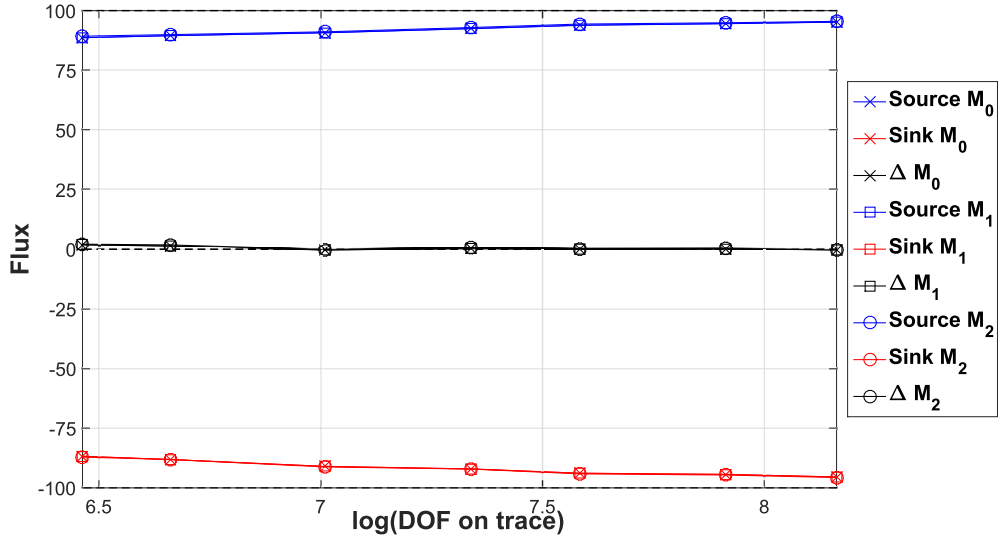


(j) $k = 2$ - Basis M_2 - Error in λ

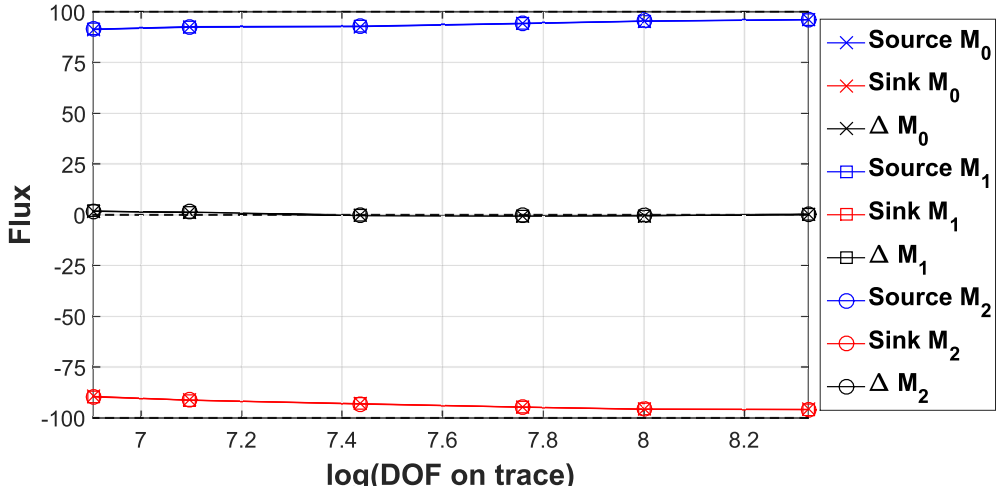
FIGURE 4.6.4: Benchmark problem: convergence curves measured on fracture F_1



(a) $k = 1$



(b) $k = 2$



(c) $k = 3$

FIGURE 4.6.5: DFN36: flux results

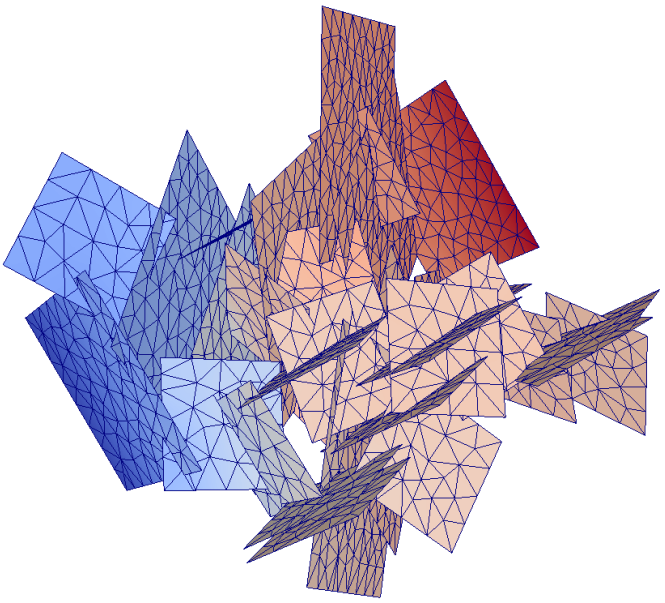


FIGURE 4.6.6: DFN36: geometry of the network and computed hydraulic head (as a scale of colours)

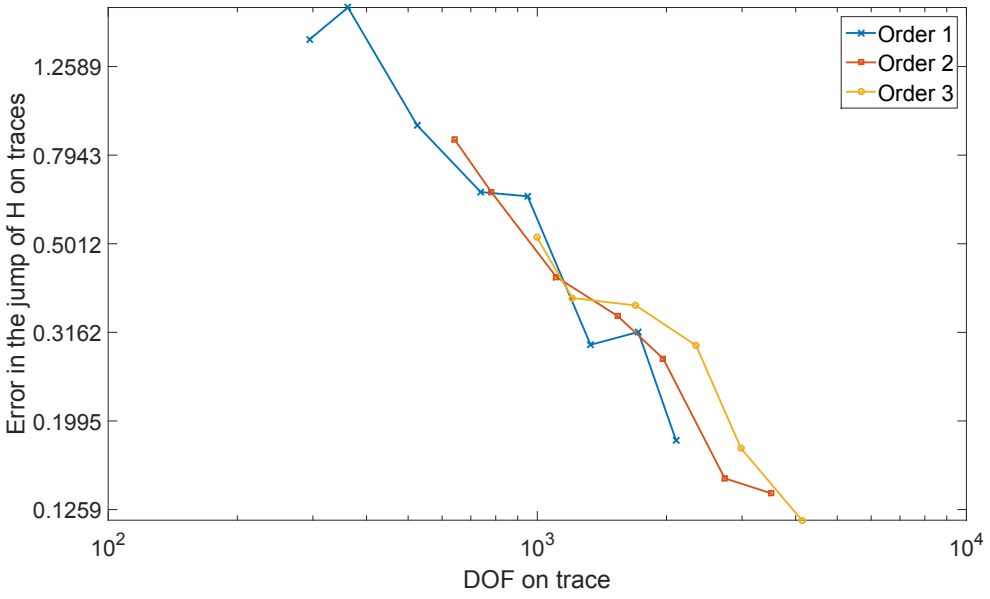


FIGURE 4.6.7: DFN36: error in the jump of the hydraulic head on traces

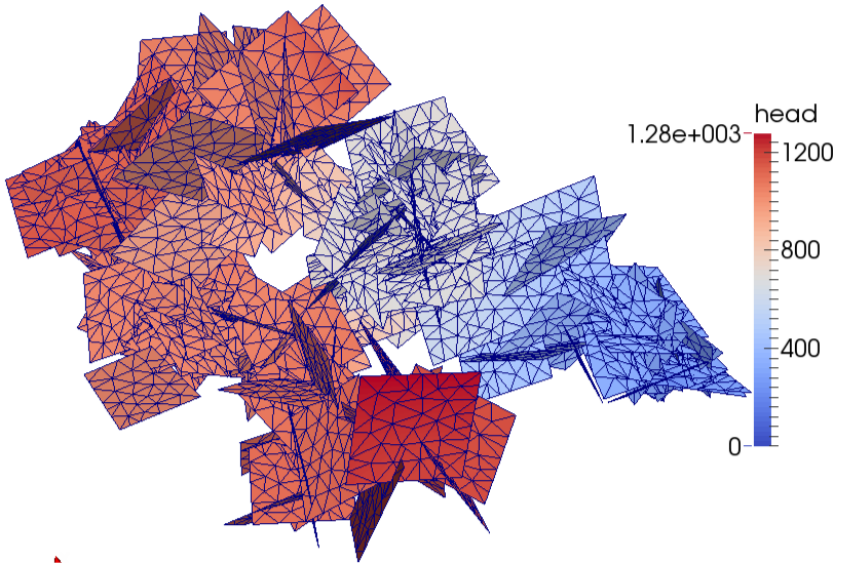
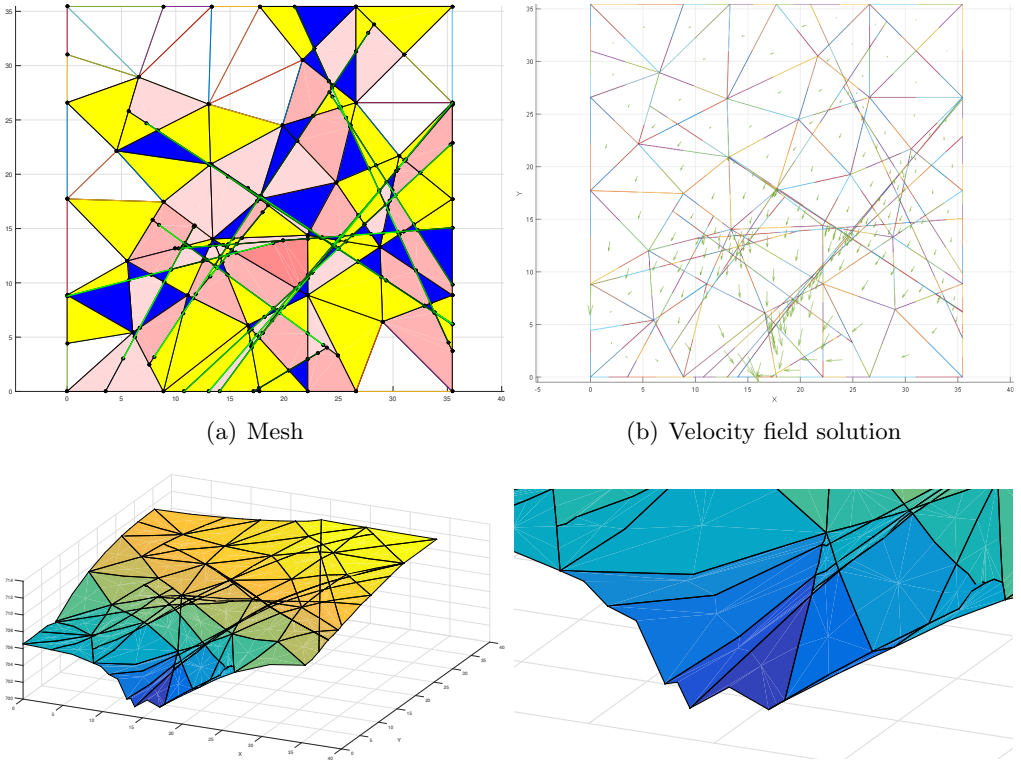


FIGURE 4.6.8: DFN134: geometry of the network and computed hydraulic head (as a scale of colors)



(a) Mesh

(b) Velocity field solution

(c) Hydraulic head solution

(d) Detail of hydraulic head solution

FIGURE 4.6.9: DFN134: a selected fracture

4.7 Conclusions

We have introduced a new approach for flow simulations in Discrete Fracture Networks. The key feature is given by its capability to work with arbitrary (good quality) meshes generated on the fractures. Taking advantage of the versatility of the Virtual Element Method in handling polygonal meshes, each arbitrary mesh is easily modified in such a way that local conformity of the meshes is obtained for almost any trace disposition. Using the hybrid formulation of the Mortar method, only "weak" continuity is required for the hydraulic head along the intersections between fractures.

The main advantage of the approach presented here, with respect to the method proposed in chapter 3, is that, besides the computation of the hydraulic head, the present approach allows for a direct approximation of the flux on each trace, whereas before the flux exchange is derived from the values of the hydraulic head.

The validity of the approach proposed is supported by numerical experiments, showing optimal convergence for the primal variable; furthermore, the behaviour of the method is quite satisfactory also when it is applied to DFNs with complex geometry.

Future developments include the extension to more complex flow models and in particular to the case of non-constant transmissivity values. Furthermore, we aim at investigating a possible parallel implementation, which is recommended for tackling large scale DFNs for realistic underground flow simulations.

4.8 Proof of Theorem 4.2

This section is devoted to the proof of Theorem 4.2. The proof follows the lines of the proofs of [68, Theorem 3] and [10, Theorem 3.1]. We first prove the following preliminary result, which extends Poincaré's inequality to a DFN.

Lemma 4.5. *Let* $\tilde{W} = \left\{ v \in V : \int_S [\![v]\!] = 0 \quad \forall S \in \mathcal{S} \right\}$. *Then*

$$\exists C_\Omega > 0 : \forall w \in \tilde{W} \qquad \left(\sum_{i=1}^{N} \|w\|^2_{\mathrm{L}^2(F_i)} \right)^{\frac{1}{2}} \leq C_\Omega \, |\!|\!| \, w \, |\!|\!| \tag{4.8.1}$$

Proof. First, notice that $|\!|\!| \cdot |\!|\!|$ is a norm on $\tilde{W}$ (see Proposition 4.4.1), thus the right hand side of (4.8.1) does not vanish, unless w is identically zero. By contradiction, suppose

$$\forall C > 0, \, \exists w_C \in \tilde{W} : \, \| \, w_C \, \|_\Omega := \left(\sum_{i=1}^{N} \|w_C\|^2_{\mathrm{L}^2(F_i)} \right)^{\frac{1}{2}} > C \, |\!|\!| \, w_C \, |\!|\!| \, ,$$

then it is possible to build a sequence $w_k \in \tilde{W}$, $k \in \mathbb{N}$, of functions such that $\| w_k \|_\Omega > k \| \! | w_k | \! \|$ and, without loss of generality, suppose that $\| w_k \|_\Omega = 1$ for all k. Then, since $\| w_k \|_{\mathrm{H}^1(F_i)}$ is limited for all $i = 1, \dots, N$, w_k converges weakly in V to a function $w^\star$ up to sub-sequences. Clearly, ∇w_k converges to $\nabla w^\star$ weakly. Then, since

$$0 \leq \|\nabla w_k - \nabla w^\star\|_{\mathrm{L}^2(F_i)} = \|\nabla w_k\|^2_{\mathrm{L}^2(F_i)} - 2(\nabla w_k, \nabla w^\star)_{\mathrm{L}^2(F_i)} + \|\nabla w^\star\|^2_{\mathrm{L}^2(F_i)} ,$$

and $\| \nabla w_k \|_{\mathrm{L}^2(F_i)} < \frac{1}{k}$, taking the limit for $k \to \infty$, it follows that $\| \nabla w^\star \|_{\mathrm{L}^2(F_i)} = 0$. Then, $w^\star$ is constant on each fracture. By the same arguments used in the proof of Proposition 4.4.1, it follows that $w^\star$ must be the null function. Moreover, since $\mathrm{H}^1(F_i)$ is compactly embedded in $\mathrm{L}^2(F_i)$, w_k converges strongly to $w^\star$ in $\mathrm{L}^2(F_i)$, for all $i = 1, \dots, N$. Since $\| w_k \|_{\mathrm{L}^2(F_i)} \overset{k\to\infty}{\longrightarrow} \| w^\star \|_{\mathrm{L}^2(F_i)}$ for all $i = 1, \dots, N$, we obtain $\| w^\star \|_\Omega = 1$, which is a contradiction. □

We can now prove the a priori error estimate.

Proof of Theorem 4.2. Let $h_l \in W^D_\delta$ be the a-orthogonal projection of $H \in V^D$ over W^D_δ, such that

$$\forall v_\delta \in W^D_\delta, \qquad a\left(H - h_l, v_\delta\right) = 0 .$$

Exploiting the properties of the projection, we have

$$\| \! | H - h | \! \|^2 = \| \! | H - h_l | \! \|^2 + \| \! | h_l - h | \! \|^2 = \left(\inf_{v_\delta \in W^D_\delta} \| \! | H - v_\delta | \! \| \right)^2 + \| \! | h_l - h | \! \|^2 .$$

As far as the second term is concerned, recalling (4.3.6) we have

$$\alpha_* \| \! | h_l - h | \! \|^2 = \alpha_* a(h_l - h, h_l - h) \leq a_\delta(h_l - h, h_l - h) .$$

By using the problem definitions (4.2.12) and (4.4.4), and introducing an arbitrary $p \in \mathbb{P}^D_k$, for which (4.3.4) holds, we have

$$\begin{aligned}
a_\delta\left(h_l - h, h_l - h\right) &= a_\delta\left(h_l - p, h_l - h\right) + a_\delta\left(p, h_l - h\right) - a_\delta\left(h, h_l - h\right) \\
&= a_\delta\left(h_l - p, h_l - h\right) + a(p, h_l - h) - (f, h_l - h)_\delta \\
&\quad + b\left(h_l - h, \lambda\right) - \left(H^N, h_l - h\right)_{\Gamma^N} \\
&= a_\delta\left(h_l - p, h_l - h\right) + a(p - H, h_l - h) + a(H, h_l - h) \\
&\quad - (f, h_l - h)_\delta + b\left(h_l - h, \lambda\right) - \left(H^N, h_l - h\right)_{\Gamma^N} \\
&= a_\delta\left(h_l - p, h_l - h\right) + a(p - H, h_l - h) \\
&\quad - (f, h_l - h)_\delta + (f, h_l - h) - b\left(h_l - h, \Lambda\right) ,
\end{aligned}$$

where we have used that $b\left(h_l - h, \lambda\right) = 0$ because $h_l - h \in W_\delta$. Introducing $\mathscr{F}$, $\mathscr{F}_\delta$ and a generical $\psi_\delta \in M_\delta$, since $b\left(h_l - h, \psi_\delta\right) = 0$ we have

$$\begin{aligned} a_\delta\left(h_l - h, h_l - h\right) &= a_\delta\left(h_l - p, h_l - h\right) + a(p - H, h_l - h) - b\left(h_l - h, \Lambda\right) \\ &\quad + {}_{V'}\langle \mathscr{F} - \mathscr{F}_\delta, h_l - h\rangle_V \\ &\leq \left(\alpha^* \left|\!\left|\!\left| h_l - p \right|\!\right|\!\right| + \left|\!\left|\!\left| H - p \right|\!\right|\!\right| + \frac{b\left(h_l - h, \Lambda - \psi_\delta\right)}{\left|\!\left|\!\left| h_l - h \right|\!\right|\!\right|}\right) \left|\!\left|\!\left| h_l - h \right|\!\right|\!\right| \\ &\quad + \| \mathscr{F} - \mathscr{F}_\delta \|_{V'} \| h_l - h \|_V \\ &\leq \left(\alpha^* \left|\!\left|\!\left| h_l - p \right|\!\right|\!\right| + \left|\!\left|\!\left| H - p \right|\!\right|\!\right| + \frac{b\left(h_l - h, \Lambda - \psi_\delta\right)}{\left|\!\left|\!\left| h_l - h \right|\!\right|\!\right|}\right. \\ &\quad \left. + (1 + C_\Omega) \| \mathscr{F} - \mathscr{F}_\delta \|_{V'}\right) \left|\!\left|\!\left| h_l - h \right|\!\right|\!\right|, \end{aligned}$$

where in the last step inequality (4.8.1) has been used (see (4.2.13) for the definition of the V-norm). The proof of (4.4.10) is thus completed using the triangle inequality and suitably taking the supremums and infimums.

In order to prove (4.4.11), let us consider an arbitrary $\psi_\delta \in M_\delta$. By applying (4.4.7), (4.2.12) and (4.4.4) we get:

$$\begin{aligned} \beta \| \psi_\delta - \lambda \|_M &\leq \sup_{v_\delta \in V_\delta} \frac{b\left(v_\delta, \psi_\delta - \lambda\right)}{\| v_\delta \|_V} = \sup_{v_\delta \in V_\delta} \frac{b\left(v_\delta, \Lambda - \lambda\right) + b\left(v_\delta, \psi_\delta - \Lambda\right)}{\| v_\delta \|_V} \\ &= \sup_{v_\delta \in V_\delta} \frac{a_\delta\left(h, v_\delta\right) - \left(f, v_\delta\right)_\delta - a\left(H, v_\delta\right) + \left(f, v_\delta\right) + b\left(v_\delta, \psi_\delta - \Lambda\right)}{\| v_\delta \|_V}. \end{aligned}$$

Next, introducing an arbitrary $p \in \mathbb{P}_k^D(\Omega)$, by (4.3.4) we get

$$\begin{aligned} \beta \| \psi_\delta - \lambda \|_M &\leq \sup_{v_\delta \in V_\delta} \| v_\delta \|_V^{-1} \Big[a_\delta\left(h - p, v_\delta\right) + a\left(p - H, v_\delta\right) \\ &\qquad + {}_{V'}\langle \mathscr{F} - \mathscr{F}_\delta, v_\delta\rangle_V + b\left(v_\delta, \psi_\delta - \Lambda\right)\Big] \\ &\leq \sup_{v_\delta \in V_\delta} \| v_\delta \|_V^{-1} \left[\sqrt{a_\delta\left(h - p, h - p\right)} \sqrt{a_\delta\left(v_\delta, v_\delta\right)}\right] + \left|\!\left|\!\left| H - p \right|\!\right|\!\right| \\ &\quad + \| \mathscr{F} - \mathscr{F}_\delta \|_{V'} + \| \Lambda - \psi_\delta \|_M \\ &\leq \sqrt{\alpha^*} \left|\!\left|\!\left| h - p \right|\!\right|\!\right| + \left|\!\left|\!\left| H - p \right|\!\right|\!\right| + \| \mathscr{F} - \mathscr{F}_\delta \|_{V'} + \| \Lambda - \psi_\delta \|_M \\ &\leq \sqrt{\alpha^*} \left|\!\left|\!\left| H - h \right|\!\right|\!\right| + (1 + \sqrt{\alpha^*}) \left|\!\left|\!\left| H - p \right|\!\right|\!\right| + \| \mathscr{F} - \mathscr{F}_\delta \|_{V'} + \\ &\quad + \| \Lambda - \psi_\delta \|_M. \end{aligned}$$

The proof is concluded by the triangle inequality and taking the infimum over $\mathbb{P}_k^D(\Omega)$. □

Chapter 5

Time dependent problems and the transport equation

5.1 Introduction

In chapters 2, 3 and 4 we have presented a family of methods and applications of the Virtual Element Method (VEM) to solve the problem of solving Darcy's flow in Discrete Fracture Networks (DFN), and we have used those results to obtain the pressure head and the velocity field in the network. In this chapter we aim to improve the approaches that we have developed and study the behavior of the Virtual Element Method applied to solving the transient diffusion-convection-reaction equation in DFNs with SUPG stabilization. This more ambitious goal has many useful potential applications like pollutant transport, fluid transport, particle tracing, etc. We give a brief review of the choice in VEM formulations followed by benchmark results and an application to DFNs.

5.2 Problem description

5.2.1 The reaction–diffusion–advection equation

This equation is used to model the concentration of some chemical substance under the processes of diffusion, transportation of mass due to the movement of the underlying medium and undergoing reaction. In its most general form, the time dependent equation is written as

$$\frac{\partial u}{\partial t} + \nabla \cdot (\beta u - \nu \nabla u) + \gamma u = f, \tag{5.2.1}$$

where u represents the concentration, ν is the diffusion coefficient, β is the flow velocity, γ is a source parameter and f is the source function that models the changes in concentration due to reactions. This equation is obtained from the conservation of mass in a continuum model and there may be nonlinearities present in the coefficients. When some of the coefficients are null, the equation reduces to special cases. In particular, when no reaction term is present ($\gamma = 0$) it simplifies to the advection–diffusion (also known as convection-diffusion) equation, which is of particular interest in transport problems and in our study of DFNs. The steady state equation is obtained by putting $\frac{\partial c}{\partial t} = 0$.

5.2.2 Numerical treatment of the equation

It is well known that this formulation of the problem (5.2.1) is not viable when the convective term is dominant with respect to the diffusive term. The classical Galerkin scheme for finite elements becomes very unstable in these situations and a stabilized form of the numerical scheme is required in order to prevent spurious oscillations that can render the numerical solution completely erroneous. Many stabilization techniques have been proposed in the literature. We refer the reader to [40] and [24] for an overview and results on stabilized finite elements.

More precisely, let us consider the classic weak formulation of the steady state problem with homogeneous Dirichlet boundary conditions. We denote $(,)$ as the standard L^2 inner product, Ω a domain and define $B\colon\ \mathrm{H}_0^1(\Omega) \times \mathrm{H}_0^1(\Omega) \to \mathbb{R}$ and $F\colon\ \mathrm{H}_0^1(\Omega) \to \mathbb{R}$ such that

$$B(u,v) = (\nu \nabla u, \nabla v) + (\beta \cdot \nabla u, v) + (\gamma u, v) \quad \forall u, v \in \mathrm{H}_0^1(\Omega), \tag{5.2.2}$$

and

$$F(v) = (f, v) \quad \forall v \in \mathrm{H}_0^1(\Omega)\ .$$

We arrive at the formulation, find $u \in \mathrm{H}_0^1(\Omega)$ such that

$$B(u,v) = F(v) \quad \forall v \in \mathrm{H}_0^1(\Omega)\,. \tag{5.2.3}$$

This formulation can be adjusted to more general boundary conditions without problems. The Streamline Upwind Petrov Galerkin (SUPG) stabilization approach redefines problem (5.2.3) as

$$B_{supg}(u,v) = F_{supg}(v) \quad \forall v \in \mathrm{H}_0^1(\Omega), \tag{5.2.4}$$

where

$$\begin{aligned} B_{supg}(u,v) &= (\nu\nabla u,\nabla v)+(\beta\cdot\nabla u,v)+(\gamma u,v) \\ &+ \sum_{E\in\mathcal{T}_h}\tau_E(-\nu^E\Delta u+\beta\cdot\nabla u+\gamma u,\beta\cdot\nabla v)_E,\ \forall u\in \mathrm{H}_0^1(\Omega)\cap\mathrm{H}^2 \text{ and } v\in \mathrm{H}_0^1(\Omega) \end{aligned}$$

and

$$F_{supg}(v)=(f,v)+\sum_{E\in\mathcal{T}_h}\tau_E(f,\beta\cdot\nabla v)_E \quad \forall v\in \mathrm{H}_0^1(\Omega)\,.$$

The subscript E stands for the elements of the triangulation $\mathcal{T}_h$, and ν^E is a constant approximation of the trace of the diffusion coefficient for each element E. The stability parameter τ_E on E of the triangulation is classically defined by

$$\begin{aligned} &Pe_E = m_k^E\frac{||\beta||_p h_E}{2\nu_{max}^E}, \\ &\tau_E = \frac{h_E}{2||\beta||_P}\min\{Pe_E,1\} = \begin{cases} m_k^E\frac{h_E^2}{4\nu_{max}^E}, & 0\le Pe_E<1 \\ \frac{h_E}{2||\beta||_p}, & 1\le Pe_E \end{cases} \\ &m_k^E = \min\left\{\frac{1}{3},2\tilde{C}_k^E\right\}, \\ &\tilde{C}_k^E h_E^2||\nabla\cdot\sqrt{\nu^E}\nabla v_k||_{0,E}^2\le||\sqrt{\nu^E}\nabla v_k||_{0,E}^2\forall v_k\in\mathbb{P}_k(E)\,, \end{aligned} \tag{5.2.5}$$

where h_E is the diameter of E, $\mathbb{P}_k(E)$ is the space of polynomials of degree k on E, and $||\beta||_p = \|\,|\beta|_p\,\|_{E,\inf}$, i.e. $||\beta||_p$ is the infinity norm on E of the p-norm of the vector β. Pe_h is called the mesh Péclet number for E. The value for m_k^E changes according to the polynomial degree of the finite element approximation and is related to the inverse inequalities for the mesh elements. Note that the proposed stabilization introduces a term that is consistent (it is zero if it is computed with the exact solution), or tends to zero when it is computed with functions that approximate the exact solution.

For the numerical experiments in this chapter, both the complete as well as a reduced form of the SUPG stabilization were used, the latter based on the one proposed by Brezzi and Pitkäranta in [27] (see also [53], [51]). Unfortunately, we cannot hope to retain optimum convergence with this stabilization due to the fact that we are introducing a non consistent stabilization term (i.e., a term that does not go to zero as we make the mesh parameter decrease).

The formulation of the reduced stabilization used for the numerical experiments in this chapter is

$$B_{bp}(u,v)=F_{bp}(v) \quad \forall v\in \mathrm{H}_0^1(\Omega), \tag{5.2.6}$$

where

$$\begin{aligned} B_{bp}(u,v) &= (\nu\nabla u, \nabla v) + (\beta\cdot\nabla u, v) + (\gamma u, v) \\ &+ \sum_{E\in\mathcal{T}_h} \tau_E(\beta\cdot\nabla u + \gamma u, \beta\cdot\nabla v)_E,\ \forall u, v \in \mathrm{H}_0^1(\Omega) \end{aligned}$$

and

$$F_{bp}(v) = (f, v) + \sum_{E\in\mathcal{T}_h} \tau_E(f, \beta\cdot\nabla v)_E \quad \forall v \in \mathrm{H}_0^1(\Omega)\,.$$

The difference lies in the fact that in (5.2.6), the Laplacian term is removed. For linear elements, the Laplacian is zero and therefore they both coincide. Starting from the second order, the Laplacian is no longer zero and the formulations are effectively different. The reason for this alternative comes from the difficulties of obtaining information about the Laplacian of a basis function of the local Virtual Element space based on the degrees of freedom in all but the most simple cases.

To solve the time dependent equations, we first discretize in space and then in time. Solutions of the initial value problem for the degrees of freedom are computed using the classical second order Crank-Nicholson scheme (see for example [46]). Explicit and Implicit Euler methods can also be used, with some stability restrictions on the size of the time steps for the explicit case.

5.3 Virtual Element formulation

A very thorough work on conforming and non conforming Virtual Elements for second order elliptic equations can be found in [33] with convergence results and guidelines on the implementation. In [56], a comprehensive presentation of the different possibilities for the VEM formulation is put forward. For other works on the VEM we refer the reader to the citations provided in previous chapters.

Unlike in other applications of the VEM to DFNs, we will now tackle the full second order equation with non constant coefficients. We have chosen the external VEM formulation in favor of the internal one, since we consider the implementation more straightforward and uses the same ideas of an L^2 projector, first described in [2], and whose implementation does not depend on whether the coefficients are constant or not. The internal formulation defines the projection operators differently, according to whether the coefficients of the differential equation are constant or can vary. Furthermore, in the external formulation, we directly compute a certain polynomial projection of the derivatives of VEM basis functions, while in the internal formulation first the

polynomial projection is obtained which we then differentiate (for example, to compute its gradient).

For our purposes, we briefly recall here the external formulation for the Virtual Element framework used for our experiments, as it differs from the internal formulation described in previous chapters. We use the conforming variety for our numerical experiments, although the conforming and non conforming VEM have been shown to perform very similarly [33].

The construction of the local Virtual Element space $V_{k,h}^E$, the choice in degrees of freedom as well as the global space were defined in 3.3. We recall the main ideas here: given a domain F covered with a triangulation τ_h, for a desired order of accuracy k and with the space $\mathbb{P}_k$ of the polynomials of maximum degree k, let us define the local space $V_{k,h}^E$ as:

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

where h is a mesh parameter, E is an element of the triangulation, ∂E is its border and e an edge. The global virtual element space is:

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

Recalling Equation 5.2.2, let us define its terms as:

$$\begin{aligned} a(u,v) &= (\nu \nabla u, \nabla v)_F, \\ b(u,v) &= (\beta \cdot \nabla u, v)_F, \\ c(u,v) &= (\gamma u, v)_F, \end{aligned}$$

and their local element counterparts as

$$\begin{aligned} a^E(u,v) &= (\nu \nabla u, \nabla v)_E, \\ b^E(u,v) &= (\beta \cdot \nabla u, v)_E, \\ c^E(u,v) &= (\gamma u, v)_E. \end{aligned}$$

The main idea in the VEM is to approximate these bilinear forms without explicitly knowing the shape functions in $V_{k,h}^E$ so as to retain optimal convergence of the error. In order to do so, we will consider a basis for $\mathbb{P}_k$ given by the polynomials p_α with $\alpha = 1, ..., dim(\mathbb{P}_k)$. We will use ϕ_i to denote the shape functions in $V_{k,h}^E$. The following projection operators are introduced: $\Pi_k^0 :\ V_{k,h}^E \to \mathbb{P}_k$ is the solution of an L^2 projection

problem given by

$$(p_\alpha, \Pi_k^0 \phi_i)_E = (p_\alpha, \phi_i)_E \quad \forall \alpha = 1, ..., dim(\mathbb{P}_k).$$

The second projector that we need is $\Pi_{k-1}^0 : V_{k,h}^E \to (\mathbb{P}_{k-1}(E))^2$ given by L^2 projection of $\nabla\phi_i$ such that

$$(p_\alpha, \Pi_{k-1}^0 \nabla\phi_i)_E = (p_\alpha, \nabla\phi_i)_E \quad \forall \alpha = 1, ..., dim(\mathbb{P}_{k-1}).$$

Note that only the basis for the space $\mathbb{P}_{k-1}$ is needed in this case, since the projection of the gradient is a polynomial of degree $k-1$. We obtain the following decomposition of the shape functions ϕ_i:

$$\begin{aligned}
\phi_i &= \Pi_k^0 \phi_i + (I - \Pi_k^0)\phi_i, \\
\phi_i &= \Pi_{k-1}^0 \nabla\phi_i + (I - \Pi_{k-1}^0)\nabla\phi_i,
\end{aligned}$$

where I is the identity operator. Both projection operators can be computed only from the knowledge of the degrees of freedom. Finally, given v_h and $u_h \in V_{k,h}^E$ we can obtain the discrete bilinear forms as

$$\begin{aligned}
a_h^E(u_h, v_h) &= (\nu \Pi_{k-1}^0 \nabla u_h, \Pi_{k-1}^0 \nabla v_h)_E + S_a^E((I - \Pi_{k-1}^0)\nabla u_h, (I - \Pi_{k-1}^0)\nabla v_h), \\
b_h^E(u_h, v_h) &= (\beta \cdot \Pi_{k-1}^0 \nabla u_h, \Pi_k^0 v_h)_E, \\
c_h^E(u_h, v_h) &= (\gamma \Pi_k^0 u_h, \Pi_k^0 v_h)_E,
\end{aligned}$$

where $S_a^E(u_h, v_h)$ is a symmetric bilinear form that scales in a desired way in order to account for the neglected contributions involving products with $(I - \Pi_k^0)$ and $(I - \Pi_{k-1}^0)$. It is called the stabilization term, and for a_h^E it is approximated by

$$S_a^E\left((I - \Pi_{k-1}^0)\nabla u_h, (I - \Pi_{k-1}^0)\nabla v_h\right) \approx \nu^E \sum_{n=1}^{N_{DOF}} \chi_n((I - \Pi_{k-1}^0)\nabla u_h)\chi_n((I - \Pi_{k-1}^0)\nabla v_h),$$

where N_{DOF} is the number of degrees of freedom of the shape functions in $V_{k,h}^E$ and χ_n is the functional that evaluates the function at its n^{th} DOF. Other stabilization terms exist for b_h^E and c_h^E, but they scale like h_E and h_E^2 respectively and are not needed for convergence purposes if the stabilization for a_h^E is present (which scales like 1). Since the discrete bilinear forms provide exact results when either u_h or v_h are polynomials, we say that it is consistent. The stabilization is then provided by S_a^E and it was proven that optimal convergence is preserved [33].

For the terms in the SUPG stabilization, we again approximate the local bilinear forms using the projection operators defined above.

$$\begin{aligned}
(-\nu^E \Delta u_h, \beta \cdot \nabla v_h)_E &\approx (\nu^E \nabla \cdot \Pi^0_{k-1} \nabla u_h, \beta \cdot \Pi^0_{k-1} \nabla v_h)_E \\
(\beta \cdot \nabla u_h, \beta \cdot \nabla v_h)_E &\approx (\beta \cdot \Pi^0_{k-1} \nabla u_h, \beta \cdot \Pi^0_{k-1} \nabla v_h)_E + \\
&\qquad S^E_\beta(\beta \cdot (I - \Pi^0_{k-1}) \nabla u_h, \beta \cdot (I - \Pi^0_{k-1}) \nabla v_h), \qquad (5.3.1) \\
(\gamma u_h, \beta \cdot \nabla v_h)_E &\approx (\gamma \Pi^0_k u_h, \beta \cdot \Pi^0_{k-1} \nabla v_h)_E, \\
(f, \beta \cdot \nabla v_h)_E &\approx (f, \beta \cdot \Pi^0_{k-1} \nabla v_h)_E,
\end{aligned}$$

where S^E_β is the stabilization of the stabilization term, and takes into account the neglected terms of the form $(I - \Pi^0_{k-1})$. In particular, we approximate it with:

$$\begin{aligned}
S^E_\beta(\beta \cdot (I - \Pi^0_{k-1}) \nabla u_h, \beta \cdot (I - \Pi^0_{k-1}) \nabla v_h) \approx \\
|\beta^E|^2 \sum_{n=1}^{N_{DOF}} \chi_n((I - \Pi^0_{k-1}) \nabla u_h) \chi_n((I - \Pi^0_{k-1}) \nabla v_h),
\end{aligned}$$

where β^E is a constant approximation of β on E. Note that when computing for the Laplacian term of the SUPG stabilization (first equation in (5.3.1)) we must compute the divergence of the projection of a shape function, since there is so far no known way of directly obtaining the projection of the Laplacian of a VEM shape function using only the information of the DOF.

Regarding the discretization in time, some results for parabolic equations using Virtual Elements can be found in [70], as well as a detailed description of the framework. We briefly recall the Crank-Nicholson scheme used in our context.

Let ϕ_i, with $1 \leq i \leq N$, be a numbering of the base functions in the global VEM space $V_{k,h}$. We want to find our discrete solution $\mathbf{u} : [0, T] \to \mathbb{R}^N$, defined by $\mathbf{u}(t) = (u_i(t))_{1 \leq i \leq N}$ in the interval $[0, T]$, with initial condition $\mathbf{u}_0 = (u_{0i})_{1 \leq i \leq N}$ and load term $\mathbf{f} : [0, T] \to \mathbb{R}^N$ defined as $\mathbf{f}(t) = (f_i(t))_{1 \leq i \leq N}$. We will denote $\mathbf{K} \in \mathbb{R}^{N \times N}$ as the stiffness matrix obtained from the Virtual Element discretization in space of problem (5.2.6), and $\mathbf{M} \in \mathbb{R}^{N \times N}$ as the mass matrix. We will compute $\mathbf{M}$ using the projector Π^0_k, by $\mathbf{M}[i, j] = (\phi_i, \phi_j) \approx (\Pi^0_k \phi_i, \Pi^0_k \phi_j)$. We are neglecting the stabilization term for this matrix since we have used the "reduced" form of the mass matrix as defined in [70], that was shown to perform similarly as the stabilized version. We arrive at the initial value problem:

$$\begin{cases}
\mathbf{M} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{K}\mathbf{u} = \mathbf{f}, \\
\mathbf{u}(0) = \mathbf{u}_0.
\end{cases}$$

The second order Crank-Nicholson scheme for a given time discretization Δt, at the step k with $k \geq 0$ is:

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}\right)\mathbf{u}^{k+1} = \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{K}\right)\mathbf{u}^{k} + \frac{\Delta t}{2}\left(\mathbf{f}(t_k) + \mathbf{t_{k+1}}\right).$$

5.4 Numerical Results

In this chapter we apply the same meshing techniques as in 3.4 and 4.5 and we refer the reader to those sections. An error convergence analysis was done on two benchmark problems to obtain results on convergence rates for the stabilized application of the method. Convergence results in a diffusion dominated regimes were presented in [33]. Random polygonal meshes were obtained using the PolyMesher algorithm [69].

The full second order equation to solve is:

$$-\nabla \cdot (\mathfrak{a}(\mathbf{x},\mathbf{y})\nabla u(\mathbf{x},\mathbf{y})) + \mathbf{b}(\mathbf{x},\mathbf{y}) \cdot u(\mathbf{x},\mathbf{y}) + c(\mathbf{x},\mathbf{y})u(\mathbf{x},\mathbf{y}) = f(\mathbf{x},\mathbf{y}). \tag{5.4.1}$$

For our application to DFNs, we are mostly interested in the advection-diffusion problem and we will consider the case with no reaction term. A non zero load term will be considered in the benchmark problem while in DFNs the forcing term will arise from non-homogeneous Dirichlet boundary conditions.

5.4.1 Benchmark problems

In this section we will consider two benchmark problems on a single fracture in the domain

$$\Omega = \left\{(x,y) \in \mathbb{R}^2 : 0 \leq x \leq 1,\ 0 \leq y \leq 1\right\},$$

and where ν will be a parameter that allows to change the relative importance of the advection term in the equation. The load term $f(\mathbf{x},\mathbf{y})$ is such that the exact solution of (5.4.1) is known and chosen beforehand. The error was obtained by comparing the exact solution to the projections of the discrete solutions, as in previous chapters (VEM shape functions are not explicitly known inside the elements). The maximum and minimum mesh Péclet (Pe_h) numbers were added for each mesh size considered.

5.4.1.1 Benchmark problem 1

The first problem is defined with constant coefficients and the range of mesh sizes cover convection-dominant and diffusion-dominant situations. The idea is to assess the method for problems with low Péclet numbers and using both definitions of the stability parameter τ (5.2.5). In a real DFN, Pe_h is expected to have a wide range of variation, since the underlying velocity field can have areas of both high and low values. The coefficients are

$$\begin{aligned}\mathfrak{a}(\mathbf{x},\mathbf{y}) =& \nu \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \\ \mathbf{b}(\mathbf{x},\mathbf{y}) =& \begin{pmatrix} 1/2 & -1/3 \end{pmatrix}, \\ c(\mathbf{x},\mathbf{y}) =& \, 0,\end{aligned}$$

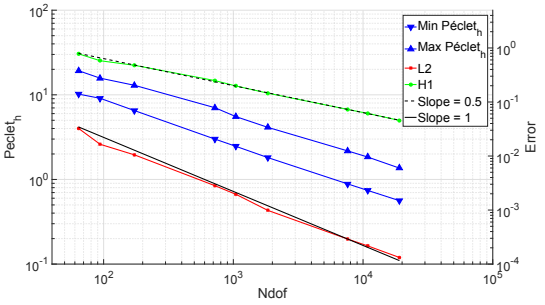
with $\nu = 10^{-3}$. The exact solution is given by

$$H(\mathbf{x},\mathbf{y}) = (65536/729)x^3(1-x)y^3(1-y).$$

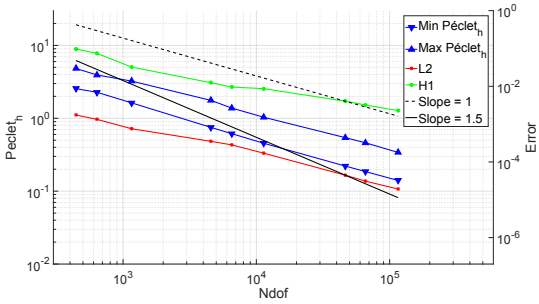
We present the convergence curves for orders of accuracy 1 to 3 (Figure 5.4.1). This problem has a Pe_h that is around 1, which is the threshold for the change in the stabilization parameter. For all orders, a perfect convergence was found for the complete stabilization. Even though the reduced stabilization shows a small absolute error, convergence is limited by the non-conformity of the stabilization scheme, and is therefore much slower than in the complete case. Nevertheless, due to its faster computation time and acceptable error values, it can be a viable alternative for solving problems with a coarse mesh.

5.4.1.2 Benchmark problem 2

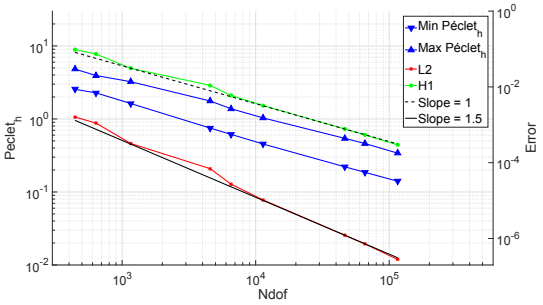
We now study a problem with non constant coefficients and a high prevalence of the convection term. The complete (5.2.4) and the reduced (5.2.6) formulations of the SUPG stabilization show similar results due to the reduced importance of the diffusion term, and therefore we will present results for two different mesh types (regular triangles and



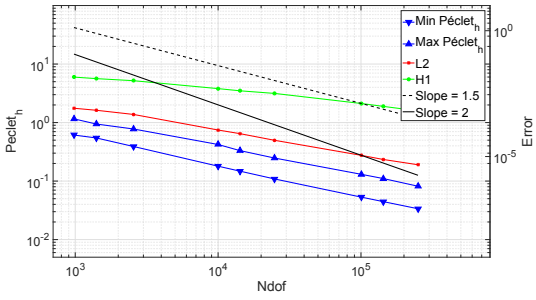
(a) Order 1



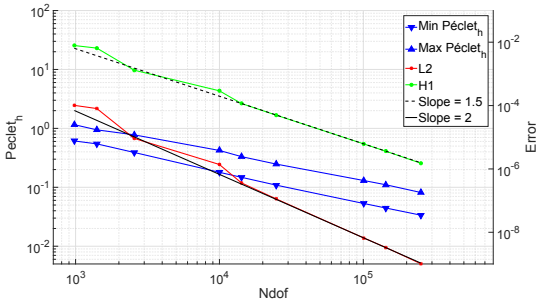
(b) Order 2 - Reduced stabilization



(c) Order 2 - Complete stabilization



(d) Order 3 - Reduced stabilization



(e) Order 3 - Complete stabilization

FIGURE 5.4.1: Convergence curves for Benchmark problem 1

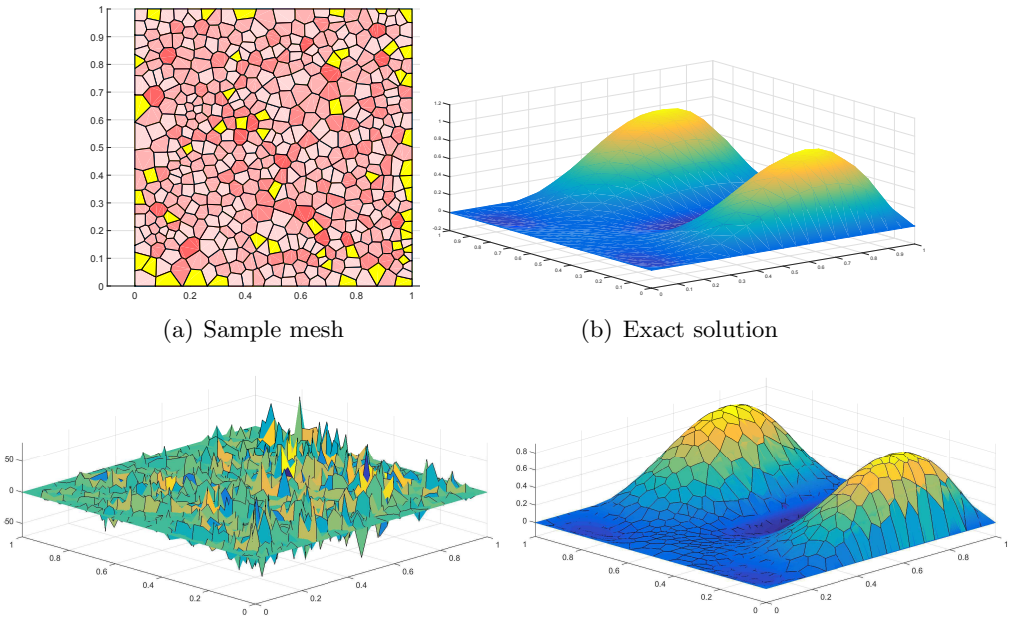
random polygons). The input data is

$$\begin{aligned}
\mathfrak{a}(\mathbf{x},\mathbf{y}) =& \nu \begin{pmatrix} 1+x^2 & xy \\ xy & 1+y^2 \end{pmatrix}, \\
\mathbf{b}(\mathbf{x},\mathbf{y}) =& \begin{pmatrix} (1/3)+10y(x+y^2)^4 & (-1/2)-5(x+y^2)^4 \end{pmatrix}, \\
c(\mathbf{x},\mathbf{y}) =& \, 0,
\end{aligned}$$

with $\nu = 10^{-7}$ and $\nabla \cdot \mathbf{b} = 0$. The exact solution is given by

$$H(\mathbf{x},\mathbf{y}) = 600\ xy(1-x)(1-y)(x-1/5)(y-2/5)(y-3/5).$$

Figure 5.4.2 shows a comparison between the unstabilized solution and the one obtained using the SUPG stabilization for second order VEM, which shows a very good



(a) Sample mesh (b) Exact solution

(c) Unstabilized solution (d) Stabilized solution

FIGURE 5.4.2: Benchmark problem 2: Sample mesh, exact, unstabilized and stabilized solutions

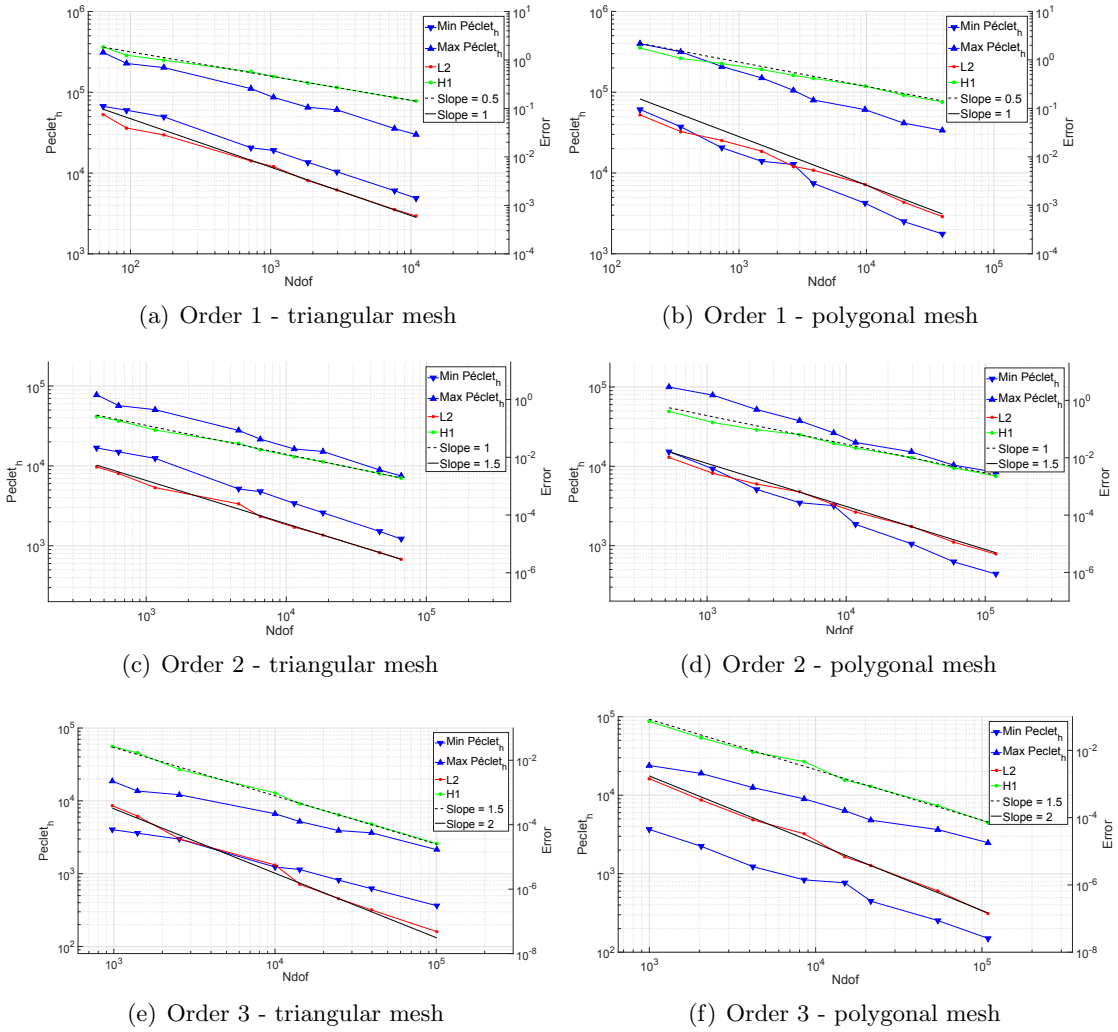
agreement with the exact solution for a given polynomial mesh. As usual, element coloring indicates number of edges. In this case we have elements with a number of edges ranging from 4 to 8.

Convergence curves were obtained for VEM formulations of orders of accuracy 1 to 3 (Figure 5.4.3). Note that for all orders and meshes, this problem is always convection-dominant ($Pe_h > 1$). For both types of meshes convergence was as expected and the absolute error were in the same range. This shows that the formulation is well suited for working in situations with a high Pe_h while retaining optimum convergence properties, and that the introduction of polygonal meshes does not affect the results significantly.

5.4.2 DFN problems

In the following, we present results obtained from applying the proposed method to DFNs. The analysis consist of two steps: We first solve the Darcy problem in the network to obtain the pressure head, as well as the velocity field. For the second step, this velocities will serve as input data for the convection term in the full transient transport equation. We will assume that we are solving the problem of the concentration of a certain solute in its medium.

Remark 5.1. At this point it would be reasonable to ask how will the problem be solved when we are dealing with a DFN and not a single fracture. In this case, any of the



(a) Order 1 - triangular mesh

(b) Order 1 - polygonal mesh

(c) Order 2 - triangular mesh

(d) Order 2 - polygonal mesh

(e) Order 3 - triangular mesh

(f) Order 3 - polygonal mesh

FIGURE 5.4.3: Convergence curves for Benchmark problem 2

developed schemes from the previous chapters (see 2.4, 3.4.2 and 4.4) will work for solving the problem on the DFN, since they only act by establishing "links" and relationships between the degrees of freedom on fractures sharing a trace and do not at all depend on how the stiffness matrices are constructed or from which type of equation they come from. The information about the DOF of the VEM space basis functions is all that is needed for the proposed schemes to work. In this chapter, all experiments were done using the conforming method (see chapter 3) for its simplicity and to avoid introducing unnecessary complications that can muddle the analysis.

5.4.2.1 7 fractures

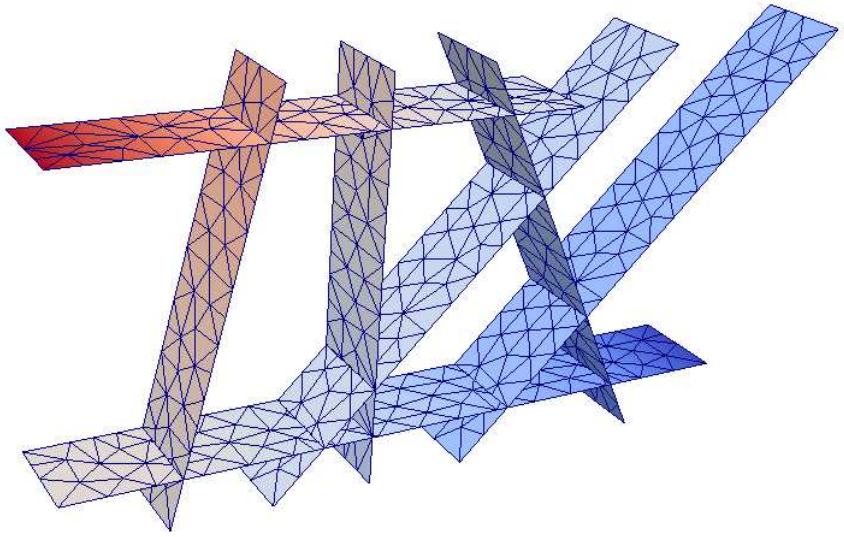
This network consisting of 7 fractures (Figure 5.4.4a), similar to one studied before in chapter 2, proves useful for the fact that the exact solution for the pressure head is made up by a union of planes and can be explicitly known. For this reason, we were able to

compare the solution by prescribing the exact velocity field and using the one obtained by solving the flow problem and experiments showed that the solutions coincided. The network has 11 traces and all of them go through the entire domain, so no trace tips are present. We prescribe an incoming unit flow in the source fracture so as to have a velocity field with speed in the range of 0 to 1. Homogeneous Dirichlet condition were imposed on an edge in the sink fracture, and for all remaining borders isolated boundaries were assigned (homogeneous Neumann). We present these results in sequence of Figures 5.4.4. With this input data for the advection term, we now solve the transient problem up with constant diffusion term $\nu = 1.10^{-5}$ and $0 \leq ||\beta||_\infty \leq 1$. The problem is clearly advection-dominant, and we solve it using linear elements. Results are shown in Figure 5.4.5. The network begins with concentration equal to 1 on the Dirichlet boundary of the source fracture and zero everywhere else.

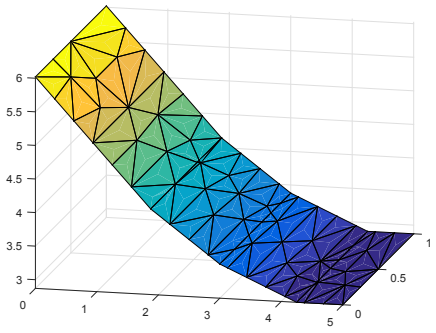
5.4.2.2 6 fractures

We now turn our attention to a smaller albeit more complicated network, that is made up of 6 fractures and 6 fractures (Figure 5.4.6). It contains trace tips and trace intersections, but since the fractures intersect perpendicularly, the angles between traces are always straight. The exact solution is not known, but due to its relative simplicity it is possible to predict a reasonable behavior of the solution. We will again solve the Darcy flow problem with a prescribed unitary Neumann condition in the source fracture, as well as one border with homogeneous Dirichlet condition. The rest of the network is isolated (homogeneous Neumann). The result is a velocity field with norm in the range of 0 to 2 (more results for the flux problem can be found in the next chapter, section 6.5.1, in the context of mixed Virtual Elements). After obtaining information for the velocity field, we impose a unitary concentration in the source fracture and solve the transient problem with and without advection terms with $\nu = 1.10^{-5}$ for the diffusivity constant. We solve the problem using quadratic elements for solving the flow problem and linear elements for the transient transport equation. We present the results for hydraulic head in Figure 5.4.6, followed by a comparison between the solutions with (Figure 5.4.7) and without convection term (Figure 5.4.8), with all other parameters equal.

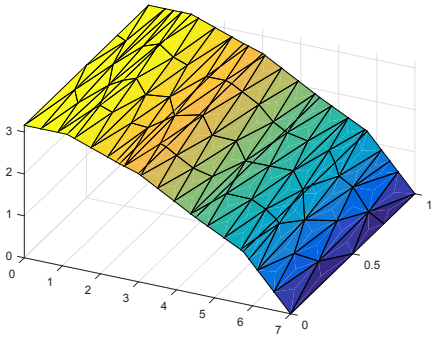
It can be clearly seen how much the convection dominates the problem from the different scales of time for the evolution of the network. Without the exact solution or physical models, it is difficult to judge the accuracy and consistency of the results. However, the results seem intuitively reasonable since the value of concentration tends to follow the the direction of the underlying velocity field.



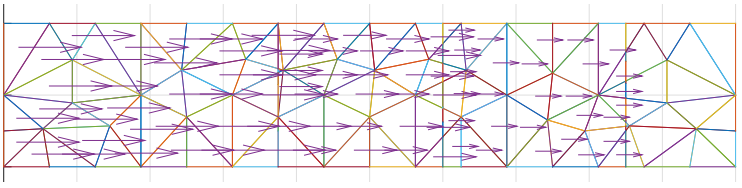
(a) Network geometry with pressure head values



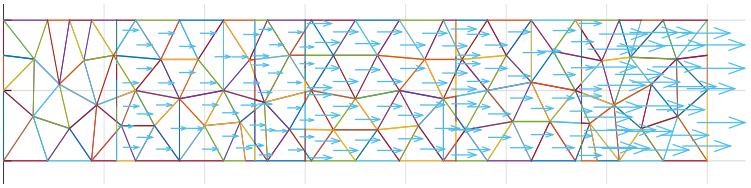
(b) Pressure head - Source fracture



(c) Pressure head - Sink fracture

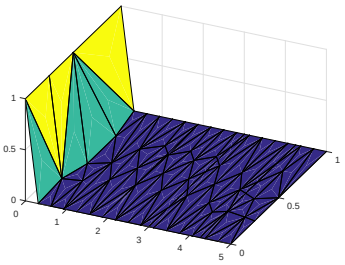


(d) Velocity field - Source fracture

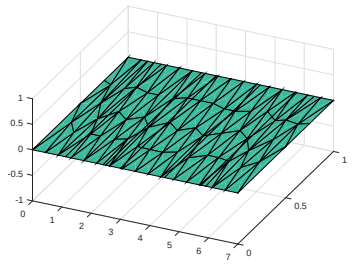


(e) Velocity field - Sink fracture

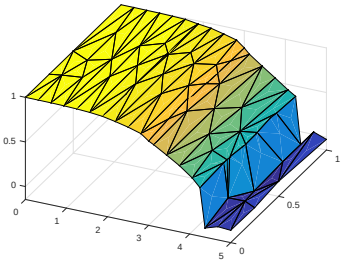
FIGURE 5.4.4: DFN7: solutions for source and sink fractures



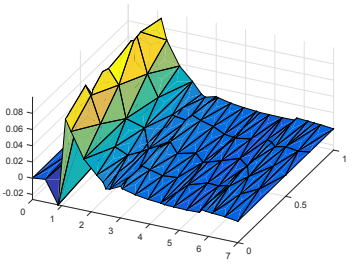
(a) Source fracture solution at $t = 0$



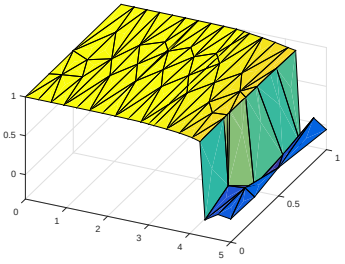
(b) Sink fracture solution at $t = 0$



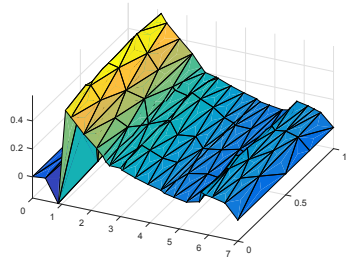
(c) Source fracture solution at $t = 6$



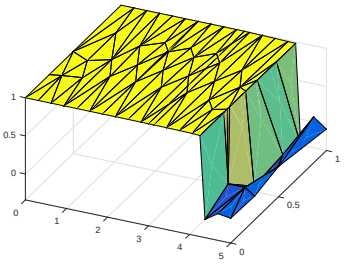
(d) Sink fracture solution at $t = 6$



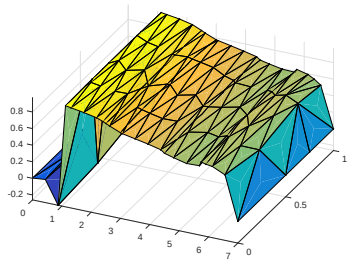
(e) Source fracture solution at $t = 12$



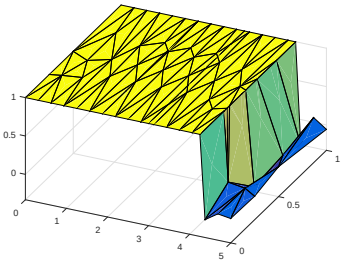
(f) Sink fracture solution at $t = 12$



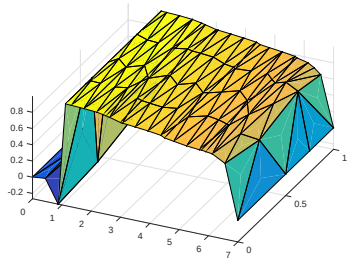
(g) Source fracture solution at $t = 24$



(h) Sink fracture solution at $t = 24$



(i) Source fracture solution at $t = 30$



(j) Sink fracture solution at $t = 30$

FIGURE 5.4.5: DFN7: evolution of the transient advection-diffusion problem

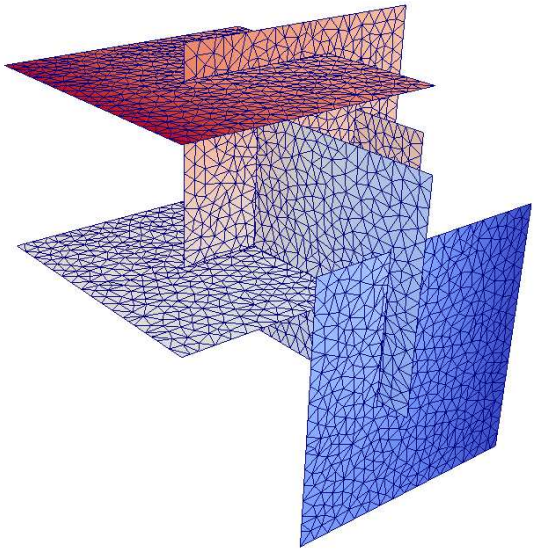
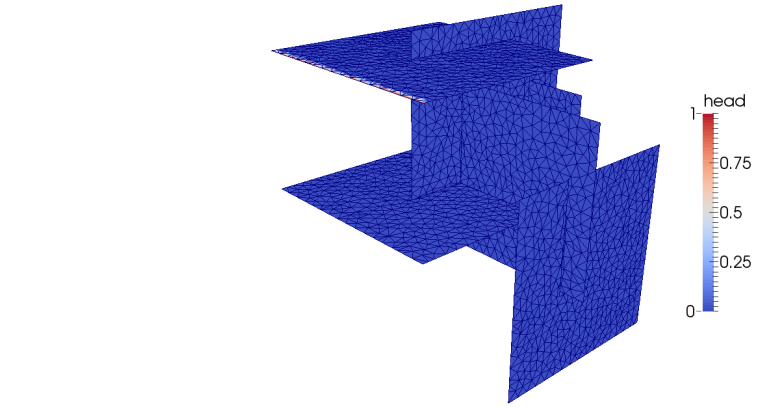
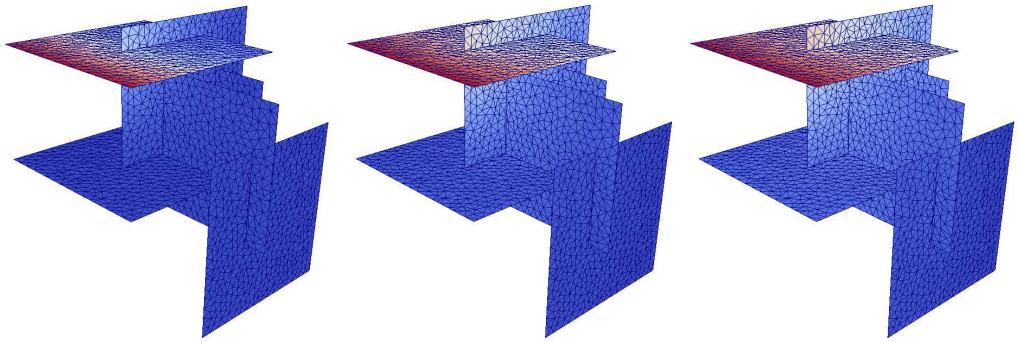


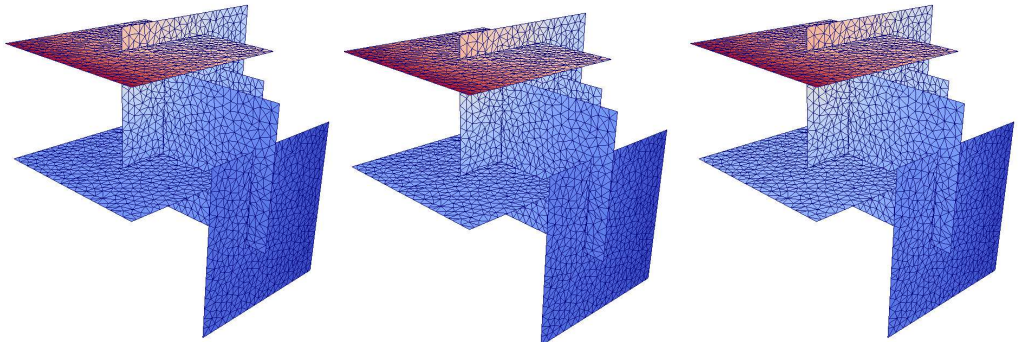
FIGURE 5.4.6: DFN6: spatial geometry with source and sink fracture



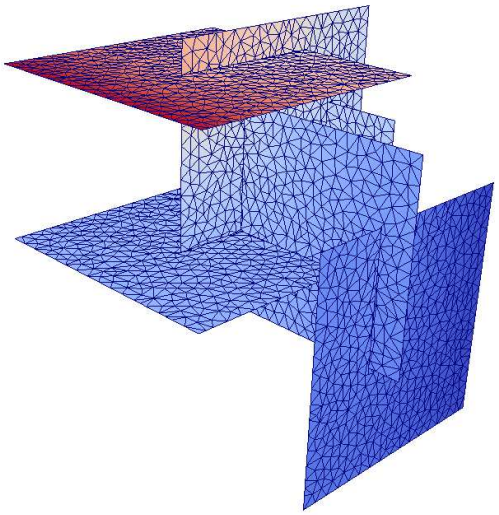
(a) $t = 0$



(b) $t = 0.6.10^6$ (c) $t = 1.8.10^6$ (d) $t = 2.4.10^6$

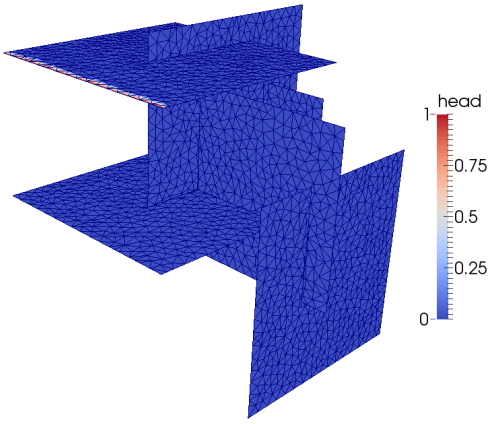


(e) $t = 3.6.10^6$ (f) $t = 4.2.10^6$ (g) $t = 5.4.10^6$

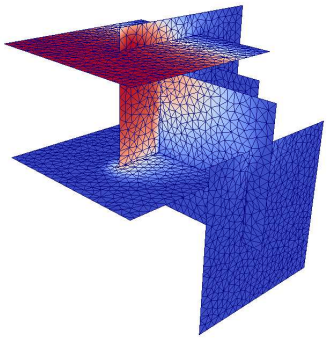


(h) $t = 6.0.10^6$

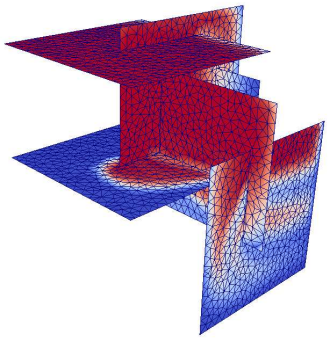
FIGURE 5.4.7: DFN6: evolution of the transient diffusion problem



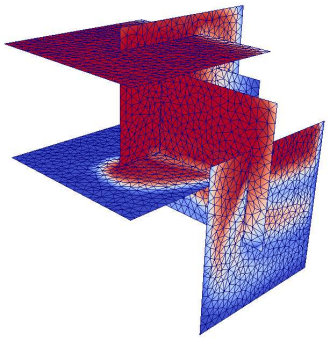
(a) $t = 0$



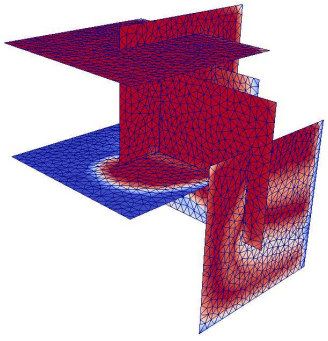
(b) $t = 6$



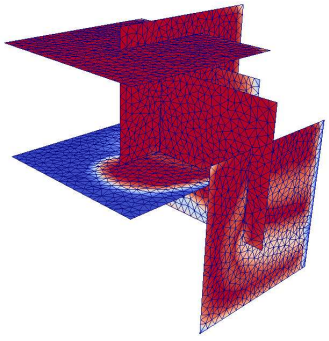
(c) $t = 18$



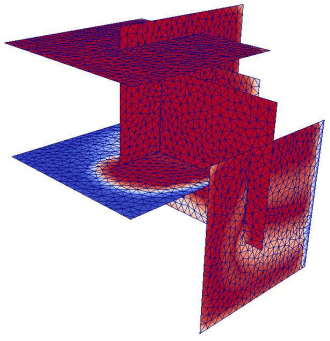
(d) $t = 24$



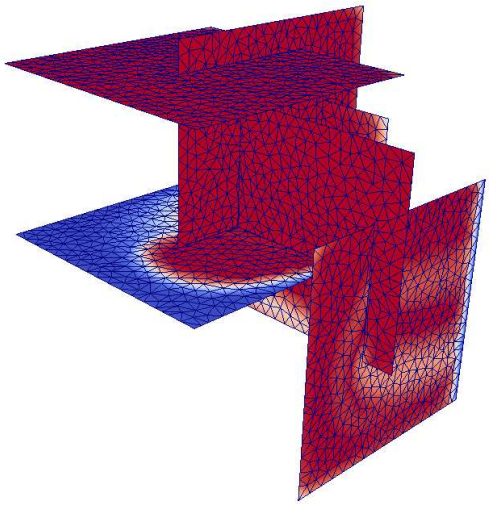
(e) $t = 36$



(f) $t = 42$



(g) $t = 54$



(h) $t = 60$

FIGURE 5.4.8: DFN6: evolution of the transient advection-diffusion problem

5.5 Conclusions

After having put forward several methods for solving flow in DFNs in the previous chapters, it was natural to approach a more generalized problem. With the flow velocity field information as input data for the advection-diffusion equation, we were able to obtain a solution for the problem of the transient advection-diffusion equation in DFNs applying some of the techniques explained previously. This can be seen seen as a very promising first step towards many useful applications and provides further validation and increases the usefulness of the proposed techniques, whose target was solving flow in DFNs but show other possible uses.

Many open problems still remain. For example, how to obtain the full SUPG stabilization by finding a way to approximate the Laplacian of a VEM shape function. Also, to explore other types of stabilizations as well as to develop a procedure to render a problem dimentionless and thus being able to introduce real physical quantities in the analysis and assess the accuracy of the model.

Chapter 6

The Mixed Virtual Element Method for Discrete Fracture Networks

6.1 Introduction

In the mixed formulation for the Finite Element Method, an extra variable is introduced for the discretization of the partial differential equation. Some ill-posed problems are better tackled using this approach. A classical book on the subject is [75], and for an introduction-level article we refer the reader to [5].

Inspired by the ideas for the globally conforming method presented in chapter 3, we aim to solve the Darcy flow in a Discrete Fracture Network (DFN). We will use a modified version of the techniques explained in that chapter to enforce flux balance along traces.

One of the advantages of using the mixed formulation is the fact that we have an explicit discretization of the Darcy velocity for the fluxes. This means that unlike in the primal formulation, we can obtain an approximation of the velocity field without using the results for the pressure head. In this way, depending on what information we want to obtain from our problem, the mixed formulation is a good alternative when our main interest is the velocity variable. Another way in which methods can complement themselves is by solving the flow problem on a DFN and obtaining results of the velocity field that will then work as input data for solving the transient transport equation (chapter 5). In any case, being a completely different approach to solving our DFN problem, the mixed formulation can also serve as a comparison parameter to evaluate the

accuracy of the other methods developed so far. We begin section 6.2 with a description of the continuous problem that we want to study, followed by the discretization with mixed Virtual Elements in section 6.3. Some notes on the implementation is given in section 6.4 and finally some numerical experiments in section 6.5.

6.2 Problem formulation

We begin by defining our domain $\Omega \subset \mathbb{R}^2$, such that $\partial\Omega = \Gamma_D \cup \Gamma_N$, that represent Dirichlet and Neumann boundaries respectively. We have a smooth positive definite symmetric tensor $\mathfrak{a} \geq \mathfrak{a}_0 > 0$, a smooth vector valued function $\mathbf{b}$ and a smooth real valued function c. We want to solve: find $P \in C^2(\Omega) \cap C(\overline{\Omega})$ such that

$$\begin{aligned} \nabla \cdot (-\mathfrak{a}(\mathbf{x})\nabla P(\mathbf{x}) + \mathbf{b}(\mathbf{x})P(\mathbf{x})) + \gamma(\mathbf{x})P(\mathbf{x}) &= f(\mathbf{x}) \quad \forall x \in \Omega, \\ P(\mathbf{x}) &= g(\mathbf{x}) \quad \forall x \in \Gamma_D, \\ \mathfrak{a}\, \partial P(\mathbf{x})/\partial \mathbf{n} &= h(\mathbf{x}) \quad \forall x \in \Gamma_N, \end{aligned} \tag{6.2.1}$$

where $f(\mathbf{x}) \in C(\Omega)$, $g(\mathbf{x}) \in C(\Gamma_D)$ and $h(\mathbf{x}) \in C(\Gamma_N)$.

We now define:

$$\begin{aligned} \mathrm{H}(\mathrm{div}, \Omega) &:= \{\mathbf{v} \in \mathrm{L}^2(\Omega) \times \mathrm{L}^2(\Omega) : \nabla \cdot \mathbf{v} \in \mathrm{L}^2(\Omega)\} \\ \mathrm{H}_{\Gamma_{N,h}}(\mathrm{div}, \Omega) &:= \{\mathbf{v} \in \mathrm{H}(\mathrm{div}, \Omega) : \mathbf{v} \cdot \mathbf{n}|_{\Gamma_N} = h\} \\ \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega) &:= \{\mathbf{v} \in \mathrm{H}(\mathrm{div}, \Omega) : \mathbf{v} \cdot \mathbf{n}|_{\Gamma_N} = 0\}, \end{aligned}$$

where $\mathbf{n}$ represents the outward unitary normal vector of Γ_D and the values on Γ_N of functions in $\mathrm{H}(\mathrm{div}, \Omega)$ are defined using density arguments of the trace operator defined on test functions, to obtain the continuous normal component trace mapping $\eta : \mathrm{H}(\mathrm{div}, \Omega) \to \mathrm{H}^{-1/2}(\Gamma_N)$. In order to deal with non homogeneous Neumann boundary conditions, we define $\mathbf{u} = \hat{\mathbf{u}} + \mathbf{u_0}$ with $\hat{\mathbf{u}} \in \mathrm{H}_{\Gamma_{N,h}}$ and $\mathbf{u_0} \in \mathrm{H}_{\Gamma_{N,0}}$. By setting $\nu := \mathfrak{a}^{-1}$, $\boldsymbol{\beta} := \nu\mathbf{b}$ and introducing the new variable $\mathbf{u} := -\mathfrak{a}\nabla P + \mathbf{b}P$ we can rewrite (6.2.1) in the classical variational form:

$$\begin{aligned} &\text{Find } \mathbf{u_0} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega) \text{ and } P \in \mathrm{L}^2(\Omega) \text{ such that} \\ &\quad (\nu\mathbf{u_0}, \mathbf{v})_\Omega - (P, \nabla \cdot \mathbf{v})_\Omega - (\boldsymbol{\beta} \cdot \mathbf{v}, P)_\Omega = -(\mathbf{v} \cdot \mathbf{n}, g)_{\Gamma_D} - (\nu\hat{\mathbf{u}}, \mathbf{v})_\Omega \ \forall \mathbf{v} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega), \\ &\quad (\nabla \cdot \mathbf{u_0}, Q)_\Omega + (\gamma P, Q)_\Omega = (f, Q)_\Omega - (\nabla \cdot \hat{\mathbf{u}}, Q)_\Omega \quad \forall Q \in \mathrm{L}^2(\Omega), \end{aligned} \tag{6.2.2}$$

where $f(\mathbf{x}) \in \mathrm{L}^2(\Omega)$, $g(\mathbf{x}) \in \mathrm{L}^2(\Gamma_D)$ and $h(\mathbf{x}) \in \mathrm{L}^2(\Gamma_N)$. A problem with pure Neumann boundary conditions requires a compatibility condition for well-posedness and the solution is only defined up to a constant for the pressure variable (if no reaction term is

present). In any case, generalizations to other boundary conditions are straightforward. Classical results for this problem hold, in the sense that for sufficiently regular data and geometry, the variational problem (6.2.2) is well posed.

6.3 Mixed Virtual Element Method

The mixed formulation for the Virtual Element Method (VEM) has been recently presented in [28], with a followup work generalizing the method in [13]. Throughout this chapter we will always refer the reader to these works as a reference for clarifications, theoretical results of error estimation and well-posedness of the problem. Due to its recent introduction, the only works regarding its application at the time of writing are [31] and [30], which deal with Stokes flow.

We first begin with a triangulation τ_h with mesh parameter h of a domain F, satisfying basic conditions of regularity [28]. The non negative integer k stands for the chosen degree of polynomial accuracy of our discretization. In broad terms, we will define a local VEM space that will be comprised by shape functions whose exact values are not known. They will be determined from a set of DOF, but using suitable projection operators it will be possible to define approximate discrete bilinear forms that will nevertheless provide the same rate of convergence as standard Finite Elements. The local VEM space for the velocity variable in an element $E \in \tau_h$ is

$$V_{k,h}^E = \{\mathbf{v}_h \in \mathrm{H}(\mathrm{div}, E) : \mathbf{v}_h \cdot \mathbf{n}|_e \in \mathbb{P}_k(e)\ \forall e \in \partial E, \quad \mathrm{div}(\mathbf{v}_h) \in \mathbb{P}_k(E),\ \text{and } \mathrm{rot}(\mathbf{v}_h) \in \mathbb{P}_{k-1}(E)\}, \tag{6.3.1}$$

and the global space is

$$V_{k,h} := \left\{\mathbf{v}_h \in \mathrm{H}(\mathrm{div}, F) : \mathbf{v}_{h|E} \in V_{k,h}^E\ \forall E \in \tau_h\right\}.$$

The condition on the rot is mainly to fix the dimension of $V_{k,h}^E$ and will not play any role in our implementation since our problem is concerned with normal components of shape functions ("face elements" in the space H(div)) and not tangential components ("edge elements" in H(rot)) [8]. The global VEM space for the pressure variable is simply

$$Q_{k,h} := \left\{Q_h \in \mathrm{L}^2(F) : Q_h|_E \in \mathbb{P}_k(E)\ \forall E \in \tau_h\right\}.$$

Note that we have made no requirements of continuity for this space. There are many possibilities for the choice of degrees of freedom (besides the main references, see also [8]). Let us first recall $\mathbb{P}_k(E)$ as the local polynomial space of order k, with dimension

TABLE 6.3.1: Dimensions for various polynomial spaces for different orders of accuracy

k	0	1	2	3
$\dim(\nabla\mathbb{P}_k^E)$	0	2	5	9
$\dim(\nabla\mathbb{P}_{k+1}^E)$	2	5	9	14
$\dim\left(\nabla\mathbb{P}_{k+1}^E\right)^\perp$	0	1	3	6
$\dim(\mathbb{P}_k(E))$	1	3	6	10
$\dim(\mathbb{P}_k(E))^2$	2	6	12	20

n_k. For the pressure space, we can trivially define the set of DOF as for example $k+1$ point values, the n_k moments with respect to the monomial basis of order k or in general any equivalent way of univocally determining a polynomial of two variables of order k. We have chosen the second option.

We then define the space

$$\nabla\mathbb{P}_{k+1}^E := \left\{\mathbf{m} \in (\mathbb{P}_k(E))^2 \text{ such that } \mathbf{m} = \nabla\hat{m} \text{ for some } \hat{m} \in \mathbb{P}_{k+1}(E)\right\}. \tag{6.3.2}$$

This space has dimension $n_k - 1$. For $k = 0$ we have that $\nabla\mathbb{P}_1 = \{0\}$. For $k = 1$, $\nabla\mathbb{P}_2 = \langle[1\ 0], [0\ 1]\rangle$. In the case of $k = 2$, a generic polynomial of degree k can be written as $\hat{m} = a + bx + cy + dx^2 + exy + fy^2$ and so $\nabla\mathbb{P}_2 = \langle[1\ 0], [0\ 1], [x\ 0], [0\ y], [y\ x]\rangle$.

For the degrees of freedom in $V_{k,h}^E$ we choose

$$\begin{aligned}
&i.\ \text{The value of } \mathbf{v}_h \cdot \mathbf{n}|_e \text{ at k+1 points on } e && \forall e \in \partial E \\
&ii.\ \int_E \mathbf{v}_h \cdot \mathbf{m}\ \mathrm{dx} && \forall m \in (\nabla\mathbb{P}_k^E) \\
&iii.\ \int_E \mathbf{v}_h \cdot \nabla m^\perp\ \mathrm{dx} && \forall m \in \left(\nabla\mathbb{P}_{k+1}^E\right)^\perp.
\end{aligned} \tag{6.3.3}$$

where $\left(\nabla\mathbb{P}_{k+1}^E\right)^\perp$ is the L^2 orthogonal complement of $\left(\nabla\mathbb{P}_{k+1}^E\right)$ in $(\mathbb{P}_k(E))^2$ so that $(\mathbb{P}_k(E))^2 = \left(\nabla\mathbb{P}_{k+1}^E\right) \oplus \left(\nabla\mathbb{P}_{k+1}^E\right)^\perp$. A proof of unisolvence can be seen in [28]. The first set of DOF can be replaced by any other way to fix a polynomial of degree k on an edge. In table 6.3.1 we present the dimensions of some of the polynomial spaces involved in the definition of the DOF. From this we can deduce that the degrees of freedom of a polygonal element of n_v vertices (or edges) is: n_v for $k = 0$, $2n_v + 2 + 1$ for $k = 1$ and $3n_v + 5 + 3$ for $k = 2$. For our purposes we will very rarely go beyond the second order, since it has been noticed before (see section 3.5) that higher orders greatly increase the number of DOF and the computational demand that is not completely justified by an increase in accuracy of the solution in the context of DFNs. As in the case of the primal VEM formulation, the shape functions in the local VEM space are not explicitly known except on its DOF. Following the same philosophy as before, we will define a projection

operator that will allow us to compute approximate discrete bilinear forms that will be stable, consistent and will retain the rate of convergence of standard finite elements. The projection operator $\mathbf{\Pi}_k^0 : V_{k,h}^E \to (\mathbb{P}_k(E))^2$ will be defined as:

$$\int_E \mathbf{\Pi}_k^0 \mathbf{v}_h \cdot \mathbf{m} \, \mathrm{dx} = \int_E \mathbf{v}_h \cdot \mathbf{m} \, \mathrm{dx} \qquad \forall \mathbf{m} \in (\mathbb{P}_k(E))^2 \tag{6.3.4}$$

It was suggested in [28] that another projector can be defined so as to work with less DOF. Not only that, but in definition (6.3.1) of the local VEM space, different integers k_e, k_d and k_r can be chosen to define $\mathbf{v}_h \cdot \mathbf{n}|_e \in \mathbb{P}_{k_e}(e)$, $\mathrm{div}(\mathbf{v}_h) \in \mathbb{P}_{k_d}(E)$ and $\mathrm{rot}(\mathbf{v}_h) \in \mathbb{P}_{k_r}(E)$ so that a new method can be developed, with some restrictions to guarantee that it retains the desired convergence properties. One possible choice would be to define the projector operator by testing it only against polynomials in $\nabla \mathbb{P}_{k+1}^E$ and not the whole $(\mathbb{P}_k(E))^2$, therefore obtaining a projection into a smaller dimensional space that can still result in a consistent and stable method.

We will now show that the knowledge of the DOF is enough to compute the projector. The left hand side of (6.3.4) is an integral between polynomials in two dimensions and can be computed. The right side however, requires more work. Since $(\mathbb{P}_k(E))^2 = \left(\nabla \mathbb{P}_{k+1}^E\right) \oplus \left(\nabla \mathbb{P}_{k+1}^E\right)^\perp$, we can find $\tilde{\mathbf{m}} \in \left(\nabla \mathbb{P}_{k+1}^E\right)$ and $\mathbf{m}^\perp \in \left(\nabla \mathbb{P}_{k+1}^E\right)^\perp$ such that $\mathbf{m} = \tilde{\mathbf{m}} + \mathbf{m}^\perp$. Thus,

$$\int_E \mathbf{v}_h \cdot \mathbf{m} \, \mathrm{dx} = \int_E \mathbf{v}_h \cdot \tilde{\mathbf{m}} \, \mathrm{dx} + \int_E \mathbf{v}_h \cdot \mathbf{m}^\perp \, \mathrm{dx} \, .$$

The second term on the right side can be obtained directly from third set of DOF. For the other term, we have that there is $\hat{m} \in \mathbb{P}_{k+1}(E)$ such that $\nabla \hat{m} = \tilde{\mathbf{m}}$ so that applying integration by parts we obtain

$$\int_E \mathbf{v}_h \cdot \tilde{\mathbf{m}} \, \mathrm{dx} = \int_E \mathbf{v}_h \cdot \nabla \hat{m} \, \mathrm{dx} = -\int_E \mathrm{div}(\mathbf{v}_h) \hat{m} \, \mathrm{dx} + \int_{\partial E} \mathbf{v}_h \cdot \mathbf{n}|_e \hat{m} \, \mathrm{ds}.$$

The second term is again computable from the DOF. If we had an explicit expression for $\mathrm{div}(\mathbf{v}_h) \in \mathbb{P}_k$ we could also compute the remaining term. In order to do so, we recall that $\mathrm{div}(\mathbf{v}_h) \in \mathbb{P}_k(E)$ so we can solve the following problem that uniquely determines $\mathrm{div}(\mathbf{v}_h)$:

$$\int_E \mathrm{div}(\mathbf{v}_h) q \, \mathrm{dx} = -\int_E \mathbf{v}_h \cdot \nabla q \, \mathrm{dx} + \int_{\partial E} \mathbf{v}_h \cdot \mathbf{n}|_e q \, \mathrm{ds} \qquad \forall q \in \mathbb{P}_k(E). \tag{6.3.5}$$

Note that using the first and second set of DOF, the right side term of (6.3.5) can be explicitly computed. Finally, we arrive at our discrete local bilinear form, defined as

$$a_h^E(\mathbf{u}_h, \mathbf{v}_h) := (\nu \mathbf{\Pi}_k^0 \mathbf{u}_h, \mathbf{\Pi}_k^0 \mathbf{u}_h)_E + S^E(\mathbf{u}_h - \mathbf{\Pi}_k^0 \mathbf{u}_h, \mathbf{v}_h - \mathbf{\Pi}_k^0 \mathbf{v}_h).$$

where S^E stands for any symmetric and definite positive bilinear form that scales like $a^E(\mathbf{u},\mathbf{v}) := (\nu\mathbf{u},\mathbf{v})$. More precisely, there exist two positive constants α_* and α^* independent of the mesh parameter such that $\alpha_* a^E(\mathbf{u},\mathbf{v}) \leq S^E(\mathbf{u},\mathbf{v}) \leq \alpha^* a^E(\mathbf{u},\mathbf{v})$ (S^E is usually taken as the standard Euclidean product). The discrete formulation of problem (6.2.2) becomes:

Find $\mathbf{u}_h \in V_{k,h}(F)$ and $P_h \in Q_{k,h}(F)$ such that

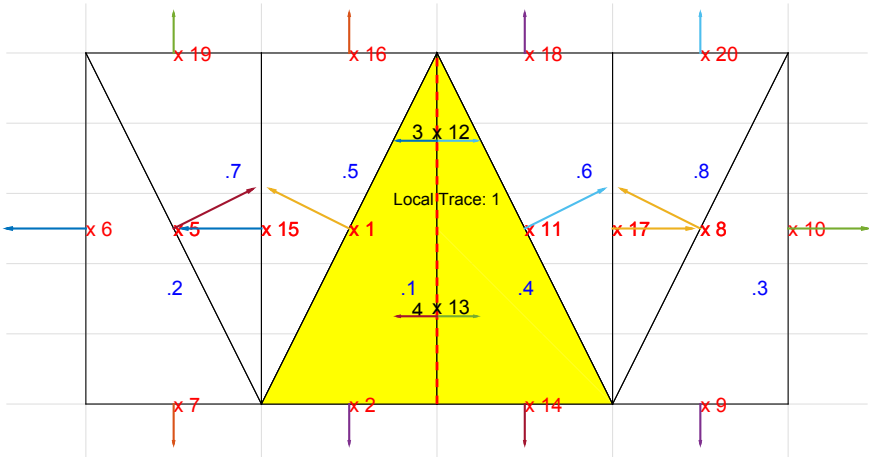
$$\begin{aligned} &a_h^E(\mathbf{u}_h,\mathbf{v}_h) - (P_h, \nabla\cdot\mathbf{v}_h) - (\boldsymbol{\beta}\cdot\boldsymbol{\Pi}_k^0\mathbf{v}_h, P_h) = 0 \quad \forall \mathbf{v}_h \in V_{k,h}(F), \\ &(\nabla\cdot\mathbf{u}_h, Q_h) + (\gamma P_h, Q_h) = (f, Q_h) \quad \forall Q_h \in Q_{k,h}(F). \end{aligned} \tag{6.3.6}$$

6.4 Implementation of mixed VEM on DFNs

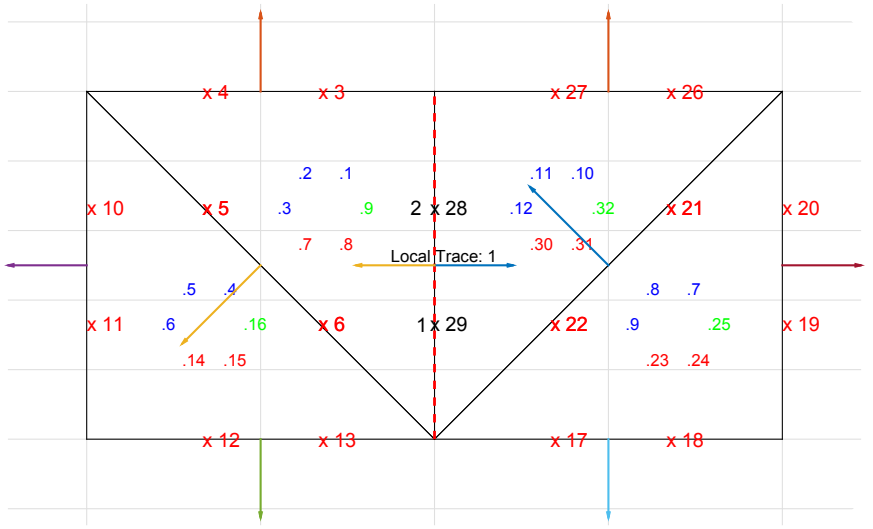
6.4.1 Meshing and Degrees of Freedom

The meshing process is done independently for each fracture. Local and global conformity are imposed to obtain the final VEM mesh. The procedure is exactly as explained in section 3.4. The main difference here comes from the definition of the degrees of freedom in edges that belong to a trace. Along those edges, there is an exchange of net flux between the two fractures whose intersection defines the trace. This means that fracture-wise, there is a jump in the normal component of the velocity field across such edges. For this reason, it would be a mistake to define only one set of DOF per edge it those cases. Therefore we propose to define two sets of DOF for every edge that belongs to a trace. In this way, two elements sharing such an edge will be able to represent a jump in the normal value of the velocity. This idea can be interpreted as having those edges on traces considered as part of the border of the domain and will be linked with their corresponding counterparts with the use of Lagrange multipliers, as explained in the following.

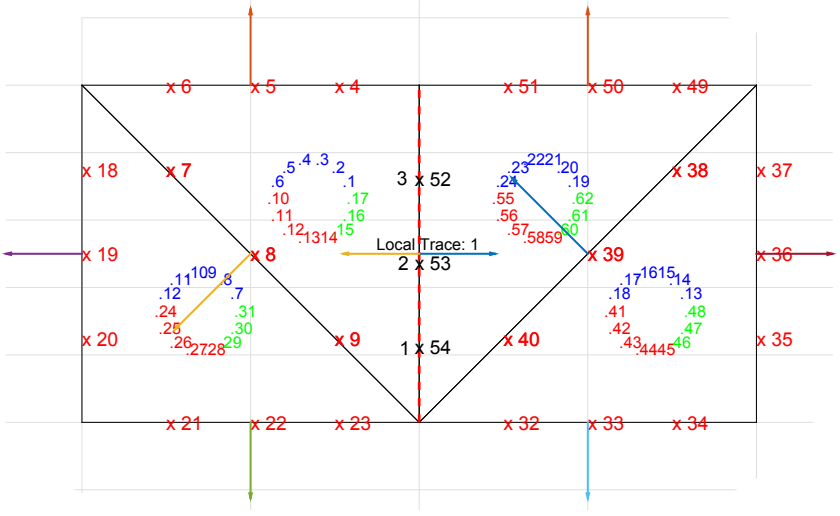
We show in Figure 6.4.1 some example meshes for mixed VEM of orders 0 to 2. The domain consists of a rectangular fracture split in the middle by a vertical trace. We show DOF on non-trace edges in red with an x, as well as internal DOF corresponding to the second set of DOF from (6.3.3) also in red with a dot. For the third set of DOF, a green with a dot is used. Note that on the edge covering the trace, there is twice as many edge DOF, represented in black with an
x. We also incorporate notation for the DOF of the pressure space, represented in blue with a dot. Finally, the arrows stand for the global orientation of each edge. An edge is positively orientated for an element if the arrow points outwards.



(a) Order 0



(b) Order 1



(c) Order 2

FIGURE 6.4.1: Degrees of freedom for mixed Virtual Elements

6.4.2 Imposing conditions on the Degrees of Freedom

In section 3.4.2 the conforming approach to imposing conditions for solving flow on a DFN was explained for the primal formulation. The case for the mixed formulation follows closely. For every fracture F_i, with $i = 1, ..., N$, we call U_{dof_i} the number of DOFs for the velocity space and P_{dof_i} the number of DOFs for the pressure space of fracture F_i. We define $n_{dof_i} := U_{dof_i} + P_{dof_i}$, so we can assemble the stiffness matrix $K_i \in \mathbb{R}^{n_{dof_i} \times n_{dof_i}}$ that arises from (6.3.6) following the procedure described in section 6.3. The structure of this matrix is

$$K_i = \begin{bmatrix} A_i & -B_i - D_i \\ D_i^T & C_i \end{bmatrix}. \tag{6.4.1}$$

where A_i, B_i, C_i and D_i stand for matrices arising from the contributions of the different terms in the variational form (6.3.6). Namely, $A_i[j,k] = a_h^E(\mathbf{u}_{h,j}, \mathbf{u}_{h,k})$, $B_i[j,l] = (\boldsymbol{\beta} \cdot \boldsymbol{\Pi}_k^0 \mathbf{u}_{h,j}, Q_{h,l})$, $C_i[l,m] = (\gamma Q_{h,l}, Q_{h,m})$ and $D_i[j,k] = (\nabla \cdot \mathbf{u}_{h,j}, Q_{h,l})$, with $1 \leq j,k \leq U_{dof_i}$, $1 \leq l,m \leq P_{dof_i}$ where $\mathbf{u}_{h,j}$ and $Q_{h,l}$ are the basis functions of the finite-dimensional global discrete spaces for the velocity and the pressure variable respectively. Then we construct the column vectors $f_i \in \mathbb{R}^{n_{dof_i}}$ as the vector of load values (including terms arising from non-homogeneous boundary conditions) and $h_i := (u_i, Q_i)$ as the vector of values of the discrete solution for the velocity variable and the pressure variable. We note that the matrix K_i is singular for fractures with pure Neumann boundary conditions whenever $\gamma = 0$ (no reaction term). For the complete DFN we have:

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

In order to obtain the linear system for the complete DFN we have to impose matching conditions for the DOF on the traces that guarantee flux/mass balance on the traces. In chapter 3, we introduced Lagrange multipliers to enforce equality between DOFs on nodes on the same trace but on different fractures that occupy the same point in space. In the current scenario and for an order of polynomial accuracy k, the DOF on traces represent the pointwise values of the VEM shape functions in the direction of the normal vector of the edge, i.e., $\mathbf{u}_h \cdot \mathbf{n}_e \in \mathbb{P}_k(e)$ where n_e is the normal of the edge e. All DOF on traces were assigned an orientation pointing from inside the element and in the direction of the trace, so as to have, by definition, flux leaving a trace for a positive value of the DOF. In this way we also have that for all DOF on the same edge and for both fractures sharing that trace, outgoing flux will have the same sign.

Given a numbering of all the edges on traces (e_t) indexed by $t = 1, ..., \ldots, T$ where T is the total number of edges on traces, we have that for each t, we can define the only two indices $i(t), j(t)$ such that $e_t \in F_{i(t)}$ and $e_t \in F_{j(t)}$. The edge e_t will have $2(k+1)$ DOF on fracture $F_{i(t)}$ and the same for fracture $F_{j(t)}$, because edges on trace have twice as many DOF as internal edges as explained in the previous section. We will call the former $D_{t,i}$ and the latter $D_{t,j}$, and $\#D_{t,i} = \#D_{t,i} = 2(k+1)$. We can split these set of DOF, using the criteria of the relative position with respect to the edge e_t. Since each edge has a fixed global orientation, we can define $D_{t,i} = L_{t,i} \cup R_{t,i}$ (analogously for $D_{t,j}$), which collect the DOF to the left and to the right of the trace respectively. We have that $\#L_{t,i} = \#R_{t,i} = k+1$. Therefore, for each edge we will establish $k+1$ conditions that will link the pointwise values of the DOF from $F_{i(t)}$ and $F_{j(t)}$ using Lagrange multipliers. Then, for each t, we define the row vectors $\mathfrak{L}_{t_q} \in \mathbb{R}^{1 \times \sum_i n_{dof_i}}$ as:

$$\mathfrak{L}_{t_q} := \Big(0 \quad \cdots \quad \overset{L_{t,i}(q)}{1} \quad \cdots \quad \overset{R_{t,i}(q)}{1} \quad \cdots \quad \overset{L_{t,j}(q)}{1} \quad \cdots \quad \overset{R_{t,j}(q)}{1} \quad \cdots \quad 0 \Big),$$

with $q = 1, ..., k+1$, and $L_{t,i}(q)$ is the q^{th} DOF on edge e_t in fracture $F_{i(t)}$ located to the left of the trace (analogously for the others). The sets of DOF are numbered in such a way that for the same q, $L_{t,i}(q)$, $R_{t,i}(q)$, $L_{t,j}(q)$ and $R_{t,j}(q)$ all represent pointwise values on the same point on the trace (albeit on two different fractures).

Finally, we set $\mathfrak{L} \in \mathbb{R}^{T(k+1) \times \sum_i n_{dof_i}}$ as the matrix:

$$\mathfrak{L} = \begin{pmatrix} \mathfrak{L}_{1_1} \\ \vdots \\ \mathfrak{L}_{1_{k+1}} \\ \mathfrak{L}_{2_1} \\ \vdots \\ \mathfrak{L}_{T_{k+1}} \end{pmatrix}.$$

The final linear system is:

$$\begin{bmatrix} K & \mathfrak{L}^T \\ \mathfrak{L} & 0 \end{bmatrix} \begin{bmatrix} h \\ \lambda \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{6.4.2}$$

This type of system falls into the study of domain decomposition methods for mixed finite elements. We refer the reader to [4, 7, 43, 74].

6.5 Numerical Results

Convergence results for single domain problems were presented in [13]. The purpose of this section is to study flux conservation and distribution, so we will again consider the problem as described before in section 3.2, i.e., we will only solve the with the Poisson problem and not include neither convection nor reaction terms.

6.5.1 DFN6

For the first DFN that we are studying, we will revisit a 6 fracture and 6 trace network, as in section 5.4.2.2. The spatial geometry is shown again, with new references, in Figure 6.5.1. We will impose a unitary incoming flow in Fracture F_1 over an edge of approximate length 5.3174, which means that the network should have a total flux of corresponding to that value. F_5 will be the sink fracture, with homogeneous Dirichlet boundary conditions on one edge. All other borders will be taken as isolated. We show results for mixed Virtual Element from orders 0 to 2 in Figures 6.5.2 and 6.5.3. The visualization for the velocity field was practically equal for different orders, so only one is shown. Note that pressure head values are also quite similar and it is interesting to note that even though there is no condition for pressure head continuity imposed on the formulation, the solutions for orders 1 and 2 nevertheless exhibit a practically continuous solution. This is a good indication, since the solution of the problem outside of traces is a harmonic function (and therefore smooth). In Figure 6.5.4 we present the results for the flux along trace T_4 and a comparison between different orders. In all cases, the approximation is very good. As expected, second order elements are able to better reproduce details in the flux function, but it is a remarkable result that in all cases the same flux is obtained when integrating the complete profile.

Table 6.5.1 details the flux exchange in fractures and traces where rows correspond to traces and columns to fractures. The last row contains the sum of all the incoming and outgoing flux for each fracture, while the last column shows the balance in flux exchange between the two fractures that share a trace. An almost exact balancing of fluxes was obtained, both within fractures and in trace exchanges, for all orders of accuracy. The incoming flux from fracture F_1 is perfectly balanced with the outgoing flux in F_5. Fracture F_1 acts as a source that provides 5.3174 of flux to the system from its Neumann edge (negative values represent flux leaving the fracture through the traces), which leaves the system at fracture F_5 with a 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. On a more curious note, the flux exchange in trace T_3 is almost zero, because fracture F_4 is neither a source nor a sink

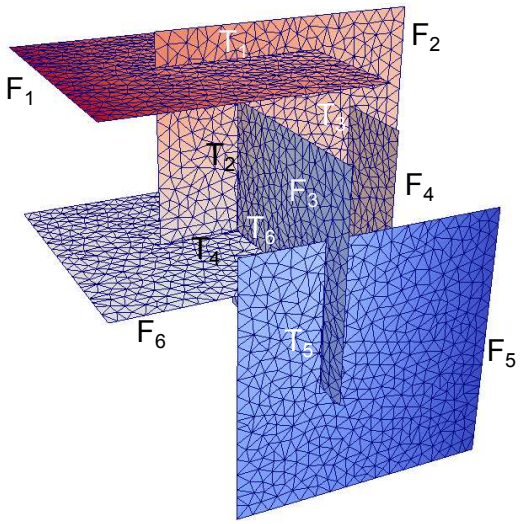
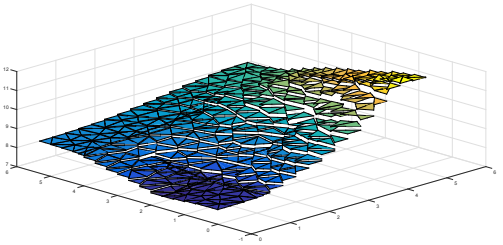
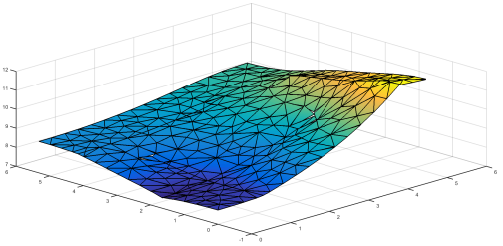


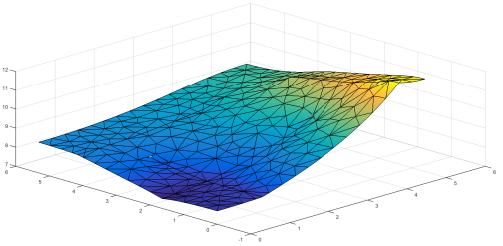
FIGURE 6.5.1: DFN6: spatial geometry with source and sink fracture



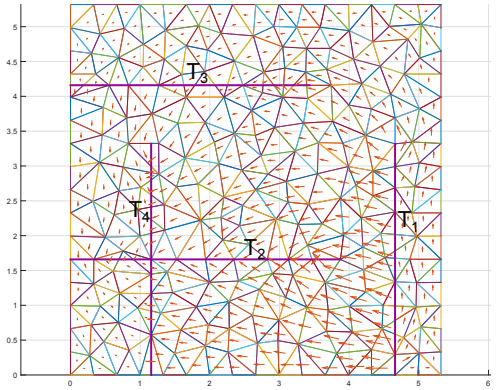
(a) Order 0 - F_2



(b) Order 1 - F_2

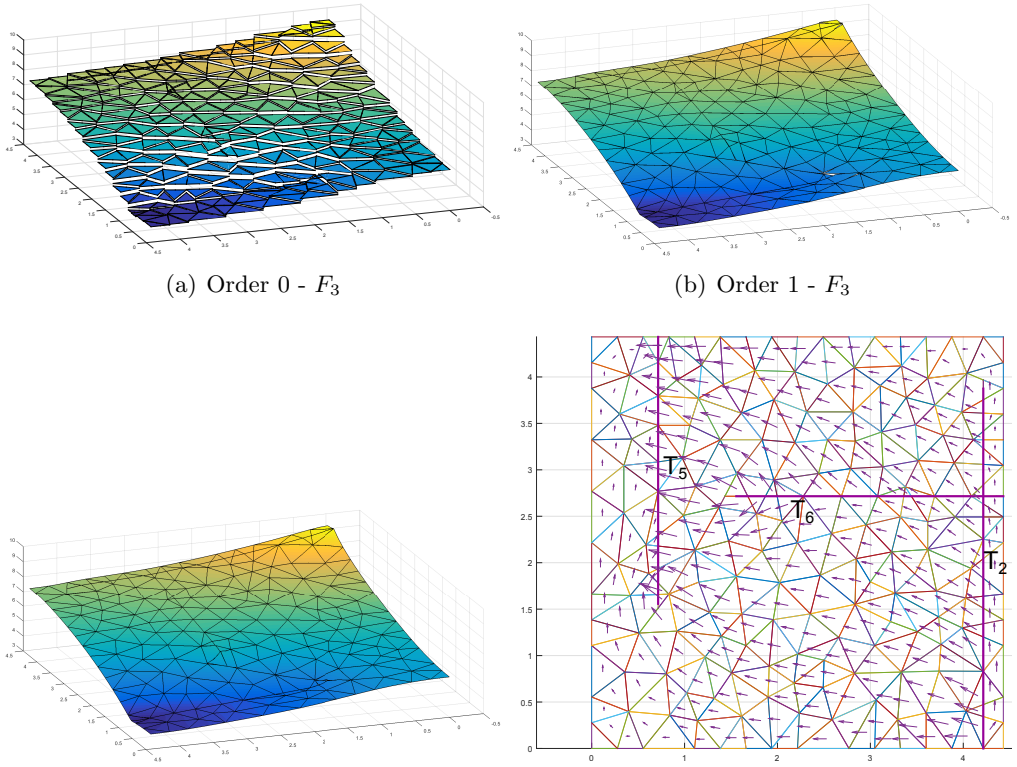


(c) Order 2 - F_2



(d) Velocity field

FIGURE 6.5.2: DFN6: pressure head solutions for fracture $F2$



(a) Order 0 - F_3 (b) Order 1 - F_3

(c) Order 2 - F_3 (d) Velocity field

FIGURE 6.5.3: DFN6: pressure head solutions for fracture $F3$

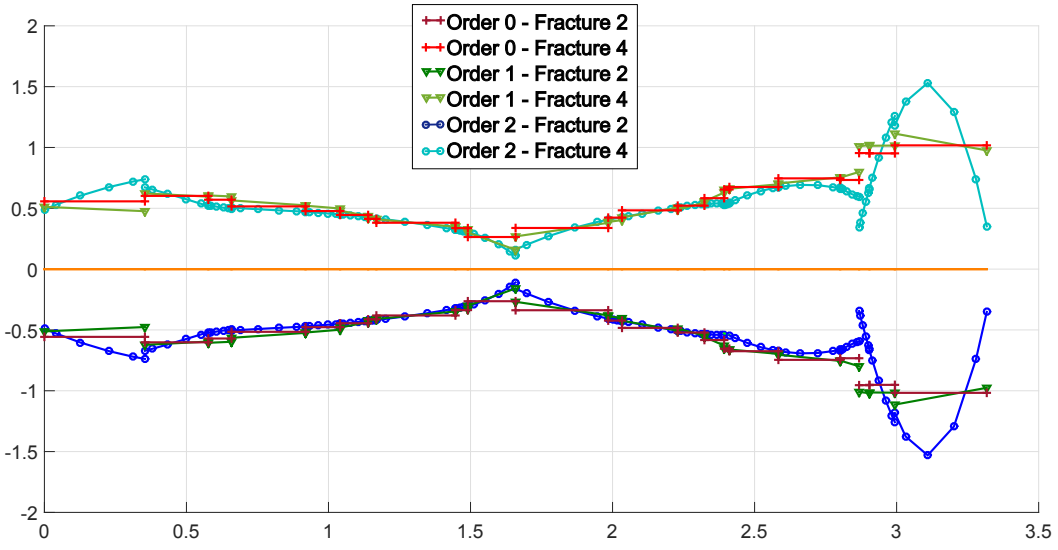


FIGURE 6.5.4: DFN6: flux on trace T_4 for orders 0 to 2

TABLE 6.5.1: Flux data for the DFN6 configuration with flux mismatches across traces (last column) and flux balance on fractures (last row) for order 0

DFN6							
	F1	**F2**	**F3**	**F4**	**F5**	**F6**	**Balance**
T1	-5.3174	5.3174	0	0	0	0	0
T2	0	-3.4147	3.4147	0	0	0	0
T3	0	1.7509e-11	0	-1.7509e-11	0	0	0
T4	0	-1.9026	0	0	0	1.9026	0
T5	0	0	-5.3174	0	5.3174	0	0
T6	0	0	1.9026	0	0	-1.9026	0
Balance	-5.3174	-2.66e-15	-2.05e-13	-1.75e-11	5.3174	-7.8e-14	0

fracture, and since it has an intersection with only one other fracture (F_2), any incoming flux must leave the fracture so as to preserve the balance. This means that there should not be any "net flux", but it does not mean however that the value of the flux is pointwise zero. We show the value of the flux on trace T_3 in Figure 6.5.6.

From this results, we can conclude that the mixed formulation showed a very high accuracy when computing flux exchange between fractures. The flux balance is almost exact for any order of accuracy, which is reminiscent of the precision of the continuity of the hydraulic head of the conforming method for the primal formulation in chapter 3. Low order elements also show a very good performance with much less computational demand.

6.5.2 DFN116

We will study a bigger network consisting of 116 fractures and 256 traces (Figure 6.5.7 and section 3.5.3). It has a wider range of fracture sizes and they intersect at various different angles. As before, we prescribe the same boundary conditions and the unitary incoming flux on the source fracture gives rise to a total of 53.1736 of prescribed incoming flux.

In Figures 6.5.8 and 6.5.9 we show results for orders 0 and 2 for the source and sink fractures respectively. The number of DOF was 68178 for order 0 and 433433 for order 2. Using only one sixth of the DOF, we can see that the results for order 0 is qualitatively very similar to the second order results. The flux balance results for both orders again show a perfect match between incoming and outgoing flow in the source and sink fractures, with a mismatch of machine-error size for all trace balances and fracture balances. When comparing specific flux exchanges, even though the sum of all fluxes on a fracture do not change by changing the order of the approximation, some of the individual fluxes do show some differences. In Table 6.5.2 we present the results for flux balance on traces belonging to the sink and source fracture. As mentioned before,

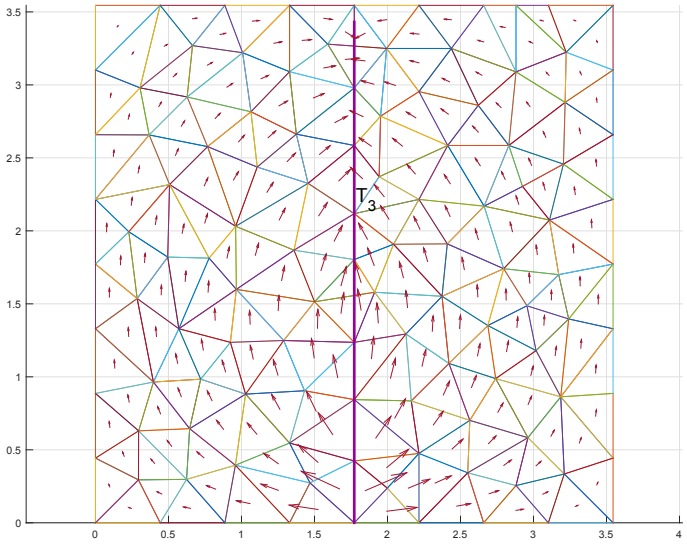


FIGURE 6.5.5: DFN6: velocity field on fracture F_4

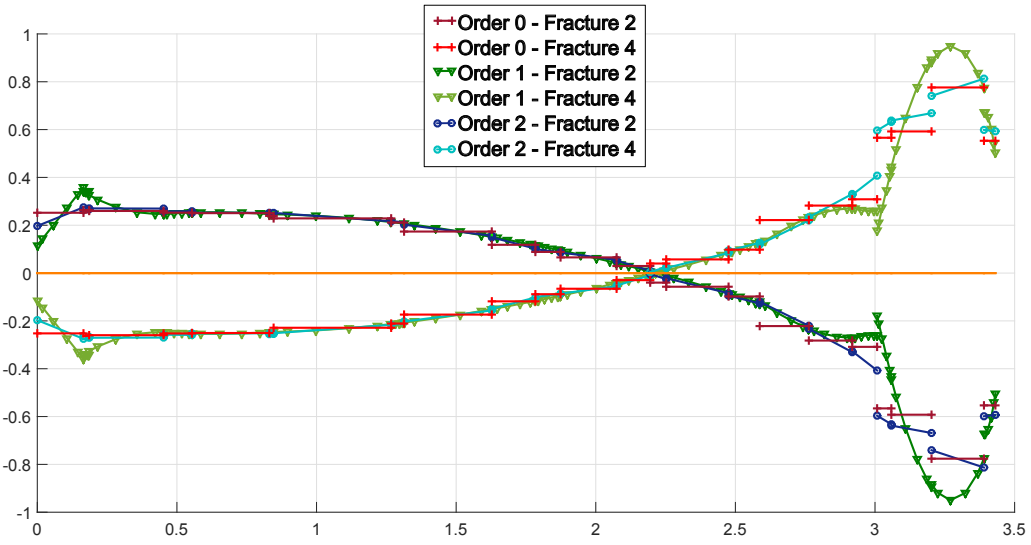


FIGURE 6.5.6: DFN6: flux on trace T_3 for orders 0 to 2

the balance is always preserved but some differences were appreciated, which is to be expected since we are using piecewise constant approximations for the flux for one case, and piecewise quadratic for the other. Nevertheless, the absolute difference (Δ) is small and of course in both cases it is expected to converge to the exact solution for finer meshes.

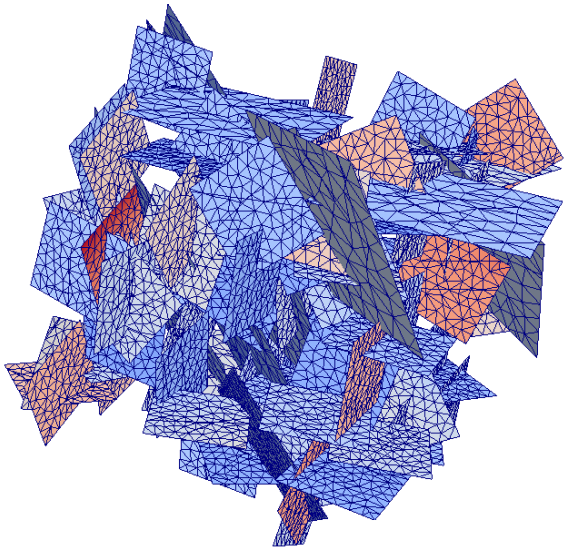
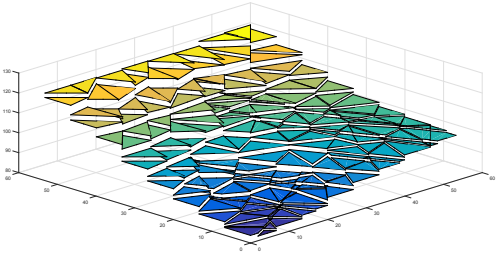
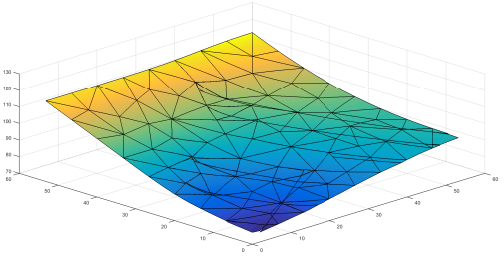


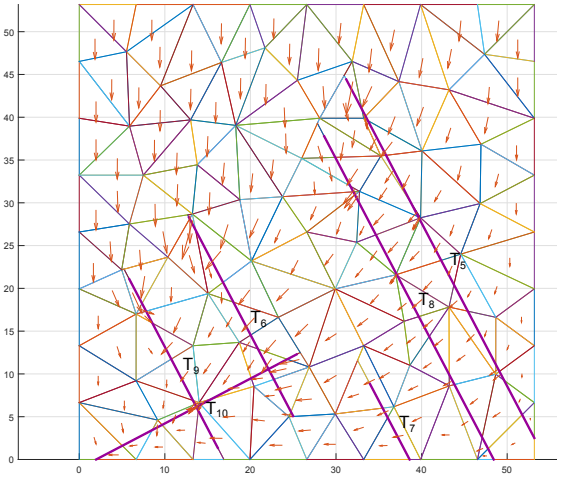
FIGURE 6.5.7: DFN116: spatial geometry



(a) Order 0 - Pressure head

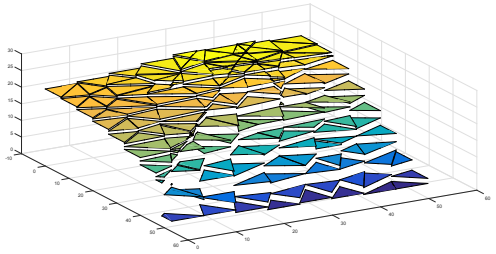


(b) Order 2 - Pressure head

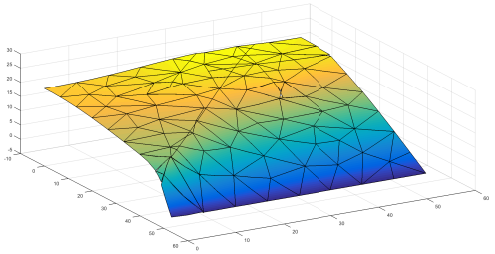


(c) Order 0 - Velocity field

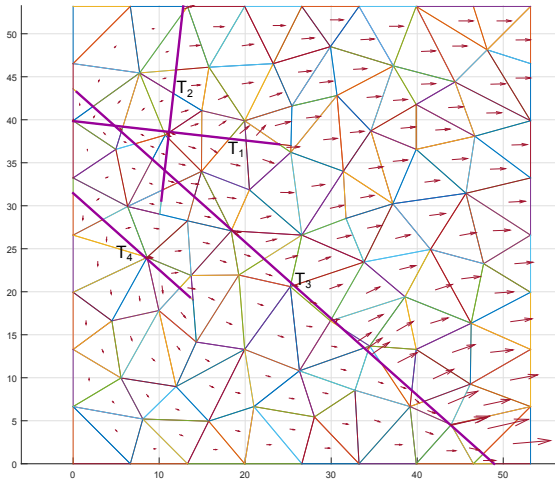
FIGURE 6.5.8: DFN116: solutions for source fracture



(a) Order 0 - Pressure head



(b) Order 2 - Pressure head



(c) Order 0 - Velocity field

FIGURE 6.5.9: DFN116: solutions for sink fracture

TABLE 6.5.2: Flux data for the DFN116 configuration with flux exchange across traces and comparison between solutions of different orders

	Order 0		Order 2		Δ
	Source	Sink	Source	Sink	
T_1		7.379		7.288	0.0913
T_2		5.868		5.419	0.449
T_3		33.683		33.983	-0.300
T_4		6.243		6.484	-0.240
T_5	-5.773e-15		-7.225e-13		7.168e-13
T_6	-15.918		-15.486		-0.432
T_7	-1.316		-1.591		0.275
T_8	-9.351		-9.835		0.484
T_9	-12.450		-12.877		0.427
T_{10}	-14.139		-13.384		-0.754
Balance	-53.174	53.174	-53.174	53.174	-3.470e-09

6.6 Conclusions

In this chapter we presented results for the application of the mixed formulation of the Virtual Element Method. All of the advantages of locally independent meshing are preserved. The solution of the problem in DFNs was obtained with a modified version of the procedure of domain decomposition from a globally conforming mesh, as introduced in the primal conforming case. The independent computation of the velocity variable allows for a more accurate discretization, is a good complement to the primal formulation and can serve as a way of providing accurate velocity fields without recomputing the mesh from one problem to the other. Flux balnaces are preserved almost exactly, far more accurate than any of the methods used for the primal formulation. This implementation may be the best choice from all the ones we have covered when solving flow in a DFN and our interest is computing flux exchanges. Pressure head results are also very satisfying. It is quite remarkable that the lowest order implementation already provides a very good approximation of the desired results with very few DOF per element, which is a very good prospect if much bigger networks are to be solved. We have shown in a medium sized DFN how flux balance is always preserved and the similarity between results obtained using elements of different orders. Preliminary experiments suggest that for future developments a mortar base could be introduced along traces which, in a dual way with respect to the primal formulation, will provide and approximation of the pressure head on the traces.

Chapter 7

Final remarks

At the beginning of this work we set out to develop an application of the very recent Virtual Element Method (VEM) for solving the pressure head problem on a system of discrete planar fractures. The main obstacle when dealing with Discrete Fracture Networks (DFNs) is their geometrical complexity. Since they are generated randomly, there is no way to predict how the fractures will intersect each other, how many traces there will be nor the angle between them. For this reason, we have developed a number of different techniques whose common factor is always the advantage of being able to perform a fracture-local meshing. This means that we can effectively tackle the biggest difficulty of our problem right away. Based on this premise, we have put forward several choices for solving problems on DFNs.

The first one (chapter 2) is an optimization approach, in which we defined a control variable that represented the flux on traces to solve a quadratic functional using iterative methods. It was an idea that was implemented before with eXtended Finite Elements (XFEM), which have the advantage of added base functions that allow for an accurate representation of jumps in derivatives of the solution. We showed one the earliest applications of the VEM as we profited from the easily obtained local conformity to avoid the use of XFEM, while still preserving the ability to represent the change in the conormal derivative of the pressure head along traces.

Afterwards, chapter 3 was dedicated to generalizing the classical approach for solving DFNs with standard finite elements, which is through global conformity. It was described in detail how we can obtain a globally conforming mesh without sacrificing the biggest advantage provided by the VEM, which is fracture-independent meshing. Convergence results were presented for different orders of accuracy and sample DFNs were solved. We also hinted at some of the issues that affect high order implementations,

as well as giving an idea on the use of FETI domain decomposition methods to speed up the implementation.

We introduce a new approach in chapter 4 using the well kwown Mortar method. We proposed a hybrid method that only imposes weak continuity of pressure head along traces, and therefore it is enough to work with local conformity. The results for pressure head were very similar to those of the conforming method, but the added degrees of freedom represented by the Lagrange multiplier provided a direct computation of the flow variable on traces.

With all these possibilities at hand, in chapter 5 we took the first steps into tackling more complicated problems of DFNs, namely the full second order elliptic equation. In particular, we presented some results for solving the transport equation on a DFN and the numerical experiments show that the proposed techniques developed in this work can be applied to this case. While many details still remain unresolved, the feasibility of expanding the approaches for solving DFNs to more complex problems was established.

We took a different approach in chapter 6, by introducing the mixed Virtual Element Formulation. Borrowing ideas from the globally conforming method, local and global conformity allowed us to discretize and solve the problem of Darcy flow on a DFN. It can prove to be an appealing complement for the primal formulation in some situations; for example, when we want an accurate description of the velocity field or some problems that are inherently better conditioned to be solved with the added velocity variable of the mixed formulation. Flux results were shown to be quite reliable, which is a clear advantage over primal formulation methods. Despite its very recent introduction creation in the literature, it already appears as a very viable possibility in the future, once more theoretical results become available. In particular, more background regarding time dependent problems and numerical schemes for stability in convection-dominant problems could open the door to many interesting applications.

One possible path for future work is to add a compatibility feature that will allow for the interaction of the proposed methods for solving the same DFN. That is, the prospect of taking a network and separating it into different subnetworks according to the information we want to obtain from its analysis. Each subnetwork can then be solved with any method of our choosing, that will then interact to provide the final solution. This has many potential uses; for instance, using the mixed formulation for subnetworks where flux balance is more important, and using the primal globally conforming formulation on areas of the network where pressure head continuity on traces must absolutely be preserved (and a rough approximation of the velocity variable is considered acceptable). The Mortar method can be applied for parts where only the velocity variable on traces is needed, and a good approximation of the pressure head is required. Finally,

with the optimization method all these subresults can interact to provide the complete solution of the complete problem.

In summary, our intention was to introduce the VEM to solving DFNs. We have shown how the new capabilities that this new method offers lend themselves perfectly for the resolution of DFNs by greatly simplifying the meshing requirements. It also shows a lot of promise for further developments: to acknowledge the surrounding rock matrix in terms of the physical and numerical models, optimization of the implementation through the use of parallel solvers, the introduction of stochastic and probabilistic components inherent to the uncertainties of DFNs and to calibrate our computation with real physical data. It was our hope that this work could serve as an entrance to a new framework for studying these types of problems.

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] P. F. Antonietti, L. Beirão da Veiga, D. Mora, and M. Verani, *A stream Virtual Element formulation of the Stokes problem on polygonal meshes*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 386–404.

[4] T. Arbogast, L. C. Cowsar, M. F. Wheeler, and I. Yotov, *Mixed Finite Element Methods on nonmatching multiblock grids*, SIAM Journal on Numerical Analysis, 37 (2000), pp. 1295–1315.

[5] D. N. Arnold, *Mixed Finite Element Methods for elliptic problems*, Computer Methods in Applied Mechanics and Engineering, 82 (1990), pp. 281–300.

[6] B. Ayuso de Dios, K. Lipnikov, and G. Manzini, *The nonconforming Virtual Element Method*, ArXiv, (2014).

[7] I. Babuška and R. Narasimhan, *The Babuška-Brezzi condition and the patch test: an example*, Computer Methods in Applied Mechanics and Engineering, 140 (1997), pp. 183 – 199.

[8] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *H(div) and H(curl) - conforming VEM*, preprint arXiv:1407.6822, (2014).

[9] L. Beirão da Veiga, K. Lipnikov, and G. Manzini, *The Mimetic Finite Difference Method for Elliptic Problems*, vol. 11 of Modeling, Simulation & Applications, Springer, 2014.

[10] 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.

[11] 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.

[12] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *The Hitchhiker's Guide to the Virtual Element Method*, Mathematical Models and Methods in Applied Sciences, 24 (2014), pp. 1541–1573.

[13] ——, *Mixed Virtual Element Methods for general second order elliptic problems on polygonal meshes*, ESAIM: M2AN, (2015).

[14] ——, *Virtual element method for general second-order elliptic problems on polygonal meshes*, Mathematical Models and Methods in Applied Sciences, 26 (2016), pp. 729–750.

[15] L. Beirão da Veiga and G. Manzini, *A Virtual Element Method with arbitrary regularity*, IMA Journal of Numerical Analysis, 34 (2014), pp. 759–781.

[16] F. B. Belgacem, *The Mortar Finite Element Method with Lagrange multipliers*, Numerische Mathematik, 84 (1999), pp. 173–197.

[17] M. F. Benedetto, S. Berrone, S. Pieraccini, and S. Scialò, *The Virtual Element Method for Discrete Fracture Network simulations*, Computer Methods in Applied Mechanics and Engineering, 280 (2014), pp. 135 – 156.

[18] C. Bernardi, Y. Maday, and A. T. Patera, *A new nonconforming approach to domain decomposition: the Mortar Element Method*, in Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991), vol. 299 of Pitman Res. Notes Math. Ser., Longman Sci. Tech., Harlow, 1994, pp. 13–51.

[19] S. Berrone, C. Canuto, S. Pieraccini, and S. Scialò, *Uncertainty quantification in Discrete Fracture Network models: Stochastic fracture transmissivity*, Computers & Mathematics with Applications, 70 (2015), pp. 603–623.

[20] 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.

[21] 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.

[22] 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.

[23] S. Berrone, S. Pieraccini, S. Scialò, and F. Vicini, *A parallel solver for large scale DFN simulations*, SIAM Journal on Scientific Computing, 37 (2015), pp. C285–C306.

[24] R. P. Bonet Chaple, *Numerical stabilization of convection-diffusion-reaction problems*, Technical report, Delft University of Technology, (2006).

[25] F. Brezzi, *On the existence, uniqueness and approximation of saddle-point problems arising from lagrangian multipliers*, Revue française d'automatique, informatique, recherche opérationelle. Analyse numérique, 8 (1974), pp. 129–151.

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

[27] F. Brezzi and J. Pitkäranta, *On the Stabilization of Finite Element approximations of the Stokes Equations*, vol. 10 of Notes on Numerical Fluid Mechanics, Vieweg-Teubner Verlag, 1984.

[28] Brezzi, F., Falk, Richard S., and Marini, Donatella L., *Basic principles of mixed Virtual element Methods*, ESAIM: M2AN, 48 (2014), pp. 1227–1240.

[29] 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.

[30] E. Cáceres and G. N. Gatica, *A mixed Virtual Element Method for the pseudostress-velocity formulation of the stokes problem*, Preprint available at ftp://ftp.ci2ma.udec.cl/pub/ci2ma/pre-publicaciones/2015/, (2015).

[31] E. Cáceres, G. N. Gatica, and F. A. Sequeira, *A mixed Virtual Element Method for quasi-newtonian stokes flow*, Preprint available at ftp://ftp.ci2ma.udec.cl/pub/ci2ma/pre-publicaciones/2015/, (2015).

[32] A. Cangiani, G. Manzini, A. Russo, and N. Sukumar, *Hourglass stabilization and the Virtual Element Method*, International Journal for Numerical Methods in Engineering, 102 (2015), pp. 404–436. nme.4854.

[33] A. Cangiani, G. Manzini, and O. J. Sutton, *Conforming and nonconforming Virtual Element Methods for elliptic problems*, arXiv:1507.03543, (2015).

[34] B. Decroux and O. Gosselin, *Computation of effective dynamic properties of naturally fractured reservoirs: Comparison and validation of methods*, in EAGE Annual Conference & Exhibition incorporating SPE Europec, Society of Petroleum Engineers, 2013.

[35] 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.

[36] 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.

[37] J. Erhel, J.-R. De Dreuzy, and B. Poirriez, *Flow simulation in three-dimensional Discrete Fracture Networks*, SIAM Journal on Scientific Computing, 31 (2009), pp. 2688–2705.

[38] 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.

[39] L. Formaggia, P. F. Antonietti, P. Panfili, A. Scotti, L. Turconi, M. Verani, and A. Cominelli, *Optimal techniques to simulate flow in fractured reservoir*, in ECMOR XIV-14th European conference on the mathematics of oil recovery, 2014.

[40] L. P. Franca, G. Hauke, and A. Masud, *Revisiting stabilized Finite Element Methods for the advective–diffusive equation*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1560 – 1572.

[41] 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.

[42] 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.

[43] R. Glowinski and M. F. Wheeler, *Domain decomposition and mixed Finite Element Methods for elliptic problems*, in First international symposium on domain decomposition methods for partial differential equations, 1998, pp. 144–172.

[44] Z. Huang, X. Yan, and J. Yao, *A two-phase flow simulation of discrete-fractured media using Mimetic Finite Difference Method*, Communications in Computational Physics, 16 (2014).

[45] J. D. Hyman, C. W. Gable, S. L. Painter, and N. Makedonska, *Conforming Delaunay triangulation of stochastically generated three dimensional Discrete Fracture Networks: A feature rejection algorithm for meshing strategy*, SIAM Journal on Scientific Computing, 36 (2014), pp. A1871–A1894.

[46] V. John and E. Schmeyer, *On Finite Element Methods for 3d time-dependent convection-diffusion-reaction equations with small diffusion*, in BAIL 2008 - Boundary and Interior Layers, A. Hegarty, N. Kopteva, E. O'Riordan, and M. Stynes, eds., vol. 69 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, 2009, pp. 173–181.

[47] 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.

[48] M. Karimi-Fard and L. J. Durlofsky, *Unstructured adaptive mesh refinement for flow in heterogeneous porous media*, in ECMOR XIV-14th European conference on the mathematics of oil recovery, 2014.

[49] A. Klawonn, *Feti Domain Decomposition Methods for second order elliptic partial differential equations*, GAMM-Mitteilungen, 29 (2006), pp. 319–341.

[50] A. Klawonn and O. Widlund, *FETI and Neumann-Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.

[51] J. C. Latché and D. Vola, *Analysis of the Brezzi-Pitkäranta stabilized Galerkin scheme for creeping flows of Bingham fluids*, SIAM Journal on Numerical Analysis, 42 (2004), pp. 1208–1225.

[52] V. Lenti and C. Fidelibus, *A BEM solution of steady-state flow problems in Discrete Fracture Networks with minimization of core storage*, Computers & geosciences, 29 (2003), pp. 1183–1190.

[53] M. Li, D. Shi, and Y. Dai, *The Brezzi–Pitkäranta stabilization scheme for the elasticity problem*, Journal of Computational and Applied Mathematics, 286 (2015), pp. 7 – 16.

[54] S. Li, Z. Xu, G. Ma, and W. Yang, *An adaptive mesh refinement method for a medium with Discrete Fracture Network: The enriched Persson's method*, Finite Elements in Analysis and Design, 86 (2014), pp. 41 – 50.

[55] K. Lipnikov, G. Manzini, and M. Shashkov, *Mimetic Finite Difference Method*, Journal of Computational Physics, 257 (2014), pp. 1163–1227.

[56] G. Manzini, A. Cangiani, and O. Sutton, *The conforming Virtual Element Method for the convection-diffusion-reaction equation with variable coefficients.*, tech. report, Los Alamos National Laboratory (LANL), Oct 2014.

[57] V. Martin, J. Jaffré, and J. E. Roberts, *Modeling fractures and barriers as interfaces for flow in porous media*, SIAM Journal on Scientific Computing, 26 (2005), pp. 1667–1691.

[58] J. Maryška, O. Severỳn, and M. Vohralík, *Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model*, Computational Geosciences, 8 (2005), pp. 217–234.

[59] H. Mustapha and R. Dimitrakopoulos, *High-order stochastic simulation of complex spatially distributed natural phenomena*, Mathematical Geosciences, 42 (2010), pp. 457–485.

[60] 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.

[61] J. Nocedal and S. J. Wright, *Numerical optimization*, Springer, Berlin, 1999.

[62] B. Nœtinger, *A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks accounting for matrix to fracture flow*, Journal of Computational Physics, 283 (2015), pp. 205–223.

[63] B. Nötinger and N. Jarrige, *A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks*, Journal of Computational Physics, 231 (2012), pp. 23 – 38.

[64] 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.

[65] P. Panfili and A. Cominelli, *Simulation of miscible gas injection in a fractured carbonate reservoir using an embedded discrete fracture model*, in Proceedings of Abu Dhabi International Petroleum Exhibition and Conference, Society of Petroleum Engineers, 2014.

[66] 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.

[67] ——, *A generalized mixed hybrid Mortar Method for solving flow in stochastic Discrete Fracture Networks*, SIAM Journal on scientific computing, 34 (2012), pp. B86–B105.

[68] P. Raviart and J. Thomas, *Primal hybrid Finite Element Methods for 2nd order elliptic equations*, Mathematics of computation, 31 (1977), pp. 391–413.

[69] C. Talischi, G. Paulino, A. Pereira, and I. Menezes, *Polymesher: a general-purpose mesh generator for polygonal elements written in Matlab*, Structural and Multidisciplinary Optimization, 45 (2012).

[70] G. Vacca and L. Beirão da Veiga, *Virtual Element Methods for parabolic problems on polygonal meshes*, Numerical Methods for Partial Differential Equations, 31 (2015), pp. 2110–2134.

[71] 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.

[72] 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.

[73] B. Wohlmuth, *Discretization methods And uterative solvers based on domain decomposition*, Springer, 2001.

[74] O. C. Zienkiewicz, S. Qu, R. L. Taylor, and S. Nakazawa, *The patch test for mixed formulations*, International Journal for Numerical Methods in Engineering, 23 (1986), pp. 1873–1883.

[75] ——, *Mixed and hybrid Finite Elements Methods*, Springer series in computational mathematics, Springer-Verlag, 1991.