A globally conforming method for solving flow in discrete fracture networks using the Virtual Element Method

Original

Availability:
This version is available at: 11583/2602373 since: 2016-03-18T13:02:42Z

Publisher:
Elsevier

Published
DOI:10.1016/j.finel.2015.10.003

Terms of use:
This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)
A globally conforming method for solving flow in discrete fracture networks using the Virtual Element Method

Matías Fernando Benedetto*, Stefano Berrone*, Stefano Scialô

*Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy.

Abstract

A new approach for numerically solving flow in Discrete Fracture Networks (DFN) is developed in this work by means of the Virtual Element Method (VEM). Taking advantage of the features of the VEM, we obtain global conformity of all fracture meshes while preserving a fracture-independent meshing process. This new approach is based on a generalization of globally conforming Finite Elements for polygonal meshes that avoids complications arising from the meshing process. The approach is robust enough to treat many DFNs with a large number of fractures with arbitrary positions and orientations, as shown by the simulations. Higher order Virtual Element spaces are also included in the implementation with the corresponding convergence results and accuracy aspects.

Keywords: VEM, Fracture flows, Darcy flows, Discrete Fracture Networks

1. Introduction

The present work deals with a new approach based on the Virtual Element Method (VEM) for the simulation of the flow in Discrete Fracture Networks (DFNs). DFN models are one of the possible approaches for simulating subsurface flows and they consist of a set of planar polygons in 3D space resembling
the fractures in the underground. Each fracture is modelled individually, as opposed to continuum models with equivalent porosity, and, for geological formations with a sparse fracture network that mainly affects the flow path, this approach is recommended [1, 2]. DFNs are used in a wide range of applications such as pollutant percolation, gas recovery, aquifers, reservoir analysis, among others [3] [4].

Stationary flow in a DFN is modelled using Darcy’s law and introducing a transmissivity tensor for each fracture that depends on its aperture and its resistance to flow. The surrounding rock matrix is considered impervious. The goal is to obtain the hydraulic head distribution in the system, which is the sum of the pressure head and the elevation. Fluid can only flow through fractures and across intersections between fractures, also called traces, but no tangential flow is considered along traces. The hydraulic head is a continuous function, but with discontinuous derivatives across the traces, which act as sources/sinks of flow. More complex models for the flow in the fractures can be found in the literature [5]. Since little is known about the subsurface fractures, stochastic models are used in order to determine distributions of aperture, hydrological properties, size, orientation, density, and aspect ratio of the fractures.

Geometrical complexity is the greatest challenge when dealing with DFN-based simulations. Since the fracture generation has a random component, many complex situations arise that render the meshing process very complicated and sometimes impossible, e.g. very small angles, very close and almost parallel traces, high disparity of traces lengths, etc. In order to use traditional finite elements, fracture grids have to match in all the intersections between fractures, since these are discontinuity interfaces for the first order derivatives of the solution. All the aforementioned geometrical configurations complicate the meshing process and are the biggest obstacle in the discretization of the problem because it becomes very computationally demanding to obtain a good mesh from such a badly predisposed geometry. Furthermore, the meshing procedure depends on the whole DFN and is not independent for each fracture. When a large DFN is
considered that can have thousands of fractures, mesh conformity requirements
can lead to a very high number of elements that are far more than those de-
manded by the required level of accuracy. In [6], a BEM (Boundary Element
Method) was applied that aims to minimize core memory usage by defining and
storing only a relation between nodal fluxes and hydraulic head on traces for
each fracture. The problem of obtaining a good globally conforming mesh is
the subject of ongoing research. In [7], an adaptive mesh refinement method is
described that aims for a high resolving mesh. Previous works [8, 9] suggest a
simplification of the geometry to ease meshing. Monodimensional pipes joining
fractures, instead of traces, have been put forward as an alternative in [10] and
[11]. In [12], a mixed formulation and a mesh modifying procedure was used
to solve DFNs and reducing the number of elements for each fracture. Another
mixed formulation was used in [13], where local corrections of traces are applied
in order to obtain a globally conforming mesh. The mortar method was used to
impose conditions between fractures with non-matching grids to obtain a mixed
hybrid formulation in [14], with a subsequent generalization in [15] that includes
trace intersections within a fracture. A novel approach was proposed in [16],
[17], [18] and [19] in which the problem was reformulated as a PDE-constrained
optimization. The minimization of a properly defined functional is adopted to
enforce hydraulic head continuity and flux conservation at fracture intersections.
Traditional finite elements (FEM) as well as extended finite elements (XFEM)
were implemented to solve the problem.

In this work, we aim to provide an easy, natural way for generating conform-
ing meshes for complex DFN problems using the VEM. The proposed approach
is a generalization of traditional conforming finite elements, keeping the method
as simple and streamlined as possible. Some of the ideas presented here where
present in a previous work by the authors [20], that introduced Virtual Ele-
ments (VEM) to DFNs. In [20] the VEM is used on locally conforming meshes
and an optimization approach is adopted to handle the non-conformity of the
global mesh. Here both local and global conformity is enforced, and classical

3
approaches, borrowed from the domain decomposition methods, can be used to solve the problem. We make absolutely no assumptions on the meshing procedure, which is done independently for each fracture and without any consideration of the position of the traces. Traces are not modified in any way, and using some of the features of the VEM, local and global conformity for the mesh is obtained by means of splitting the original elements of the meshes independently generated on each fracture into polygons of an arbitrary number of vertices.

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

Section 2 provides the formulation of the DFN problem in the present context, whereas a brief summary of the VEM is reported in Section 3, and in Section 4 the proposed method is described in detail. Numerical results are presented in Section 5, where some convergence results are given and the applicability of the method to DFNs is discussed.

2. The continuous problem

Let us consider a set of open convex planar polygonal fractures \( F_i \subset \mathbb{R}^3 \) with \( i = 1, \ldots, N \), with boundary \( \partial F_i \). Our DFN is \( \Omega = \bigcup_i F_i \), with boundary \( \partial \Omega \).

Even though the fractures are planar, their orientations in space are arbitrary, such that \( \Omega \) is a 3D set. The set \( \Gamma_D \subset \partial \Omega \) is where Dirichlet boundary conditions are imposed, and we assume \( \Gamma_D \neq \emptyset \), whereas \( \Gamma_N = \partial \Omega \setminus \Gamma_D \), is the portion of the boundary with Neumann boundary conditions. Dirichlet and Neumann boundary conditions are prescribed by the functions \( h^D \in H^{\frac{1}{2}}(\Gamma_D) \) and \( g^N \in H^{-\frac{1}{2}}(\Gamma_N) \) on the Dirichlet and Neumann part of the boundary, respectively.

We further set \( \Gamma_{i,i} = \Gamma_D \cap \partial F_i \), \( \Gamma_{i,N} = \Gamma_N \cap \partial F_i \), and \( h_{i,D} = h^D \mid \Gamma_{i,D} \) and \( g_{i,N} = g^N \mid \Gamma_{i,N} \). The set \( \mathcal{T} \) collects all the traces, i.e. the intersections between fractures, and each trace \( T \in \mathcal{T} \) is given by the intersection of exactly two fractures, \( T = \bar{F}_i \cap \bar{F}_j \), such that there is a one to one relationship between a trace \( T \) and a couple of fracture indexes \( \{i, j\} = \mathcal{I}(T) \). We will also denote by
\( \mathcal{T}_i \) the set of traces belonging to fracture \( F_i \).

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

We define the following functional spaces:

\[
V_i = H^1_0(F_i) = \{ v \in H^1(F_i) : v|_{\Gamma_{iD}} = 0 \},
\]

\[
V^D_i = H^1_D(F_i) = \{ v \in H^1(F_i) : v|_{\Gamma_{iD}} = h^D_i \},
\]

and

\[
V = \{ v : v|_{F_i} \in V_i, \forall i = 1, \ldots, N, \gamma_T(v|_{F_i}) = \gamma_T(v|_{F_j}), \forall T \in \mathcal{T}_i, \{i,j\} = I(T) \},
\]

where \( \gamma_T \) is the trace operator onto \( T \). It is then possible to formulate the DFN problem, given by the Darcy’s law in its weak form on the fractures with additional constraints of continuity of the hydraulic head across the traces: for \( i = 1, \ldots, N \), find \( H_i \in V^D_i \) such that \( \forall v \in V \)

\[
\sum_{i=1}^N \int_{F_i} K_i \nabla H_i \nabla v|_{F_i} \, dF_i = \sum_{i=1}^N \left( \int_{F_i} f_i v|_{F_i} \, dF_i + \langle g^N_i, v|_{\Gamma_{Ni}} \rangle_{H^{-\frac{1}{2}}(\Gamma_{Ni}), H^{\frac{1}{2}}(\Gamma_{Ni})} \right),
\]

\[
\gamma_T(H_i) = \gamma_T(H_j), \forall T \in \mathcal{T}, \{i,j\} = I(T)
\]

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

\[
a_i(H_i, v|_{F_i}) = \int_{F_i} K_i \nabla H_i \nabla v|_{F_i} \, dF_i. \tag{2.1}
\]
3. The Virtual Element Method

This section provides a quick overview of the VEM, recalling the main features useful in the present context. We refer the reader to the original paper [21] for a proper introduction and to [22] for a guide on implementation. Further developments can be found in [23], [24], [25] and [26]. The VEM has also been applied to problems in elasticity [27], plate bending [28], the Stokes problem [29] and has sparked interest in other applications as well.

Borrowing ideas from the Mimetic Finite Difference method [30, 31], the VEM can be regarded as a generalization of regular finite elements to meshes made up by polygonal elements of any number of edges. The discrete functional space on each element has, in general, not only polynomial functions but also other functions that are only known at a certain set of degrees of freedom. Given a bilinear form to be approximated with the VEM, our goal is to build a discrete bilinear form that coincides with the exact one when at least one of the arguments is a polynomial. For the other cases, a rough approximation that scales in a desired way is enough to obtain the desired convergence qualities of the method.

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

$$V_{k,h}^E = \{ v_h \in H^1(E) : v_{h|\partial E} \in C^0(\partial E), \ v_{h|e} \in P_k(e) \ \forall e \subset \partial E, \ \Delta v_h \in P_{k-2}(E) \}$$

where $\partial E$ is the border of $E$, and $e$ an edge.

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

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

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

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

and $(x_c, y_c)$ and $h_E$ are the centroid and the diameter of the element $E$ respectively. Different choices for the second type of degree of freedom is possible instead of point values, e.g. edge moments. We have chosen point values on Gauss-Lobatto nodes on edges for numerical integration purposes. The selected set of degrees of freedom is unisolvent [21], and therefore, given an element $E$ with $n_v$ vertices, we have that the dimension of $V_{k,h}^E$ is $\#V_{k,h}^E = n_v k + \frac{k(k-1)}{2}$.

We finally choose a basis for $V_{k,h}^E$, made of functions $\phi_i$ with $i = 1, \ldots, \#V_{k,h}^E$, such that, calling $\text{dof}_j(v)$ for $j = 1, \ldots, \#V_{k,h}^E$ the $j$-th degree of freedom applied to $v$, we have $\text{dof}_j(\phi_i) = \delta_{ij}$, being $\delta_{ij}$ the Kronecker delta. The global virtual element space is:

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

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

Let us now consider the restriction of the bilinear form (2.1) to a mesh element $E$, $a^E_i(\cdot, \cdot)$. We aim at building a discrete bilinear form $a_{i,h}^E : V_{k,h}^E \times V_{k,h}^E \mapsto \mathbb{R}$ having the previously stated polynomial consistency, i.e. the discrete bilinear form has to coincide with the exact one when at least one of the arguments is a polynomial of maximum degree $k$. To this end let us consider the projector operator of order $k$ on $E$:

$$\Pi_{E,k}^\Sigma : V_{k,h}^E \mapsto \mathcal{P}_k(E)$$
such that
\[ \Pi_{E,k} q_k = q_k \text{ for all } q_k \in \mathcal{P}_k(E), \]
defined by the equations
\[ \int_E \nabla q_k \cdot \nabla v_h = \int_E \nabla q_k \cdot \nabla \Pi_{E,k} v_h \text{ for all } q_k \in \mathcal{P}_k(E), \]
\[ \int_E \Pi_{E,k} v_h = \int_E v_h. \]
The projection \( \Pi_{E,k} v_h \) can be uniquely defined starting from the degrees of freedom of \( v_h \) using integration by parts [22] and represents an orthogonality condition in the \( H^1 \) inner product. The first equation defines the projection up to a constant, which is defined by the second equation. Other options for the second equation exist [26]. For order \( k = 1 \), it can be taken as
\[ \frac{1}{N^r} \sum_{i=1}^{N^r} \Pi_{E,k} v_h(\mathcal{V}_i) = \frac{1}{N^r} \sum_{i=1}^{N^r} v_h(\mathcal{V}_i) \]
where \( \mathcal{V}_i \) are the vertices of the element and \( N^r \) its number.

Remark 1. In the case of a more complex equation than the Laplacian (or even the Laplacian with non-constant coefficients), other projectors have to be considered [26].

Let us now take any symmetric, positive definite bilinear form \( S_{i,h}^E : V_{k,h}^E \times V_{k,h}^E \to \mathbb{R} \), such that there exist \( c_0 \) and \( c_1 \) positive constants, independent of the element \( E \) and its diameter, that verify
\[ c_0 a_i^E(v_h, v_h) \leq S_{i,h}^E(v_h, v_h) \leq c_1 a_i^E(v_h, v_h) \quad \forall v_h \in V_{k,h}^E \text{ with } \Pi_{E,k} v_h = 0. \]
This implies that \( S_{i,h}^E \) scales like \( a_i^E(v_h, v_h) \), and then the local discrete bilinear form \( a_{i,h}^E \) is set as
\[ a_{i,h}^E(u_h, v_h) = a_i^E(\Pi_{E,k} u_h, \Pi_{E,k} v_h) + S_{i,h}^E(u_h - \Pi_{E,k} u_h, v_h - \Pi_{E,k} v_h) \forall u_h, v_h \in V_{k,h}^E. \]
The first terms ensures the \textit{consistency} and the second one the \textit{stability} of the form. Finally, the complete discrete bilinear form becomes

\[ a_{i,h}(u_h, v_h) = \sum_{E \in \tau_h} a_{i,h}^E(u_h, v_h) \quad \forall u_h, v_h \in V_{k,h}. \]

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

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

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

For the right hand side with load term $f$, it is enough for optimal convergence \cite{22} to consider

\[ (f, v_h) = \sum_{E \in \tau_h} \int_E f \Pi_{E,k-1}^0 v_h \quad \text{for order } k = 1, 2, \]
\[ (f, v_h) = \sum_{E \in \tau_h} \int_E f \Pi_{E,k-2}^0 v_h \quad \text{for order } k \geq 3, \]

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

4. Problem implementation

4.1. Mesh generation

Mesh generation is done independently for each fracture regardless of traces and their positions. The process of mesh generation consists of three steps: the first task is the generation of a baseline triangulation of each fracture, not necessarily conforming to trace disposition, and independent on each fracture; the second step is the generation of a fracture-local conforming mesh, splitting
the triangles of the baseline mesh into polygons conforming to the traces; finally
on each fracture $F_i$, nodes are added on the traces $T \in \mathcal{T}_i$ corresponding to the
nodes of the intersecting fracture $F_j$ with $\{i, j\} = \mathcal{I}(T)$, $\forall T \in \mathcal{T}_i$, thus gaining
global conformity. The three steps are depicted in Figure 4.1, and, the second
and third steps are further described in full details in the next paragraphs.

4.1.1. Local conformity

Local conformity is obtained as in the previous work [20]. Every time a
trace intersects an edge of the triangulation, a new node is created there. Nodes
are also created at trace tips. If a trace tip is inside a triangular element, we
extend the geometrical segment coinciding with the trace up to the nearest edge
of the triangulation, thereby creating a new edge and a new node. The trace is
not modified, being now a subset of the extended segment. By doing this, we
split the original elements of the triangulation into new convex “sub-elements”,
which are elements of the mesh in their own right. The end result is a mesh
of polygonal elements for which all traces are covered by element edges, see
Figures 4.1a and 4.1b, where element colouring indicates the number of edges.
A careful inspection of those subfigures reveals all of the situations described
above.

Remark 2. An optional mesh modification has been implemented that rear-
ranges some of the nodes of the baseline triangulation before the splitting process,
so as to make them coincide with nearby traces, trace tips and trace intersec-
tions. This leads to better shaped elements and fewer DOFs for the final mesh
and it is not computationally demanding.

4.1.2. Global conformity

After obtaining the locally conforming mesh the subsequent step is to ensure
that all the nodes on the traces are included in the meshes of both fractures that
share the trace. These nodes are the ones shared by more than one fracture.
This is the most important feature of the method we are proposing and takes
full advantage of VEM versatility. Given a trace $T$ shared by fractures $F_i$ and
$F_j$, we define $U^F_T$ as the set of all nodes on the trace $T$ in fracture $F_i$ and
analogously $U^F_T$ for $F_j$. The procedure used to obtain the global conforming
mesh guarantees that both trace tips are included and that the discretization includes all nodes on the traces and covers it precisely. The complete trace discretization is then $U_T = U_T^F_i \cup U_T^F_j$. What remains now is to simply add the set of nodes $U_T \setminus U_T^F_i$ on the corresponding elements of fracture $F_i$ and analogously for fracture $F_j$. This can be done since the VEM allows for elements of arbitrary number of edges and $180^\circ$ angles between them. The final globally conforming mesh is shown in Figure 4.1c and is identical to the previous mesh except for the new added nodes on the traces and a change in element colouring that is an indication of the increment in the number of edges and DOFs.

Figure 4.1: Original mesh, VEM mesh and final globally conforming mesh
4.2. Imposing matching conditions

For every fracture $F_i$, with $i = 1, ..., N$, we call $n_{dof_i}$ the number of DOFs of fracture $F_i$ and we assemble the stiffness matrix $K_i \in \mathbb{R}^{n_{dof_i} \times n_{dof_i}}$ following the procedure described in Section 3. Then we construct the column vectors $f_i \in \mathbb{R}^{n_{dof_i}}$ as the vector of load values (including terms arising from non-homogeneous boundary conditions) and $h_i$ as the vector of nodal values of the discrete solution. We note that the matrix $K_i$ is singular for fractures with pure Neumann boundary conditions. For the complete DFN we have:

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

In order to obtain the saddle point linear system for the complete DFN we have to impose matching conditions for the nodes on the traces that guarantee the continuity condition of the hydraulic head. We do that by means of Lagrange multipliers $\lambda_t$, for $t = 1, ..., n_{dof_t}$. They are introduced for each node on the traces in a non-redundant way (see [32]) which means that in the case of two intersecting traces, i.e. three fractures sharing a single point in space (as in the example of Section 5.1.2), only two multipliers are added. To each index $t = 1, ..., n_{dof_t}$ corresponds a node on a trace $T$ that is shared by fractures $F_i$ and $F_j$, and we denote by $dof_i(t)$ the corresponding global DOF for node $t$ on $F_i$ and analogously by $dof_j(t)$ the DOF on $F_j$. We define $N^h = \sum_{i=1}^{N} n_{dof_i}$, and the row vector $L_t \in \mathbb{R}^{N^h}$ as:

$$L_t = \begin{pmatrix} 0 & \cdots & 0 & 1 & 0 & \cdots & 0 & -1 & 0 & \cdots & 0 \end{pmatrix}.$$  

Finally, we set $L \in \mathbb{R}^{n_{dof_t} \times N^h}$ as the matrix:
The final linear system is:

\[
\begin{bmatrix}
K & L^T \\
L & 0
\end{bmatrix}
\begin{bmatrix}
h \\
\lambda
\end{bmatrix} =
\begin{bmatrix}
f \\
0
\end{bmatrix},
\]

(4.2)

This saddle point problem has a unique solution as it can be easily proven resorting to classical results of quadratic programming [33].

When the dimensions of the system 4.2 are large, the use of an iterative method and of a preconditioner is advised. We briefly recall the one-level FETI method for domain decomposition as described in [34] here implemented. In this method the primal variables are determined in terms of the Lagrange multipliers. More precisely, we define a block diagonal matrix \( R \) as

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

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

\[
h = K^*(f - L^T \lambda) + R \alpha
\]
where $K^*$ is the pseudoinverse of $K$ and the vector $\alpha$ depends on $\lambda$ but not on the primal variables $h$. This means that if we solve a system for $\lambda$, this completely determines the solution. In order to solve this system for $\lambda$, a choice of several preconditioners is possible.

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

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

The local Schur complement $S_i$ is defined as:

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

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

$$M^{-1} = L_D S L_D^T.$$ 

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

\[ M^{-1} = L_D K^{(\tau \tau)} L_D^T, \]

where \( K^{(\tau \tau)} \) is the block diagonal matrix made up by the local \( K_i^{(\tau \tau)} \). We note that in order to define inner products for the Preconditioned Conjugate Gradient (PCG) FETI algorithm, a symmetric, positive definite matrix \( Q \) is used [34]. In our experiments we have considered \( Q = M^{-1} \).

5. Numerical results

In this section we present some numerical results, beginning with convergence results for benchmark problems and VEM spaces of various orders. We also compare the results obtained with this approach to the results of a validated XFEM based method on a medium size DFN [18, 19]. We conclude showing some examples of numerical instabilities arising mainly with the higher order VEM approximation spaces for certain particularly adverse geometrical configurations. All of the results were obtained using a constant transmissivity tensor \( K = 1 \) for all fractures.

5.1. Convergence results

The error norms used for the convergence curves are the usual \( L^2 \) and \( H^1 \) norms. The error is computed by taking the projection of the discrete solution on the space of polynomials, since the values of the discrete solution are only known at the DOFs and are not explicitly known inside the elements (see [26]):

\[ Err^2_{L^2} = \sum_{E \in T_5} ||H - \Pi^e_{E,k} h_E||_{L^2(E)}^2, \]

\[ Err^2_{H^1} = \sum_{E \in T_5} ||H - \Pi^\nabla_{E,k} h_E||_{H^1(E)}^2, \]

where \( \Pi^\nabla_{E,k} \) is the projection operator of order \( k \) as defined in Section 3, \( H \) is the exact solution and \( h_E \) is the discrete solution restricted to element \( E \).
The flux incoming in a fracture through the traces is computed as the jump of the conormal derivative of the discrete solution across the traces. For every trace we fix a tangential orientation and a normal unit vector obtained by clockwise rotating by $90^\circ$ the tangent vector of the trace in the fracture plane. For every mesh edge $e \subseteq T$, i.e. an edge included in trace $T$, we consider a unique normal vector $n_{e,i}$ in $F_i$ with an orientation given by the normal vector fixed for the trace, and we define the flux incoming in the fracture $F_i$ through the edge $e$, named $u_{e,i}$, as follows:

$$u_{\text{left},e,i} = \nabla \Pi_{E_{l,i}} h_{E_{l,i}} \cdot n_{e,i},$$

$$u_{\text{right},e,i} = -\nabla \Pi_{E_{r,i}} h_{E_{r,i}} \cdot n_{e,i},$$

$$u_{e,i} = u_{\text{left},e,i} + u_{\text{right},e,i},$$

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

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

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

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

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

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

5.1.1. Benchmark problem 1

This first problem has been considered before in the context of the XFEM (eXtended finite elements) [17] and of the VEM [20] as a single-fracture problem. Nevertheless, it remains interesting for the fact that it includes a trace tip inside the domain and the exact solution is known. In this work the problem is
Figure 5.1: Spatial distribution of fractures for benchmark problem 1

Figure 5.2: Convergence curves for benchmark problem 1 - Fracture 1
considered as a 2-fracture DFN, as shown in Figure 5.1 and the error calculations and convergence curves are shown for the first fracture, $F_1$.

Let us define the domains $F_1$ and $F_2$ as

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

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

with a single trace $T = \{(x, y) \in \mathbb{R}^2 : y = 0, z = 0 \text{ and } -1 \leq x \leq 0\}$ ending in the interior of $F_1$ (Figure 5.1).

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

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

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

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

\[- \Delta H = -\Delta H_1^{ex} \text{ on } F_1 \setminus T, \]
\[ H = 0 \text{ on } \partial F_1, \]

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

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

Convergence curves for the VEM of orders from 1 to 4 are shown in Figure 5.2. The expected rates of convergence are obtained for orders 1 and 2, whereas a slower rate of convergence for orders 3 and 4 was obtained as a consequence of the insufficient regularity of the exact solution in the sense of Sobolev spaces.

Numerical solutions for the hydraulic head \( H_1 \) with the VEM of orders 1 and 2 are shown in Figure 5.3 a) and b). In Figure 5.3 c) and d), we present a comparison between the exact solution and the approximate solution of the flux incoming in \( F_1 \), as well as its left and right components. Note how the approximation of the trace flux \( U \) is piecewise constant for order 1 VEM and piecewise linear for order 2 VEM, and the approximation of the exact flux (dashed line) with the VEM of second order is greatly improved.

5.1.2. Benchmark problem 2

This problem shows the performance of the proposed approach in presence of trace intersections. The considered system consists of 3 fractures and 3 traces.
Figure 5.4: Spatial distribution of fractures for benchmark problem 2

![Spatial Distribution of Fractures](image)

Figure 5.5: Convergence curves for benchmark problem 2 - Fracture 1

(a) Order 1  
(b) Order 2  
(c) Order 3  
(d) Order 4
Figure 5.6: Solutions for benchmark problem 2 - Fracture 1 and trace 1

as shown in Figure 5.4:

\[
F_1 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq x \leq 1, \ -1 \leq y \leq 1, \ z = 0\}, \\
F_2 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq y \leq 1, \ -1 \leq z \leq 1, \ x = 0\}, \\
F_3 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq z \leq 1, \ -1 \leq x \leq 1, \ y = 0\},
\]

\[
T_1 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq x \leq 1, \ y = 0, \ z = 0\}, \\
T_2 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq y \leq 1, \ z = 0, \ x = 0\}, \\
T_3 = \{(x, y, z) \in \mathbb{R}^3 : -1 \leq z \leq 1, \ x = 0, \ y = 0\}.
\]

Note that all of the three traces intersect in a single point \(P = (0, 0, 0)\) in space

(as it is always the case for the intersection of 3 planar fractures).

Exact solutions are known for all fractures:
\[ H_1^e(x, y) = |x|(1 + x)(1 - x)y(1 + y)(1 - y), \]
\[ H_2^e(y, z) = y(1 + y)(1 - y)|z|(1 + z)(1 - z), \]
\[ H_3^e(z, x) = z(1 + z)(1 - z)x(1 + x)(1 - x). \]

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

The problem is then:

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

Convergence curves for the VEM of orders from 1 to 4 are shown in Figure 5.5 and solutions for order 1 and 2 are reported in Figure 5.6. In contrast with benchmark problem 1, the expected convergence speed is achieved for all orders, since now the exact solution has \( C^\infty \) regularity on each of the subdomains defined by the traces and the mesh for the numerical solution is conforming to the traces. This is a sufficient condition for optimal convergence rates, [35, 36].

The error in the discrete solution for VEM of order 6 is \( ||H - h||_{L^2}^2 = 3.53e - 19, \)
\( ||\partial_x(H - h)||_{L^2}^2 = 5.09e - 18 \) and \( ||\partial_y(H - h)||_{L^2}^2 = 5.85e - 18 \), being then of the same order of the round-off error in double precision. This confirms that the discrete solution coincides numerically with the exact solution.

### 5.2. DFN - 27 fractures

Let us consider the DFN shown in Figure 5.7 consisting of 27 fractures. A sink fracture \( F_1 \) and a source fracture \( F_2 \) are defined, both having a non
Figure 5.7: DFN 27: Spatial distribution of fractures for a DFN with 27 fractures

Table 5.1: DFN 27: Net flux in source (So) and sink (Si) fractures and flux mismatch $\Delta$ for various mesh sizes and VEM orders

<table>
<thead>
<tr>
<th>Method</th>
<th>mesh 150</th>
<th>mesh 120</th>
<th>mesh 90</th>
<th>mesh 60</th>
<th>mesh 30</th>
<th>mesh 15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Si</td>
<td>So</td>
<td>$\Delta$</td>
<td>Si</td>
<td>So</td>
<td>$\Delta$</td>
</tr>
<tr>
<td>VEM-1</td>
<td>8.75</td>
<td>-8.22</td>
<td>0.53</td>
<td>8.70</td>
<td>-7.92</td>
<td>0.78</td>
</tr>
<tr>
<td>VEM-2</td>
<td>11.23</td>
<td>-9.78</td>
<td>1.45</td>
<td>11.16</td>
<td>-10.05</td>
<td>1.09</td>
</tr>
<tr>
<td>VEM-3</td>
<td>11.60</td>
<td>-10.36</td>
<td>1.23</td>
<td>11.64</td>
<td>-10.60</td>
<td>1.04</td>
</tr>
<tr>
<td>VEM-4</td>
<td>11.88</td>
<td>-10.76</td>
<td>1.12</td>
<td>11.89</td>
<td>-10.92</td>
<td>0.98</td>
</tr>
<tr>
<td>VEM-1</td>
<td>9.01</td>
<td>-7.75</td>
<td>1.26</td>
<td>9.73</td>
<td>-8.32</td>
<td>1.42</td>
</tr>
<tr>
<td>VEM-2</td>
<td>11.18</td>
<td>-10.03</td>
<td>1.08</td>
<td>11.40</td>
<td>-10.26</td>
<td>1.14</td>
</tr>
<tr>
<td>VEM-3</td>
<td>11.64</td>
<td>-10.73</td>
<td>0.91</td>
<td>11.80</td>
<td>-10.89</td>
<td>0.9</td>
</tr>
<tr>
<td>VEM-4</td>
<td>11.91</td>
<td>-10.99</td>
<td>0.92</td>
<td>12.03</td>
<td>-11.17</td>
<td>0.86</td>
</tr>
<tr>
<td>VEM-1</td>
<td>10.56</td>
<td>-8.51</td>
<td>2.05</td>
<td>10.71</td>
<td>-9.49</td>
<td>1.23</td>
</tr>
<tr>
<td>VEM-2</td>
<td>11.83</td>
<td>-10.77</td>
<td>1.06</td>
<td>11.91</td>
<td>-11.00</td>
<td>0.91</td>
</tr>
<tr>
<td>VEM-3</td>
<td>12.11</td>
<td>-11.25</td>
<td>0.86</td>
<td>12.13</td>
<td>-11.53</td>
<td>0.59</td>
</tr>
<tr>
<td>VEM-4</td>
<td>12.26</td>
<td>-11.48</td>
<td>0.78</td>
<td>12.21</td>
<td>-13.01</td>
<td>-2.81</td>
</tr>
<tr>
<td>VEM-1</td>
<td>10.98</td>
<td>-9.18</td>
<td>1.81</td>
<td>11.36</td>
<td>-10.26</td>
<td>1.12</td>
</tr>
<tr>
<td>VEM-2</td>
<td>12.00</td>
<td>-11.09</td>
<td>0.90</td>
<td>12.12</td>
<td>-11.65</td>
<td>0.47</td>
</tr>
</tbody>
</table>

23
homogeneous Dirichlet boundary conditions on one edge of their boundary and
homogeneous Neumann boundary conditions on the remaining edges. All other
fractures have homogeneous Neumann boundary conditions and are therefore
insulated on their boundaries. In absence of an exact solution, the difference $\Delta$
between the flux entering the system from $F_2$ (the source fracture), “$S_0$”, and
the flux leaving it from $F_1$ (sink fracture), “$S_i$”, is considered for assessing the
quality of the obtained numerical approximation.

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

Table 5.1 shows the net flux in the source and sink fractures, $S_i$ and $S_0$,
respectively, as well as the difference $\Delta$ for mesh parameters (area of the largest
element of the mesh) ranging from 5 to 150 and orders of the VEM space from
1 to 4. Only orders 1 and 2 are considered on the two finer meshes.

After extensive numerical experiments a trend emerged in the results; for or-
der 1, convergence can be quite slow in the flux variable on these coarse meshes
and displays oscillations, this can be attributed to the fact that the approxima-
tion of the flux is only piecewise constant and the projection of the VEM space
functions for each element is onto a polynomial space of degree one, regardless
of the number of edges of the element. Moving to higher order discretization
spaces, the approximation of the flux improves. A marked improvement is ob-
tained with second order VEM with respect to the first order, probably due to
the piecewise linear structure of $U$. Further increasing the VEM order has a less
noticeable effect, with practically no gain in moving to a third or fourth order
Table 5.2: Comparison of iterations for different choices of $Q$ and preconditioner $M^{-1}$

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

approximation. In addition, higher order discretizations might suffer from numerical instabilities due to very badly shaped elements. This is for example the case for the fourth order approximation on the mesh size 15, where instabilities cause a degenerate discrete solution as shown by the parameter $\Delta$ reported in Table 5.1. Further details on possible causes of instabilities are discussed later in Paragraph 5.4.

Remark 3. When tackling a new DFN, a good practice would be to run it the first time with a coarse mesh and first order elements. The values of $h$ and of $u$ already provide a reliable indication of the order of magnitude of the correct solution, and using the flux values on each fracture one can establish a rule for selecting the fractures for which a mesh refinement is advisable. Fractures with less important contribution to the total flux through the DFN do not require a finer mesh. Afterwards, a new simulation can be launched with second order elements and the new adapted mesh.

5.3. DFN - 120 fractures

We now consider a DFN consisting of 120 fractures, as shown in Figure 5.8. Dirichlet boundary conditions are imposed on a source and sink fracture whereas all other fractures have homogeneous Neumann boundary conditions. In Figure 5.9 we plot the solution for the sink fracture and for a selected fracture with insulated boundaries. As a comparison, results are shown for both the VEM approach of order 2 depicted in the present work and for the XFEM based optimization approach described in [18], starting from the same baseline mesh. A very good agreement between the solutions can be appreciated in the figure. Good agreement was also obtained for VEM of orders 1 and 3.
Figure 5.8: DFN 120: Spatial distribution of fractures for a DFN with 120 fractures

(a) $h$ for fracture 1 (VEM - order 2) (b) $h$ for fracture 1 (XFEM)

(c) $h$ for fracture 57 (VEM - order 2) (d) $h$ for fracture 57 (XFEM)

Figure 5.9: DFN 120: Large DFN comparison
In Table 5.2, we report the behaviour of 2 preconditioning techniques. Different mesh parameters and VEM of order 1 and 2 are considered. The table displays the number of iterations required by the conjugate gradient (CG) routine compared to the performances of the preconditioned algorithm with the Lumped and Dirichlet preconditioners. For the non preconditioned CG algorithm, a rapid increase in the iteration number with mesh refinement can be appreciated for both orders 1 and 2. As expected, the increase in iterations with a preconditioner is much smaller, with the Dirichlet preconditioner performing better than the Lumped preconditioner.

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

5.4. A survey of troublesome situations

In this subsection we describe some situations arisen in the simulations that have proven to be difficult to handle numerically. The monomial basis for the space of polynomials is notoriously bad conditioned, and the situation worsens with increasing orders. We believe that this is the cause of the issues we are presenting in this section, and they appear in elements with unsuitable shapes. Some of these issues can be prevented if a mesh modifying procedure as mentioned in Remark 2 is used.

A first example is related to the DFN with 120 fractures, where a fracture has two traces that are almost parallel and very close to each other, as in Figure 5.10. This inevitably leads to elements with a bad aspect ratio, since any attempt to obtain an adequate mesh would require a very large number of small elements to fill the space between the two traces. The solution is stable up to VEM of order 3, while when using a fourth order approximation the obtained solution drastically changes (see Figure 5.11), and even falls below zero, which is not compatible with the imposed boundary conditions, necessarily leading
Figure 5.10: DFN 120: Detail of two very close and almost parallel traces

Figure 5.11: DFN 120: Comparison of results for problematic situations
Figure 5.12: DFN 27: Detail of an unfortunate disposition of a mesh edge and a trace

Figure 5.13: DFN 27: Comparison of results for problematic situations

to a solution bounded between 0 and 100. As a reference, one particularly problematic mesh element has an almost rectangular shape and an area of 0.58, with a length of 10.26 in one direction and 0.058 in the other (a 177 ratio). This is a degenerate octagon and for order 4 it has 38 DOFs (Figure 5.10). We remark that this particular configuration can be successfully dealt with VEM of orders from 1 to 3, and problems only appear with order 4 and higher.

A second documented problematic configuration, occurred on the DFN 27 problem, concerns badly shaped elements due not to the geometry of the DFN but to an unfortunate starting mesh, and is such that it may not be present with either a finer or a coarser mesh. This situation could be prevented applying the mesh smoothing process described in Remark 2. The situation is depicted in
Figure 5.14: DFN 130: Spatial distribution of fractures for a DFN with 130 fractures

Figure 5.12, where we can see that the edge of an element is very close to a trace and has originated elements much more stretched in one direction than in the other. Furthermore, a very small element was generated next to the stretched element. The solution for VEM of order 5 becomes numerically unstable in this case, as shown by Figure 5.13. We remark that the major source of instability in this case is again the elongated element and not the neighboring small element.

Finally, we present the last case that is part of a medium size DFN with 130 fractures, shown in Figure 5.14, that includes parallel traces very close to each other, large disparity between trace lengths, highly heterogeneous element areas, element angles of less than 1 degree and complex trace intersections among other complications. More precisely, we have for the whole DFN that: minimum angle $= 0.41^\circ$, maximum trace length $\approx 45$, minimum trace length $\approx 0.01$ and largest number of traces in a fracture $= 24$. An adequate globally conforming triangular mesh for this system would be quite difficult to obtain, if not impossible. With our approach, meshing can be done as usual (Figure 5.15) although it may lead to elements with undesirable shapes. It can be seen that irregularities in the
Figure 5.15: DFN 130: Detail of two traces meeting at a very small angle

Figure 5.16: DFN 130: Comparison of results for problematic situations
solution were present only starting from VEM of order 4 approximations, again
at a very elongated element between two traces meeting at a very small angle
(Figure 5.16). The solution shows an uneven and rough behaviour that is further
propagated to other fractures that have traces in common, and was not present
in the solution obtained with the VEM of order 3.

6. Conclusions

In this work we have presented a novel method that constitutes a natural
generalization of conforming Finite Elements for Discrete Fracture Network flow
simulations. Local and global conformity is obtained using some of the features
of the Virtual Element Method, and most importantly, global conformity is
achieved without any constraints in the initial meshing process, that is per-
formed independently for each fracture, nor any modification of DFN geometry.
Convergence curves were presented as well as results for DFNs of small and
medium scale, and the method has been shown to be robust enough to handle
complex geometrical situations that arise in randomly generated DFNs.

After extensive numerical experiments, the following patterns were noticed:
in general, all methods give a good approximation for the hydraulic head $H$, and due to how the problem was implemented, continuity of $H$ for the whole
DFN is guaranteed. Even with VEM of order 1 the solutions are reliable for this
variable, and this is due to the fact that we are using the primal formulation
of the problem and the local conformity of the mesh allows for a more accurate
representation of the jump of the derivative of $H$ along the traces. In the case of
the flux exchanged at the traces, $U$, the situation is different; only starting with
a somewhat fine mesh can acceptable results be obtained for order 1. Order 2
on the other hand, shows a marked improvement that can be attributed to the
larger number of DOF but also to the improved approximation of the gradient
of $H$ and consequently of $U$. We remark that $U$ is not obtained directly, but
deriving the projection onto a polynomial space of the computed primal variable
$H$. 32
Concerning the use of discretizations with increasing polynomial accuracy, for this application, we discourage going beyond order 2 based on the obtained results. Higher orders are not only less stable numerically on strongly distorted meshes, but also much more computationally expensive, and the improvement in accuracy is often not considerable. In fact, the exact solution of a DFN does not have in general high regularity and a cubic approximation of $H$ and a quadratic approximation for $U$ might be excessive. As we have seen however, whenever regularity is guaranteed, convergence for higher orders is as good as expected.

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

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

**Acknowledgements**

This research has been partially supported by the Italian MIUR through PRIN research grant 2012HBLYE4_001 *Metodologie innovative nella modellistica differenziale numerica* and by INdAM-GNCS through project *Tecniche numeriche per la simulazione di flussi in reti di fratture di grandi dimensioni*. The first author was also supported by the European Commission through the Erasmus Mundus Action 2-Strand1 ARCOIRIS programme, Politecnico di Torino.
References


