UNIVERSIDAD DE BUENOS AIRES

Facultad de Ciencias Exactas y Naturales

Departamento de Matemática

Tesis de Licenciatura

LA FORMULACIÓN MIXTA DEL MÉTODO DE ELEMENTOS VIRTUALES PARA PROBLEMAS TRIDIMENSIONALES CON PLANOS EMBEBIDOS

Matías Fernando Benedetto

Director: Ricardo Durán

17/12/2018

Matías Fernando Benedetto

UNIVERSIDAD DE BUENOS AIRES

Facultad de ciencias exactas y naturales

Thesis presented for the degree in Mathematics

THE MIXED VIRTUAL ELEMENT FORMULATION FOR THREE DIMENSIONAL PROBLEMS WITH EMBEDDED PLANES

Matías Fernando Benedetto

Advisor
Prof.Ricardo Durán

December 28, 2018

UNIVERSIDAD DE BUENOS AIRES

Abstract/Resumen

DM - Departamento de Matemática

Degree in Mathematics

THE MIXED VIRTUAL ELEMENT FORMULATION FOR THREE DIMENSIONAL PROBLEMS WITH EMBEDDED PLANES

by Matías Fernando Benedetto

In this thesis, the methodology for solving second order elliptic equations by means of the Mixed Virtual Element Method is presented, for 2D, 3D and hybrid problems. The main features of the method, its basic theoretical results and details on its implementation are provided. Several numerical results in various contexts are given showing the performance of the method for pure 3D problems, Discrete Fracture Networks (DFN) and the combination of both situations.

En esta tesis se presenta una metodología para la resolución de problemas elípticos de segundo orden mediante el Método de Elemento Virtuales, en problemas 2D, 3D e híbridos. Se detallan las principales características del método, sus resultados teóricos básicos y su implementación. Se concluye con variados resultados numéricos en distintos contextos mostrando el desempeño del método para problemas 3D puros, Sistemas Discretos de Fracturas (DFN) y una combinación de ambas.

Acknowledgements/Reconocimientos

Firstly I would like to thank Professor Ricardo Durán for his predisposition and assistance with the development of the thesis, as well as for his lectures in the course about the Finite Element Method offered by the Department of Mathematics of the University, whose content has been invaluable for the comprehension of the topics presented here. Additionally, a great deal of this work was possible from the building point of the work developed with the GEOSCORE group of the Politecnico di Torino, regarding the numerical study of Discrete Fracture Networks (DFN) by means of the Virtual Element Method (VEM). A mention goes also to F. Kyburg and J. Mollica, engineers from the Instituto Tecnológico de Buenos Aires (ITBA) for the contributions to the coding of three-dimensional meshes. Finally, I would like to highlight the University of Buenos Aires, and the Faculty of Exact and Natural sciences in particular, for the excellent education provided to the students of this demanding but ultimately very rewarding career.

En primer lugar me gustaría agradecer al profesor Ricardo Durán por su predisposición y ayuda en el desarrollo de la tesis, y además por el curso de Elementos Finitos dictado por el departamento de Matemática de la Facultad, contenido imprescindible para la comprensión de los temas que se presentan aquí. Por otro lado, gran parte del trabajo fue posible como continuación del trabajo desarrollado en el Politécnico di Torino junto al grupo GEOSCORE concerniente al estudio numérico de Sistemas Discretos de Fracturas mediante el Método de Elementos Virtuales y cuyos avances constituyeron la plataforma a partir de la cual se construye esta tesis. Un agradecimiento también va para los Ingenieros F. Kyburg y J. Mollica del Instituto Tecnológico de Buenos Aires (ITBA) por sus contribuciones con respecto a la programación de mallas tridimensionales. Finalmente quiero destacar a la Universidad de Buenos Aires y en especial a la Facultad de Ciencias Exactas y Naturales por la excelente formación brindada a los alumnos de esta demandante pero al final muy recompensante carrera.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Numerical methods for underground flow simulations attract great interest in the scientific community. In geophysical sciences, taking advantage from an increased and easily available computational power, several problems not considered in the past have been tackled. There are many practical contexts where effective flow simulations in underground fractured media are strategic. These include geothermal applications, protection of water resources, Oil&Gas enhanced production and geological waste storage. Regardless of the application, they all share the demand for high accuracy and reliability in the results. Unfortunately, due to the high uncertainty in assessing underground data, a large number of simulations is needed to provide a statistically robust estimation of the required quantities.

In this work, the context considered is the simulation of the hydraulic head distribution in the subsoil. The physical components of the problem are the rock matrix and its fractures. One approach present in the scientific literature is by using "Discrete Fracture Networks" (DFN) [35, 39, 50, 54]. In such a model, and in contrast to a "continuous approach", the fractures are considered as discrete 2D domains with a much higher permeability than the surrounding matrix. These plane fractures are generated randomly following geotechnical parameters, and intersect each other in 1D segments called "traces".

In some cases the rock matrix is considered impervious due to its much lower permeability, and hence it is left out of the analysis, since subsurface flow will be mostly determined by the fracture distribution. These pure DFN problems are characterized by huge geometrical complexities forming an intricate network of intersections. The main challenge lies in obtaining the mesh, which usually requires a very considerable computational effort for any slightly complex DFN. Standard numerical

methods demand for a 'conforming mesh', which means that meshes on different fractures should be compatible with each other, thus imposing severe constraints on the meshing process that can't be done independently for each fracture and usually require an iterative process.

Over the last decade, there has been a great development of numerical methods to tackle the problem of efficient flow simulations of realistic DFNs. The complexity of DFN flow simulations is reduced in [51, 52] by removing the unknowns in the interior of the fractures, reducing the dimension of the problem and rewriting it at the interfaces. By using eXtended Finite Element (XFEM), mesh conformity between fractures is no longer required ([40]). In [18, 20–23], an optimization approach is proposed which also avoids the need for conformity and instead the problem is solved by minimizing a functional. In recent times, techniques as the Mimetic Finite Difference method (MFD) [11, 48] have been used for flow simulations in DFNs by [3, 5], as well as the recently introduced Virtual Element Method (VEM) [9, 10, 28]). This method was applied in [13–15], where advantage is taken from the flexibility of virtual elements to easily generate a conforming polygonal meshes. Another approaches include classic Finite Elements [4], gradient schemes [26], upscaling [45], multi-point flux approximation [1, 58] and the Finite Volume Method [38, 56].

The use of a mixed formulation in DFN simulations is a widely common choice, for the possibility of a direct computation of the flux variable velocity [2, 3, 5, 34, 46, 49, 55, 60]. Conservation of mass is strongly enforced in a mixed formulation, so that the accuracy is very good when approximating the flux variable. This is particularly important when this information is to be used as the underlying convection field of an advection-diffusion process of a passive scalar, *e.g.* when studying the concentration of a certain pollutant.

Lately, and due to the large advances in simulating complex DFNs, there has been a trend towards including the rock matrix in the simulation. For instance, in [19] a 3D simulation of porous media and embedded fractures is solved using a constrained minimization approach for a combined primal FEM and BEM (Boundary Element Method) discretization, lowest order mixed VEM in [43] and a 3D primal VEM approach was used to impose matching conditions between 3D blocks and 2D fractures in [17].

The main topic of this thesis is to combine and extend what was put forward in [15] and [16], by taking the globally conforming approach of the former work combined with the mixed VEM discretization of the latter. It is organized as follows: in Chapter 2 the formulation for the problem at hand is presented. Chapter 3 is devoted to providing a description of the mixed formulation of the Virtual Element Method for general second order elliptic equations and for hybrid 3D-2D domains.

A guide for the implementation of the method is given afterwards, in Chapter 4, including meshing process, computational details and constraint impositions. Next, the main original component and longest part of the thesis consisting of numerical results is contained in Chapter 5, where convergence is studied for 3D problems, where also pure DFN problems and hybrid problems are solved and their results analysed. The work ends with some concluding remarks in Chapter 6.

Chapter 2

Problem formulation

2.1 Problem statement

In this work the focus will be put on solving a second order elliptical differential equation over a domain $\Omega \subset \mathbb{R}^d$, with $d = 2, 3$. The boundary $\partial\Omega$ is divided in $\partial\Omega = \Gamma_D \cup \Gamma_N$, that stand for Dirichlet and Neumann boundaries respectively, and $\hat{\boldsymbol{n}}$ is the outward pointing normal vector to the boundary. The data for the problem consists of a smooth positive definite symmetric tensor $\mathfrak{a}(\boldsymbol{x}) \geq \mathfrak{a}_0 > 0$ for all $\boldsymbol{x}$, a smooth vector valued function $\boldsymbol{b}$ and a smooth real valued function c.

Thus, the strong form of the problem is stated as: find $P \in C^2(\Omega) \cap C(\overline{\Omega})$ such that

$$\begin{aligned} \nabla \cdot (-\mathfrak{a}(\boldsymbol{x})\nabla P(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x})P(\boldsymbol{x})) + c(\boldsymbol{x})P(\boldsymbol{x}) &= f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Omega, \\ P(\boldsymbol{x}) &= g(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma_D, \\ (-\mathfrak{a}(\boldsymbol{x})\nabla P(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x})P(\boldsymbol{x})) \cdot \hat{\boldsymbol{n}} &= h(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma_N, \end{aligned} \tag{2.1.1}$$

where $f(\boldsymbol{x}) \in C(\Omega)$, $g(\boldsymbol{x}) \in C(\Gamma_D)$ and $h(\boldsymbol{x}) \in C(\Gamma_N)$ and the notation '$\nabla\cdot$' represents the divergence operator. Besides the usual Lebesgue space $\mathrm{L}^2(\Omega)$ of square-integrable functions, the following Sobolev spaces are needed in the following:

$$\begin{aligned} \mathrm{H}(\mathrm{div}, \Omega) &:= \{\boldsymbol{v} \in \left[\mathrm{L}^2(\Omega)\right]^d : \nabla \cdot \boldsymbol{v} \in \mathrm{L}^2(\Omega)\}, \\ \mathrm{H}_{\Gamma_{N,h}}(\mathrm{div}, \Omega) &:= \{\boldsymbol{v} \in \mathrm{H}(\mathrm{div}, \Omega) : \boldsymbol{v} \cdot \hat{\boldsymbol{n}}|_{\Gamma_N} = h\}, \\ \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega) &:= \{\boldsymbol{v} \in \mathrm{H}(\mathrm{div}, \Omega) : \boldsymbol{v} \cdot \hat{\boldsymbol{n}}|_{\Gamma_N} = 0\}, \end{aligned}$$

where $\hat{\boldsymbol{n}}$ represents the outward unitary normal vector of Γ_D. 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 $\eta : \mathrm{H}(\mathrm{div}, \Omega) \to \mathrm{L}^2(\Gamma_N)$. Inner products for these spaces will be denoted with the $(\cdot, \cdot)$ notation and norms in the Sobolev spaces H^k will be denoted by $||.||_{k,X}$ where the subscripts will indicate the corresponding space and domain respectively. (Notation: $\mathrm{L}^2 := \mathrm{H}^0$). In physical terminology, (2.1.1) is known as the stationary reaction-convection-diffusion equation.

By setting $\nu := \mathfrak{a}^{-1}$, $\boldsymbol{\beta} := \nu \boldsymbol{b}$ and introducing the flux variable $\boldsymbol{u} := -\mathfrak{a}\nabla P + \boldsymbol{b}P$, (2.1.1) can be rewritten as

$$\begin{aligned} \nabla \cdot \boldsymbol{u}(\boldsymbol{x}) + c(\boldsymbol{x})P(\boldsymbol{x}) &= f(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Omega, \\ P(\boldsymbol{x}) &= g(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma_D, \\ \boldsymbol{u}(\boldsymbol{x}) \cdot \hat{\boldsymbol{n}} &= h(\boldsymbol{x}) \quad \forall \boldsymbol{x} \in \Gamma_N, \end{aligned} \tag{2.1.2}$$

In order to deal with non homogeneous Neumann boundary conditions, we define $\boldsymbol{u} = \hat{\boldsymbol{u}} + \boldsymbol{u_0}$ with $\hat{\boldsymbol{u}} \in \mathrm{H}_{\Gamma_{N,h}}$ and $\boldsymbol{u_0} \in \mathrm{H}_{\Gamma_{N,0}}$. Finally, by multiplying the strong form (2.1.2) by test functions $\boldsymbol{v} \in \mathrm{H}_{\Gamma_{N,0}}$ and $Q \in \mathrm{L}^2(\Omega)$, the variational form of the problem is obtained:

Find $\boldsymbol{u_0} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega)$ and $P \in \mathrm{L}^2(\Omega)$ such that

$$\begin{aligned} &(\nu \boldsymbol{u_0}, \boldsymbol{v})_{2,\Omega} - (P, \nabla \cdot \boldsymbol{v})_{2,\Omega} - (\boldsymbol{\beta} \cdot \boldsymbol{v}, P)_{2,\Omega} = (\boldsymbol{v} \cdot \hat{\boldsymbol{n}}, g)_{2,\Gamma_D} - (\nu \hat{\boldsymbol{u}}, \boldsymbol{v})_{2,\Omega}, \ \forall \boldsymbol{v} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega), \\ &(\nabla \cdot \boldsymbol{u_0}, Q)_{2,\Omega} + (cP, Q)_{2,\Omega} = (f, Q)_{2,\Omega} - (\nabla \cdot \hat{\boldsymbol{u}}, Q)_{2,\Omega}, \quad \forall Q \in \mathrm{L}^2(\Omega), \end{aligned} \tag{2.1.3}$$

where $f(\boldsymbol{x}) \in \mathrm{L}^2(\Omega)$, $g(\boldsymbol{x}) \in \mathrm{L}^2(\Gamma_D)$ and $h(\boldsymbol{x}) \in \mathrm{L}^2(\Gamma_N)$. The bilinear forms for this problem are $a(\boldsymbol{u}, \boldsymbol{v}) := (\nu \boldsymbol{u}, \boldsymbol{v})_{2,\Omega}$ and $b(P, \boldsymbol{v}) := (P, \nabla \cdot \boldsymbol{v})_{2,\Omega}$. 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 (2.1.3) is well posed [24].

For a particular application we can consider the case of single-phase flow in porous media in the presence of lower-dimensional geometrical entities (which in the context of this work context will be 2D planes in 3D domains), modeled by means of Darcy's law. In that case, there are no convection nor reaction term, and the positive definite diffusion tensors $\mathfrak{a}$ represents the ratio of permeability of the medium and the viscocity of the fluid. The strong and variational forms in this case, considering the same hipothesis and notation as before, are as follows:

$$\begin{cases} \boldsymbol{u}(\boldsymbol{x}) = -\mathfrak{a}\nabla P(\boldsymbol{x}) \quad \forall x \in \Omega, \\ \nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = f(\boldsymbol{x}) \quad \forall x \in \Omega, \\ P(\boldsymbol{x}) = g(\boldsymbol{x}) \quad \forall x \in \Gamma_D, \\ \boldsymbol{u}(\boldsymbol{x}) \cdot \hat{\boldsymbol{n}} = h(\boldsymbol{x}) \quad \forall x \in \Gamma_N, \end{cases} \tag{2.1.4}$$

Find $\boldsymbol{u_0} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega)$ and $P \in \mathrm{L}^2(\Omega)$ such that

$$\begin{aligned} &(\nu \boldsymbol{u_0}, \boldsymbol{v})_{2,\Omega} - (P, \nabla \cdot \boldsymbol{v})_{2,\Omega} = (\boldsymbol{v} \cdot \hat{\boldsymbol{n}}, g)_{2,\Gamma_D} - (\nu \hat{\boldsymbol{u}}, \boldsymbol{v})_{2,\Omega} \ \forall \boldsymbol{v} \in \mathrm{H}_{\Gamma_{N,0}}(\mathrm{div}, \Omega), \\ &(\nabla \cdot \boldsymbol{u_0}, Q)_{2,\Omega} = (f, Q)_{2,\Omega} - (\nabla \cdot \hat{\boldsymbol{u}}, Q)_{2,\Omega} \quad \forall Q \in \mathrm{L}^2(\Omega), \end{aligned} \tag{2.1.5}$$

2.2 Mixed dimensional coupling

In this work we will focus only on the intermensional coupling between 3D elements and 2D planar fractures. It is possible to extend this quite straightforwardly by adding the interaction between 2D fractures and the segments defined by their intersection (called 'traces'). Furthermore, intersection between traces define points which can be coupled as well, giving rise to several dimensional hierarchies in the model [53]. A model for solving flow in DFN with arbitrary order mixed virtual elements was presented in [16]. An study of the application of the lowest order mixed VEM [44] and a solution of a hybrid-dimensional for flow and transport problems using is applied in [43]. For other sources providing a complete descriptions of problems within this framework, the reader is referred to [25, 27, 32, 34, 41, 42, 59]. Figure 4.1.1 clarifies the notation.

Strong form of the hybrid 3D-2D problem

We will consider Ω_3 representing the three-dimensional domain and F_r, with $r = 1, ..., N_f$ will stand for the fractures. Fracture intersections will be denoted by T_t, $t = 1, ..., N_t$, with T_t^1 and T_t^2 standing for the indices of the fractures such that $F_{T_t^1} \cap F_{T_t^2} = T_t$. Moreover, superscripts $2, 3$ make reference to the corresponding dimensional domain and r to the respective fracture F_r. The strong form of the problem is stated (implicitly acknowledging the dependence on the spatial variable $\boldsymbol{x}$)

as:

$$\left\{\begin{array}{l} \underline{\text{3D domain}} \\ \boldsymbol{u}^3 = -\mathfrak{a}^3 \nabla P^3, \quad \text{in } \Omega^3 \\ \textit{Darcy's law} \\ \nabla \cdot \boldsymbol{u}^3 = f^3, \quad \text{in } \Omega^3 \\ \textit{Mass conservation} \\ P^3|_{\Gamma_D} = g^3, \quad \text{on } \Gamma_D^3 \\ \textit{Dirichlet BC} \\ \boldsymbol{u}^3 \cdot \hat{\boldsymbol{n}}^3 = h^3, \quad \text{on } \Gamma_N^3 \\ \textit{Neumann BC} \end{array}\right. \rightarrow \left\{\begin{array}{l} \underline{\text{Fracture } F_r} \\ \boldsymbol{u}^{2,r} = -\mathfrak{a}^{2,r} \nabla P^{2,r}, \quad \text{in } F_r \\ \textit{Darcy's law} \\ \nabla \cdot \boldsymbol{u}^{2,r} = f^{2,r} + [\![\boldsymbol{u}^3 \cdot \hat{\boldsymbol{n}}^{3,r}]\!], \quad \text{in } F_r \\ \textit{Mass conservation} \\ P^{2,r}|_{\Gamma_D^{2,r}} = g^{2,r}, \quad \text{on } \Gamma_D^{2,r} \\ \textit{Dirichlet BC} \\ \boldsymbol{u}^{2,r} \cdot \hat{\boldsymbol{n}}^2 = h^{2,r}, \quad \text{on } \Gamma_N^{2,r} \\ \textit{Neumann BC} \end{array}\right. \tag{2.2.1}$$

where $\hat{\boldsymbol{n}}^{3,r}$ is the normal vector to fracture F_r with a globally defined orientation, $\hat{\boldsymbol{n}}_T^{2,s}$ is the normal to trace T embedded in fracture F_s and $\hat{\boldsymbol{n}}^{2,3}$ are the outward pointing normals of the 3D and 2D domains respectively and $[\![\cdot]\!]$ represents the jump of a function. The gradient and divergence operator, as well as the permeability tensor, are intended to operate on a local 2D coordinate system when defined on a fracture. The term in red indicates the contribution of mass to the fracture provided by the 3D matrix. The functions g and h stand for the known values for the Dirichlet and Neumann boundary conditions respectively, where the subscripts indicate matrix or fracture. The complete problem requires the addition of the following coupling terms:

$$\left\{\begin{array}{l} \underline{\text{Matrix-Fracture}} \\ \boldsymbol{u}_+^3 \cdot \hat{\boldsymbol{n}}^{3,r} = \eta_r (P_+^3|_{F_r} - P^{2,r}), \quad r = 1, ..., N_f \\ \boldsymbol{u}_-^3 \cdot \hat{\boldsymbol{n}}^{3,r} = \eta_r (P_-^3|_{F_r} - P^{2,r}), \quad r = 1, ..., N_f \\ \textit{Darcy's law for transverse flow} \end{array}\right. \longrightarrow \left\{\begin{array}{l} \underline{\text{Fracture-Fracture}} \\ \sum_{s=T_t^1, T_t^2} [\![\boldsymbol{u}^{2,s} \cdot \hat{\boldsymbol{n}}_{T_t}^{2,s}]\!] = 0, \quad t = 1, ..., N_t \\ \textit{Mass conservation on traces} \\ P^{2,T_t^1}|_{T_t} = P^{2,T_t^2}|_{T_t} \\ \textit{Pressure continuity on traces} \end{array}\right. \tag{2.2.2}$$

where the subscripts $+, -$ indicate the side of F_r, with respect to the globally defined normal $\hat{\boldsymbol{n}}^r$ and η_r is the normal permeability between matrix and fracture F_r. This system can be described as follows: for each dimension, a problem of Darcy flow is solved. There are many physical parameters and assumptions made to simplify the model. For instance, flow on a fracture is usually not constant along its width but depends on fracture aperture. However, from a numerical viewpoint we can assume that the coefficients and data parameters have already been homogenized and non-dimensionalized. In the lower dimension, there is a mass flux arising from the higher order entities, which is modeled as part of the loading term and is internal to the elements. For the higher order entity, there is a leakage through certain element faces resulting in 'internal boundaries' which can

be regarded as a Robin boundary condition if the flux has a dependence on the pressure or as pure Neumann boundary condition otherwise.

The Matrix-Fracture coupling equations state that the drop of pressure is proportional to η_r, which is sort of Darcy's law for 2D-3D flux exchange. When permeability is very high and fracture width is small ($\eta_r \to \infty$), the coupling conditions can be modelled as requiring that the pressure be continuous across fractures $\Omega \cap F_r$, for $r = 1, ..., N_f$. This means that there is no jump in pressure in a path that leaves the matrix, goes through the fracture and enters the matrix again on the other side of the fracture.

The first Fracture-Fracture coupling is the conservation of mass across traces and the second is the continuity of the pressure head. Since we are using the mixed formulation, the discrete approximation will not explicitly consider the coupling of the pressure, since the pressure spaces is not even continuous on element boundaries. On the other hand, mass conservation will be strongly imposed through the flux DOFs (See 4.3). The exact opposite happens in the case of the primal formulation, where continuity of pressure is imposed and no condition for the flux is imposed in the discretization [19].

Weak form of the hybrid 3D-2D problem

Establishing the weak form for this problem [5] requires the introduction of considerable notation and conceptually has not much interest if the aim is the application of the method. We will assume that homogeneous Neumann conditions are applied to all fracture boundaries and homogeneous Dirichlet boundary conditions on $\partial\Omega_3$ to simplify the notation. The inclusion of other boundary conditions is standard and follows what was shown in (2.1.3). The functional spaces for this problem are $V = \left\{\mathrm{H}(\mathrm{div}, \Omega_3) \times \mathrm{H}_0(\mathrm{div}, F_1) \times \cdots \times \mathrm{H}_0(\mathrm{div}, F_{N_f})\right\}$, $Q = \left\{\mathrm{L}^2(\Omega_3) \times \mathrm{L}^2(F_1) \times \cdots \times \mathrm{L}^2(F_{N_f})\right\}$ and $L = \left\{\mathrm{H}^{-1/2}(T_{N_1}) \times \cdots \times \mathrm{H}^{-1/2}(T_{N_t})\right\}$ equipped as usual with the induced norm whose square is the sum of the squares of the norm of each individual space. The space L will basically consist of delta functions that impose the pointwise continuity of the flux variable on fracture intersections to preserve mass balance. The symbols $\nu^3 = (\mathfrak{a}^3)^{-1}$ and $\nu^{2,r} = (\mathfrak{a}^{2,r})^{-1}$ represent the permeability of the rock and the tangential permeability of the fractures respectively.

Incorporating the coupling conditions results in the following weak form:

$$
\begin{aligned}
&\text{Find } \{\boldsymbol{u}, P, \lambda\} = \left[\left\{\boldsymbol{u}^3, \boldsymbol{u}^{2,1}, \cdots, \boldsymbol{u}^{2,N_f}\right\}, \left\{P^3, P^{2,1}, \cdots, P^{2,N_f}\right\}, \left\{\lambda^{1,1}, \cdots, \lambda^{1,N_t}\right\}\right] \in V \times Q \times L \\
&\quad \text{such that:} \\
&\quad (\nu^3 \boldsymbol{u}^3, \boldsymbol{v}^3)_{2,\Omega_3} + \sum_{r=1,\ldots,N_f} (\eta_r \boldsymbol{u}^3 \cdot \hat{\boldsymbol{n}}^{3,r}, \boldsymbol{v}^3 \cdot \hat{\boldsymbol{n}}^{3,r})_{2,F_r^{\pm}} - (P^3, \nabla \cdot \boldsymbol{v}^3)_{2,\Omega_3} = 0, \quad \forall \boldsymbol{v}^3 \in \mathrm{H}_0(\mathrm{div}, \Omega_3), \\
&\quad (\nabla \cdot \boldsymbol{u}^3, Q^3)_{2,\Omega_3} = (f^3, Q^3)_{2,\Omega_3}, \quad \forall Q^3 \in \mathrm{L}^2(\Omega_3) \\
&\text{For } F_r,\ r = 1, ..., N_f \text{ and for } T_t,\ t = 1, ..., N_t \\
&\quad (\nu^{2,r} \boldsymbol{u}^{2,r}, \boldsymbol{v}^{2,r})_{2,F_r} + (\lambda^{1,t}, \sum_{s=T_t^1, T_t^2} [\![\boldsymbol{v}^{2,s} \cdot \hat{\boldsymbol{n}}_t^{2,s}]\!])_{2,T_t} + ([\![P^{2,r}, \boldsymbol{v}^3 \cdot \hat{\boldsymbol{n}}^{3,r}]\!])_{2,F_r} \\
&\qquad -(P^{2,r}, \nabla \cdot \boldsymbol{v}^{2,r})_{2,F_r} = 0, \quad \forall \boldsymbol{v}^{2,r} \in \mathrm{H}_0(\mathrm{div}, F_r), \\
&\quad (\nabla \cdot \boldsymbol{u}^{2,r}, Q^{2,r})_{2,F_r} + ([\![\boldsymbol{u}^3 \cdot \hat{\boldsymbol{n}}^{3,r}]\!], Q^{2,r})_{2,F_r} = (f^2, Q^2)_{2,\Omega_3}, \quad \forall Q^{2,r} \in \mathrm{L}^2(F_r), \\
&\quad (\sum_{s=T_t^1, T_t^2} [\![\boldsymbol{u}^{2,t} \cdot \hat{\boldsymbol{n}}_t^{2,s}]\!], \mu)_{2,T_t} = 0, \quad \forall \mu^{1,t} \in \mathrm{H}^{-1/2}(T_t),
\end{aligned}
\tag{2.2.3}
$$

The colouring makes reference to the coupling conditions in (2.2.2) and the values of $\lambda^{1,t}$ will represent the pressure on trace T_t and act as a Lagrange multiplier for imposing flux continuity on traces. The proof of the wellposedness and convergence properties of these type of hybrid problems is highly technical and very notation-heavy, but conceptually is very straightforward since the inf-sup condition can be easily proven once all the required ingredients have been properly defined. The reader is referred to [37, 41, 49]. A description of the discretization of problem (2.2.3) is given in Chapter 4.

A hint to more general cases

It is clear from (2.2.1) and (2.2.2) that the coupling process could continue by introducing 1D elements representing flow along the traces which need to be coupled with flow on the fractures, just as flow in the matrix is coupled with flow on the fractures. In this case, the first equation may also have a source term instead of 0 and the pressure could display a jump between its value on the fractures and the trace they determine. The final coupling would be at the points defined by the intersection of traces. Furthermore, the procedure presented above can also be applied (with some generalizations) to second order elliptic equations. In addition, the time dependent parabolic equation could be considered, where the semi-discretization in time is done exactly as in standard FEM. With this modifications, the case of the transport of scalar quantity could be studied for instance.

Chapter 3

The Mixed Virtual Element Method

3.1 Mixed Virtual Element Method

This chapter provides an outline of the main definitions and features of the mixed formulation of VEM. Its initial introduction is given in [28], with a follow-up work generalizing the method [10] and a related work on virtual H(Div) and H(Curl) spaces is [12]. Despite its recent introduction, a variety of applications can be found in the scientific literature. Namely, Stokes flow in [31] and [29], the Brinkman problem [30], plane elasticity [7] and flow in Discrete Fracture Networks [16].

3.1.1 Overview of the method

Definitions

The mesh of a domain Ω comprised of arbitrary polyhedra of mesh parameter h is indicated by $\mathcal{T}_h$, and it satisfies basic conditions of regularity. The non negative integer k stands for the chosen order of interpolation of the method. In a nutshell, a Virtual Element will be comprised of an element P, a set of DOFs and a discrete functional space defined on P, as in standard FEM. The difference will be that P may be an arbitrary polyhedral and the shape functions will (almost) never be explicitly known. 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 definition of the local VEM space for the velocity variable in an element $E \in \mathcal{T}_h$ allows some play between different parameters which result in slightly different local spaces. By defining the parameters $[k_f, k_d, k_r] \geq 0$ representing a kind of "interpolation order", the local space is

$$V_{k,h}^E = \{\boldsymbol{v}_h \in \mathrm{H}(\mathrm{div}, E) : \boldsymbol{v}_h \cdot \boldsymbol{n}|_f \in \mathbb{P}_{k_f}(e)\ \forall f \in \partial E,\ \mathrm{div}(\boldsymbol{v}_h) \in \mathbb{P}_{k_d}(E),\ \text{and } \mathrm{rot}(\boldsymbol{v}_h) \in \mathbb{P}_{k_r}(E)\}, \tag{3.1.1}$$

where $\mathbb{P}_{-1} = 0$. For the choice $k_f = k$, $k_d = k-1$ and $k_r = k-1$, the local space will resemble an extension of the BDM elements ($k \geq 1$), while taking $k_f = k$, $k_d = k$ and $k_r = k-1$ ($k \geq 0$) is a natural generalization of the classical RT elements of order k. Even the choice $k_r = 0$ can be considered, where the approximation will be such that solutions which are gradients of polynomials will still retain optimal convergence. Although other combinations are possible [12], only these 2 will be considered in the following which will be denoted 'BDM-VEM' and 'RT-VEM'.

For two-dimensional virtual elements that will comprise the fractures, only RT-type VEM will be considered. The reason for this is that the coupling between 3D and 2D elements (Section 2.2) is done through the faces, so that the jump in the flux variable across faces of adjacent 3D elements crossed by a fracture becomes the load term for the 2D element. In order to have compatible orders of interpolation, the flux variable across the faces in a 3D element should have the same order of the pressure variable for the 2D element, so that for instance 2D RT0-VEM are compatible with 3D RT0-VEM but not with any 3D BDM-VEM. In general, all 2D RTk-VEM are compatible with 3D RTk-VEM and 3D BDMk-VEM, for $k > 0$. However, if we had 2D BDM-VEM whose pressure discretization is an order lower than the velocity, the elements would need to be an order higher to compatibilize with the 3D elements, namely 2D BDM(k+1)-VEM are compatible with 3D RTk-VEM and 3D BDMk-VEM. Therefore, this requirement of higher order is not justified for mixed dimensional problems since the advantage of having a higer interpolation order for the 2D velocity is reduced by the lower order coupling of dimensions. The global space is

$$V_{k,h} := \left\{\boldsymbol{v}_h \in \mathrm{H}(\mathrm{div}, \Omega) : \boldsymbol{v}_{h|E} \in V_{k,h}^E\ \forall E \in \mathcal{T}_h\right\},$$

resulting in a $\mathrm{H}(\mathrm{div}, \Omega)$ conforming space The global VEM space for the pressure variable is simply

$$Q_{k_d,h}(\Omega) := \left\{Q_h \in \mathrm{L}^2(\Omega) : Q_h|_E \in \mathbb{P}_{k_d}(E)\ \forall E \in \mathcal{T}_h\right\}$$

for which there is no requirement of continuity.

There are many possibilities for the choice of degrees of freedom (DOFs). Let us first recall $[\mathbb{P}_k(e), \mathbb{P}_k(f), \mathbb{P}_k(E)]$ as the local polynomial spaces of order k over an edge e, a face f or an element E, with dimensions $n_{k,1} := k+1$, $n_{k,2} := \frac{(k+1)(k+2)}{2}$ and $n_{k,3} := \frac{(k+1)(k+2)(k+3)}{6}$ respectively. A basis for these spaces are the so-called monomial bases $\mathfrak{M}$:

$$\begin{aligned}
\mathfrak{M}_k(e) &:= \left\{ \left(\frac{x-x_0}{h_e}\right)^{\alpha_1}, \quad 0 \le \alpha_1 \le k \right\}, \\
\mathfrak{M}_k(f) &:= \left\{ \left(\frac{x-x_0}{h_f}\right)^{\alpha_1} \left(\frac{y-y_0}{h_f}\right)^{\alpha_2}, \quad 0 \le \alpha_1+\alpha_2 \le k \right\}, \\
\mathfrak{M}_k(E) &:= \left\{ \left(\frac{x-x_0}{h_E}\right)^{\alpha_1} \left(\frac{y-y_0}{h_E}\right)^{\alpha_2} \left(\frac{z-z_0}{h_E}\right)^{\alpha_3}, \quad 0 \le \alpha_1+\alpha_2+\alpha_3 \le k \right\},
\end{aligned}$$

where x_0, y_0 and z_0 are the coordinates of the centroid of the edge/face/element and $h_{e,f,E}$ is its diameter. This scaling for the monomials is needed since all computations will be done on the actual geometry of the element as explained in Chapter 4. A local coordinate system is considered in the 1D and 2D cases for each edge/face, with origin in the barycentre of the edge/face. The first 3 monomial spaces in 3D are:

$$\begin{aligned}
\mathfrak{M}_0(E) &= \{1\}, \\
\mathfrak{M}_1(E) &= \mathfrak{M}_0(E) \cup \left\{ \left(\frac{x-x_0}{h_E}\right), \left(\frac{y-y_0}{h_E}\right), \left(\frac{z-z_0}{h_E}\right) \right\}, \\
\mathfrak{M}_2(E) &= \mathfrak{M}_0(E) \cup \mathfrak{M}_1(E) \cup \left\{ \left(\frac{x-x_0}{h_E}\right)^2, \left(\frac{x-x_0}{h_E}\right)\left(\frac{y-y_0}{h_E}\right), \left(\frac{y-y_0}{h_E}\right)^2, \right. \\
&\qquad \left. \left(\frac{y-y_0}{h_E}\right)\left(\frac{z-z_0}{h_E}\right), \left(\frac{z-z_0}{h_E}\right)^2, \left(\frac{x-x_0}{h_E}\right)\left(\frac{z-z_0}{h_E}\right) \right\}.
\end{aligned}$$

We then define the space

$$\nabla\mathbb{P}_{k+1}(E) := \left\{ \boldsymbol{g} \in (\mathbb{P}_k(E))^d \text{ such that } \boldsymbol{g} = \nabla \hat{m} \text{ for some } \hat{m} \in \mathbb{P}_{k+1}(E) \right\}. \tag{3.1.2}$$

This space has dimension $n_{k,\nabla} := n_{k,3} - 1$. As an example, in the case of $k = 1$ a generic polynomial of degree $k+1$ in 3 variables can be written as $\hat{m} = c_1 + c_2x + c_3y + c_4z + c_5x^2 + c_6xy +$

TABLE 3.1.1: Dimensions for various polynomial spaces for different orders of accuracy

2D k	0	1	2	3	4	5	6
$\dim(\mathbb{P}_k(E))$	1	3	6	10	15	21	28
$\dim(\mathbb{P}_k(E))^2$	2	6	12	20	30	42	56
$\dim(\nabla\mathbb{P}_k(E))$	0	2	5	9	14	20	27
$\dim(\nabla\mathbb{P}_{k+1}(E))$	2	5	9	14	20	27	35
$\dim(\nabla\mathbb{P}_{k+1}(E))^{\oplus}$	0	1	3	6	10	15	21

3D k	0	1	2	3	4	5	6
$\dim(\mathbb{P}_k(E))$	1	4	10	20	35	56	84
$\dim(\mathbb{P}_k(E))^3$	3	12	30	60	105	168	252
$\dim(\nabla\mathbb{P}_k(E))$	0	3	9	19	34	55	83
$\dim(\nabla\mathbb{P}_{k+1}(E))$	3	9	19	34	55	83	120
$\dim(\nabla\mathbb{P}_{k+1}(E))^{\oplus}$	0	3	11	26	60	85	132

$c_7y^2 + c_8yz + c_9z^2 + c_{10}xz$ and therefore:

$$\nabla\mathbb{P}_0(E) = \emptyset$$

$$\nabla\mathbb{P}_0^{\oplus}(E) = \emptyset$$

$$\nabla\mathbb{P}_1(E) = \langle[1\ 0\ 0], [0\ 1\ 0], [0\ 0\ 1]\rangle$$

$$\nabla\mathbb{P}_1^{\oplus}(E) = \emptyset$$

$$\nabla\mathbb{P}_2(E) = \nabla\mathbb{P}_1 \cup \langle[2x\ 0\ 0], [y\ x\ 0], [0\ 2y\ 0], [0\ z\ y], [0\ 0\ 2z], [z\ 0\ x]\rangle$$

$$\nabla\mathbb{P}_2^{\oplus}(E) = \langle[y\ -x\ 0], [z\ 0\ -x], [0\ z\ -y]\rangle$$

where by $(\nabla\mathbb{P}_{k+1}(E))^{\oplus}$ we denote the L^2 orthogonal complement of $\nabla\mathbb{P}_{k+1}(E)$ in $(\mathbb{P}_k(E))^d$ so that $(\mathbb{P}_k(E))^d = (\nabla\mathbb{P}_{k+1}(E)) \oplus (\nabla\mathbb{P}_{k+1}(E))^{\oplus}$. Its dimension is $n_{k,\oplus} := dn_{k,d} - n_{k,\nabla}$ and of course the choice of basis for this space is not unique. A summary of the cardinality of the different spaces in 2D and 3D are given in Table 3.1.1 while a graphical representation of the polinomial basis known as Pascals's pyramid and listings of all polynomials comprising the space of gradients and its complement are found in the Appendix (Figure A.1.1, Tables A.1.1 and A.1.2).

Degrees of freedom

The DOFs for $\boldsymbol{v}_h \in V_{k,h}^E$, following the choice in [10], are :

$$\begin{aligned} &i.\ \frac{1}{|f|}\int_E (\boldsymbol{v}_h \cdot \boldsymbol{n}_f) m_r \ \mathrm{dV} && m_r \in \mathbb{P}_{k_f}(f), \quad r \in \{1, ..., n_{k,2}\}, \quad \forall f \in \partial E \\ &ii.\ \frac{1}{|E|}\int_E \boldsymbol{v}_h \cdot \boldsymbol{g}_\beta^\nabla \ \mathrm{dV} && \boldsymbol{g}_\beta^\nabla \in (\nabla \mathbb{P}_{k_d}(E)), \quad \beta \in \{1, ..., n_{k,\nabla}\} \\ &iii.\ \frac{1}{|E|}\int_E \boldsymbol{v}_h \cdot \boldsymbol{g}_\gamma^\oplus \ \mathrm{dV} && \boldsymbol{g}_\gamma^\oplus \in (\nabla \mathbb{P}_{k_r+1}(E))^\oplus, \quad \gamma \in \{1, ..., n_{k,\oplus}\} \end{aligned} \tag{3.1.3}$$

where $\boldsymbol{n}_f$ is the outward-pointing normal of face f, $|f|$ its area and $|E|$ the volume of the element. In 2 dimensions $f \in \partial E \to e \in \partial E$, so faces become edges, areas become lengths and volumes become areas. In general, type iii) DOFs are very cheap to compute due to the definition itself, usually avoiding the need for integration since the values of the integrals are a priori known by definition. A proof of unisolvence can be seen in [10, 28] for BDM- and RT-BEM respectively, but for more general 'Face' spaces the main reference is [12]. The first set of DOF can be replaced by any other way to fix a polynomial of two variables of degree k on a face. In Table 3.1.1 we present the dimensions of some of the polynomial spaces involved in the definition of the DOF.

In 2D, an RT-VEM of n_e edges has $[n_{k,2}n_e] + [n_{k,2} - 1] + [2n_{k,2} - n_{k,\nabla}]$ DOFs. In 3D, an RT-VEM of n_f faces has $[n_{k,2}n_f] + [n_{k,3} - 1] + [3n_{k,3} - n_{k,\nabla}]$ DOFs while a BDM-VEM has $[n_{k,2}n_f] + [n_{k-1,3} - 1] + [3n_{k,3} - n_{k,\nabla}]$. The square brackets were added to separate DOFs of each type. Figures 3.1.1 and 3.1.2 graphically exemplify the DOFs for a conver hexagon and convex polyhedron of 11 faces, where face normals are depicted. DOFs of types i, ii and iii are represented in blue, red and green respectively while internal pressure DOFs are in black (for the 2D case, see [16]).

For the pressure space, the set of DOF can be trivially define as for example $k+1$ point values. For computation purposes, the choice for this implementation was to take the $n_{k,d}$ moments with respect to the monomial basis of order k, but in general any set of DOFs that univocally determine a polynomial oof order k in d variables is equivalent.

L^2 Projector operator

As in the case of the primal VEM formulation, the shape functions in the local VEM space are not explicitly known except for its values on the DOF. The crucial idea is to define a projection operator that will allow the computation of approximate discrete bilinear forms that will be stable, consistent and will retain the rate of convergence of standard finite elements. The projection

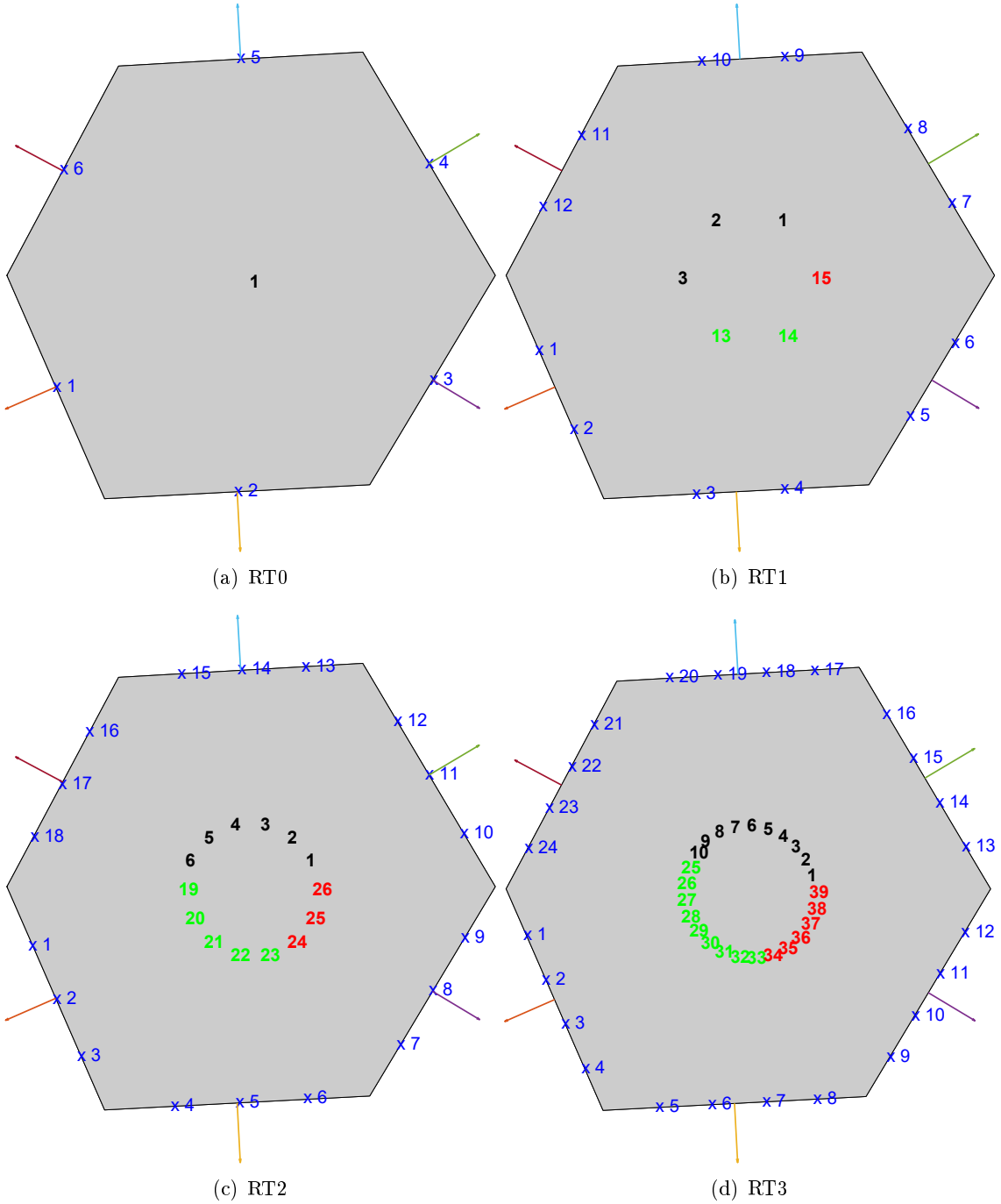
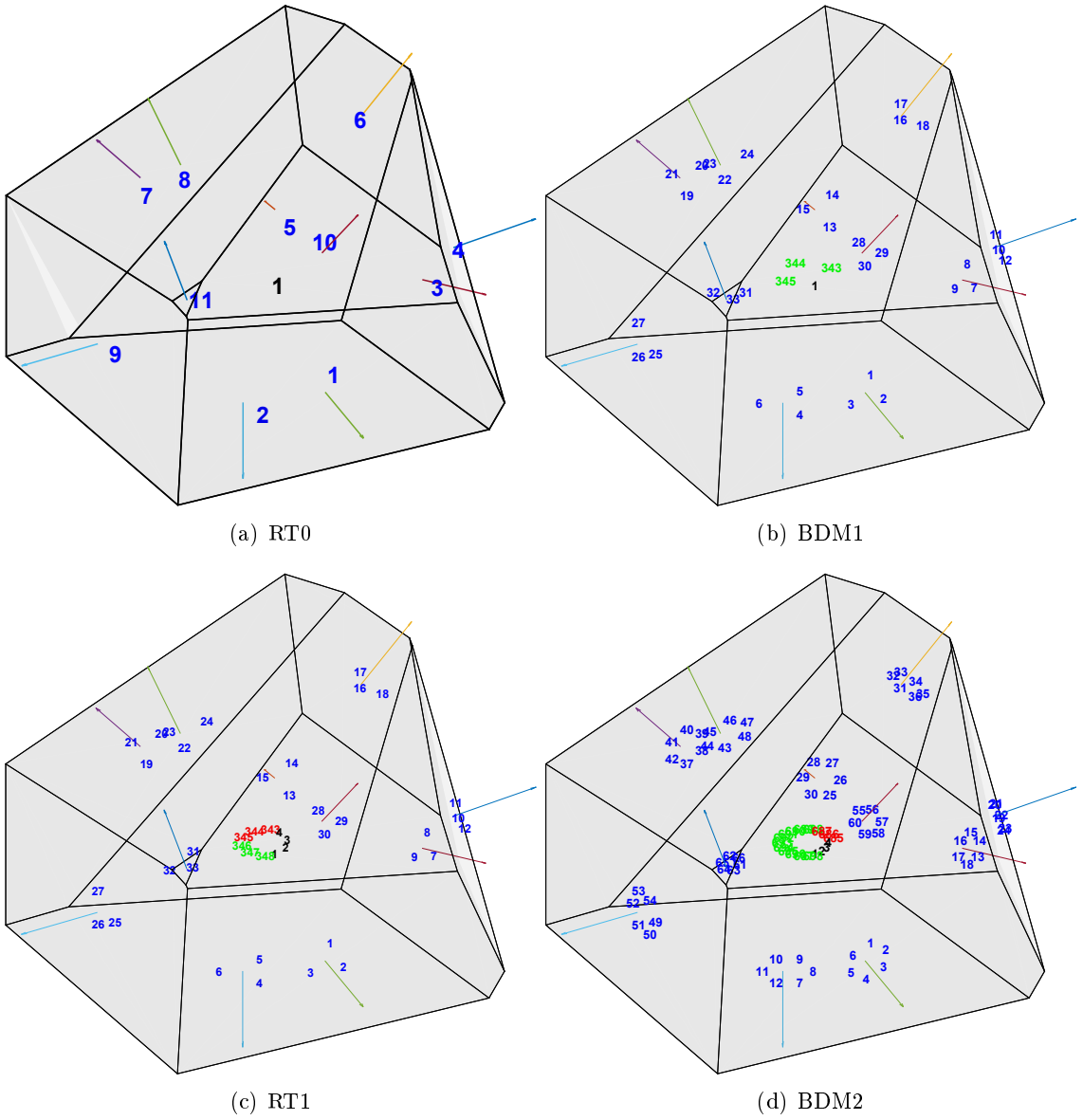


FIGURE 3.1.1: DOFs for the lowest order elements



(a) RT0 (b) BDM1

(c) RT1 (d) BDM2

FIGURE 3.1.2: DOFs for the lowest order elements

operator $\mathbf{\Pi}_k^0 : \ V_{k,h}^E \to (\mathbb{P}_k(E))^d$ will be defined as:

$$\int_E \mathbf{\Pi}_k^0 \boldsymbol{v}_h \cdot \boldsymbol{g} \ \mathrm{dV} \ = \int_E \boldsymbol{v}_h \cdot \boldsymbol{g} \ \mathrm{dV} \qquad \forall \boldsymbol{g} \in (\mathbb{P}_k(E))^d \,, \tag{3.1.4}$$

where the symbols $\mathbf{\Pi}_k^0$ and Π_k^0 are used when referring to the operator and its matrix expression respectively. It basically means that the bilinear form will be computed exactly when the local shape function is tested against a polynomial, which can be akin to passing the 'path-test' used by engineers for elasticity problems. In other words, the energy term (from the $a(\cdot,\cdot)$ bilinear form) is computed exactly using the projector for polynomial trial functions.

TABLE 3.1.2: Number of degrees of freedom for local VEM spaces in 2D and 3D

	DOF	2D RT-VEM		3D RT-VEM		3D BDM-VEM	
k	type	$V_{k,h}^E$	$Q(E)$	$V_{k,h}^E$	$Q(E)$	$V_{k,h}^E$	$Q(E)$
	i	n_e		n_f			
0	ii	0	1	0	1	-	-
	iii	0		0			
	i	$2n_e$		$3n_f$		$3\ n_f$	
1	ii	2	3	3	4	0	1
	iii	1		3		3	
	i	$3n_e$		$6n_f$		$6n_f$	
2	ii	5	6	9	10	3	4
	iii	3		11		11	
	i	$4n_e$		$10n_f$		$10n_f$	
3	ii	9	10	19	20	9	10
	iii	6		26		26	
	i	$5n_e$		$15n_f$		$15n_f$	
4	ii	14	15	34	35	19	20
	iii	10		50		50	
	i	$6n_e$		$21n_f$		$21n_f$	
5	ii	20	21	55	56	34	35
	iii	15		85		85	
	i	$7n_e$		$28n_f$		$28n_f$	
6	ii	27	28	83	84	55	56
	iii	21		133		133	

We will now show that the knowledge of the DOF is enough to compute the projector. The left hand side of (3.1.4) is an integral between polynomials in d dimensions and can be explicitly computed. The right hand side however, is 'a priori' not computable. However, since $(\mathbb{P}_k(E))^2 = (\nabla\mathbb{P}_{k+1}(E)) \oplus (\nabla\mathbb{P}_{k+1}(E))^\perp$, we can find $\tilde{\boldsymbol{g}} \in (\nabla\mathbb{P}_{k+1}(E))$ and $\boldsymbol{g}^\perp \in (\nabla\mathbb{P}_{k+1}(E))^\perp$ such that $\boldsymbol{g} = \tilde{\boldsymbol{g}} + \boldsymbol{g}^\perp$. Thus,

$$\int_E \boldsymbol{v}_h \cdot \boldsymbol{g} \ \mathrm{dV} \ = \int_E \boldsymbol{v}_h \cdot \tilde{\boldsymbol{g}} \ \mathrm{dV} \ + \int_E \boldsymbol{v}_h \cdot \boldsymbol{g}^\perp \ \mathrm{dV} \ . \tag{3.1.5}$$

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

$$\int_E \boldsymbol{v}_h \cdot \tilde{\boldsymbol{g}} \ \mathrm{dV} \ = \int_E \boldsymbol{v}_h \cdot \nabla\hat{m} \ \mathrm{dV} \ = -\int_E \mathrm{div}(\boldsymbol{v}_h)\hat{m} \ \mathrm{dV} \ + \int_{\partial E} \boldsymbol{v}_h \cdot \boldsymbol{n}|_f \hat{m} \ \mathrm{dS}. \tag{3.1.6}$$

Once again, the second term on the right hand side can be computed directly as an integration on the faces/edges of the 3D/2D element, by using the DOF of type i. In the case of the first term, if $\mathrm{div}(\boldsymbol{v}_h) \in \mathbb{P}_k$ were known, the integral would involve the product of known polynomials and thus

computable. Therefore, recalling that $\operatorname{div}(\boldsymbol{v}_h) \in \mathbb{P}_k(E)$, the following equation uniquely determines $\operatorname{div}(\boldsymbol{v}_h)$:

$$\int_E \operatorname{div}(\boldsymbol{v}_h) q \, \mathrm{dV} = -\int_E \boldsymbol{v}_h \cdot \nabla q \, \mathrm{dV} + \int_{\partial E} \boldsymbol{v}_h \cdot \boldsymbol{n}|_f q \, \mathrm{dS} \qquad \forall q \in \mathbb{P}_k(E). \tag{3.1.7}$$

Note that using the set of DOFs of type i and ii, the right hand side term of (3.1.7) can be readily computed. In conclusion, $\operatorname{div}(\boldsymbol{v}_h)$ is computed using (3.1.7), which is subsituted in (3.1.6), which in turn is needed in (3.1.5) that allows the computation of the projector using (3.1.4). Finally, we arrive at our discrete local bilinear form, which for $\boldsymbol{u}_h, \boldsymbol{v}_h \in V_{k,h}^E$ is:

$$a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h) := (\nu \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h, \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h)_E + S^E(\boldsymbol{u}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h, \boldsymbol{v}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h). \tag{3.1.8}$$

where S^E stands for any symmetric and definite positive bilinear form that scales like $a^E(\boldsymbol{u}, \boldsymbol{v}) := (\nu \boldsymbol{u}, \boldsymbol{v})$. More precisely, there exist two positive constants α_* and α^* independent of the mesh and data such that

$$\alpha_* a^E(\boldsymbol{u}_h, \boldsymbol{u}_h) \leq S^E(\boldsymbol{u}_h, \boldsymbol{u}_h) \leq \alpha^* a^E(\boldsymbol{u}_h, \boldsymbol{u}_h) \qquad \forall E \in \mathfrak{T}_h \tag{3.1.9}$$

S^E is usually taken as the standard Euclidean product of the vector of values at the DOFs scaled by E and the constant $\overline{\nu}$, that approximates ν at the barycenter or as an average. More precisely,

$$S^E(\boldsymbol{u}_h, \boldsymbol{v}_h) = \overline{\nu}|E| \sum_{i=1}^{ndofs} \mathrm{DOF}_i(\boldsymbol{u}_h)\mathrm{DOF}_i(\boldsymbol{v}_h), \tag{3.1.10}$$

where DOF_i stands for evaluating at the i-th DOF on the element (see 4.2). In other words, the bilinear form S^E defines an internal product that associates the identity matrix (of with size equal to the number of DOFs of the element) to the local basis of VEM shape function $V_{k,h}^E$ defined by the DOFs (3.1.1). The global bilinear form is defined as

$$a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) := \sum_{E \in \mathcal{T}_h} a_h^E(\boldsymbol{u}_h, \boldsymbol{v}_h). \tag{3.1.11}$$

Continuity and coercivity of the bilinear form is deduced from continuity of $a(\cdot, \cdot)$ and the assumptions on S.

Lemma 3.1. *The bilinear form $a_h(\cdot, \cdot)$ is continuous and coercive in $V_{k,h}^E$*

Proof.

- Continuity of S^E Using (3.1.9), symmetry and the continuity of a:

$$S^E(\boldsymbol{u}_h, \boldsymbol{v}_h) \leq (S^E(\boldsymbol{u}_h, \boldsymbol{u}_h))^{1/2}(S^E(\boldsymbol{v}_h, \boldsymbol{v}_h))^{1/2} \leq sup(\nu)\alpha^* ||\boldsymbol{u}_h||_{0,E}||\boldsymbol{v}_h||_{0,E}.$$

- Continuity of a_h: from (3.1.8) and (3.1.11) and the continuity of S^E we have

$$\begin{aligned} |a_h(\boldsymbol{u}_h, \boldsymbol{v}_h)| &= (\nu \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h, \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h)_E + S^E(\boldsymbol{u}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h, \boldsymbol{v}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h) \\ &\leq sup(\nu)||\boldsymbol{\Pi}_k^0 \boldsymbol{u}_h||_{0,\Omega}||\boldsymbol{\Pi}_k^0 \boldsymbol{v}_h||_{0,\Omega} + sup(\nu)\alpha^* ||\boldsymbol{u}_h - \boldsymbol{\Pi}_k^0||_{0,\Omega}||\boldsymbol{v}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h||_{0,\Omega} \end{aligned}$$

  Considering error properties of the L^2 projection $||\boldsymbol{u}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h||_{0,\Omega} \leq C||\boldsymbol{v}_h||_{0,\Omega}$,

$$|a_h(\boldsymbol{u}_h, \boldsymbol{v}_h)| \leq \sup(\nu)(1 + \alpha^* \hat{C}^2)||\boldsymbol{u}_h||_{0,\Omega}||\boldsymbol{v}_h||_{0,\Omega}.$$

- Coercivity of a_h: Given the coercivity of a, the L^2 projection properties and (3.1.9),

$$|a_h(\boldsymbol{u}_h, \boldsymbol{u}_h)| \geq \inf(\nu)(||\boldsymbol{u}_h||_{0,\Omega}^2 + \alpha_* ||\boldsymbol{u}_h - \boldsymbol{\Pi}_k^0 \boldsymbol{u}_h||_{0,\Omega}^2 \geq \hat{C}||\boldsymbol{u}_h||_{0,\Omega}^2$$

□

Discrete problem

To obtain the discrete formulation of problem (2.1.3), the terms corresponding to the velocity field are approximated by the projection, while those associated with the pressure can be directly computed. The problem becomes:

Find $\boldsymbol{u}_h \in V_{k,h}(\Omega)$ and $P_h \in Q_{k_d,h}(\Omega)$ such that

$$\begin{cases} a_h(\boldsymbol{u}_h, \boldsymbol{v}_h) - (P_h, \nabla \cdot \boldsymbol{v}_h)_{0,\Omega} - (\boldsymbol{\beta} \cdot \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h, P_h)_{0,\Omega} = (\boldsymbol{v} \cdot \boldsymbol{n}, g)_{0,\Gamma_D} - (\nu \hat{\boldsymbol{u}}, \boldsymbol{v})_{0,\Omega}, \\ \qquad\qquad \forall \boldsymbol{v}_h \in V_{k,h}, \\ (\nabla \cdot \boldsymbol{u}_h, Q_h)_{0,\Omega} + (c(\boldsymbol{x})P_h, Q_h)_{0,\Omega} = (f, Q_h)_{0,\Omega} - (\nabla \cdot \hat{\boldsymbol{u}}, Q)_{0,\Omega}, \\ \qquad\qquad \forall Q_h \in Q_{k_d,h}(\Omega) \end{cases} \tag{3.1.12}$$

where the assumptions on the data is as in (2.1.3). Well posedness of this problem under the above assumptions is due to coercivity of the bilinear form $a(.,.)$ as well as the satisfaction of the $inf-sup$ condition, as shown in the following section.

3.1.2 Main results for Mixed VEM

This section will be devoted to stating the most important theoretical results concerning the mixed formulation of the Virtual Element Method. Assumptions on the data are as given previously, while the mesh $\mathcal{T}_h$ is such that there exist a constant $C_\tau > 0$ such that for any mesh parameter h and any $E \in \mathcal{T}_h$, the ratio of the shortest edge and its diameter is bigger than C_τ and that E is star-shaped with respect to a ball of radius C_τ times its diameter. These assumptions are given for 2D problems and imply that the elements will be simply connected and the number of edges of every polygon will be bounded. For 3D problems, a generalization of these two consequences will be assumed. These assumptions are needed for the results used in some of the proofs. However, it has been experimentally observed that the method is very robust with respect to mesh distortion, including small edges and angles, collapsing nodes, etc. Furthermore, the method also works with non-convex elements.

Theorem 3.2. *Original a priori error estimates for BDM-VEM Under certain assumptions, for a sufficiently small mesh parameter h and $k > 0$, the discrete problem (3.1.12) with $\beta = 0 = c$ and piecewise constant ν has a unique solution $\boldsymbol{u}_h \in V_{k,h}(\Omega)$ and $P_h \in Q_{k_d,h}$, with the following error estimates:*

$$\begin{aligned} ||p - p_h||_{0,\Omega} &\leq Ch^k(||\boldsymbol{u}||_{k,\Omega} + ||p||_{k,\Omega}) \\ ||\boldsymbol{u} - \boldsymbol{u}_h||_{0,\Omega} &\leq Ch^{k+1}(||\boldsymbol{u}||_{k+1} + ||p||_{k+1}), \\ ||\nabla \cdot (\boldsymbol{u} - \boldsymbol{u}_h)||_{0,\Omega} &\leq Ch^k|f|_{k,\Omega}. \end{aligned}$$

Proof of this problem can be found in [28] and uses duality arguments such as those found in [36]. It is important to stress that these results were proven for BDM-like VEM spaces and thus they do not hold for RT-like elements. In fact, this is clear form the convergence rate of the flux variable compared with the pressure and divergence, which are one order lower due to the approximation order of the respective spaces.

Theorem 3.3. *Original Inf-sup condition for BDM-VEM In the same context as Theorem 3.2,*

$$\exists \overline{\beta} \text{ such that } \forall h, \forall \mathcal{T}_h \quad \inf_{Q_h \in Q_{k_d,h}(\Omega)} \sup_{\boldsymbol{v}_h \in V_{k,h}} \frac{(\nabla \cdot \boldsymbol{v}, Q)_{0,\Omega}}{||Q||_{0,\Omega}||\boldsymbol{v}_h||_{(H(div),\Omega)}} \geq \overline{\beta} > 0.$$

For the proof, see [9].

Theorem 3.4. *(A priori error estimates for RT-VEM) Under certain assumptions on the data and the mesh, for a sufficiently small mesh parameter h and $k \geq 0$, the discrete problem 3.1.12 has a unique solution $\boldsymbol{u}_h \in V_{k,h}(\Omega)$ and $P_h \in Q_{k_d,h}$, with the following error estimates:*

$$||p - p_h||_{0,\Omega} \leq Ch^{k+1}(||\boldsymbol{u}||_{k+1,\Omega} + ||p||_{k+1,\Omega})$$
$$||\boldsymbol{u} - \boldsymbol{u}_h||_{0,\Omega} \leq Ch^{k+1}(||\boldsymbol{u}||_{k+1,\Omega} + ||p||_{k+1,\Omega}||)$$
$$||\nabla \cdot (\boldsymbol{u} - \boldsymbol{u}_h)||_{0,\Omega} \leq Ch^{k+1}(|f|_{k+1,\Omega} + ||p||_{k+1,\Omega}||)$$

Proof of this problem can be found in [**?**] and uses duality arguments such as those found in [36]. In this case, these results were proven for RT-like VEM spaces for general second order elliptic equations with non constant coefficients. For this equation, the coercivity of the leading term $a(\cdot,\cdot)$ and inf-sup condition were proven in the same cited work, by means of a 'Fortin-type' operator and using classical commuting diagram arguments as in Mixed FEM.

In general, it can be seen that the a priori error estimates are in line with those for standard Finite Elements, with the added feature of handling much more general meshes.

Chapter 4

Implementation

The details for an implementation of the three dimensional mixed formulation of the VEM method is provided. The main reference for this chapter is [33], from where much of the procedures provided here are based.

4.1 Meshing and conformity

For non trivial 3D geometries, the meshing process is usually a costly and complex procedure. In general it is much easier to obtain a good quality tetrahedral mesh than an hexahedral one. When working with arbitrary polygons, few libraries exist that can produce random polytope meshes. For this work, the library voro++ [57] was used, which is coded in the C++ language. Simple regular hexahedral meshes were manually generated for some tests.

In the case of embedded fractures, the original polyhedral mesh requires further processing to account for the interaction between 2D elements and faces of their 3D counterparts. In such a situation, since cutting convex polyhedral elements by planes results in a subdivision into new convex polyhedral elements, a new Virtual Element mesh is obtained which is comprised of the new elements originated from the subdivision of the original mesh. This particular feature is a clear advantage over regular FEM discretizations, which are significantly more restrictive in the possible geometries of the elements. The algorithm for the meshing-cutting procedure, although clear conceptually, is computationally very complex. When a plane transverses completely through a polyhedra, the cut is somewhat simple. However, in the presence of fractures that end inside an element, it is not so straightforward to partition the original polyhedral element in order to

represent. For this reason, no details will be given regarding the computational implementation of a mesh-cutting algorithm.

The proposed method rest on the fact that the mesh for the whole domain is 'conforming'. This term is used in this context to explain that the faces of the 3D elements coincide perfectly with elements in the planar fractures. In other words, the 2D elements are geometrically identical to the 3D faces that they intersect so that there is no need for introducing compatibility conditions or approximations on the geometry. There are other approaches in the literature, as in the references provided previously. However, this method takes full advantage of the features of the VEM in dealing with practically any geometry while retaining the simplicity of implementation and imposition of constraints between 3D and 2D geometries. A simple example is presented in Figure 4.1.1, where only 3 fractures are considered. The resulting conforming mesh is made up of six 3D elements (light yellow) representing the matrix, seven 2D elements modelling the fractures (gray) and two 1D segments (red) representing the traces.

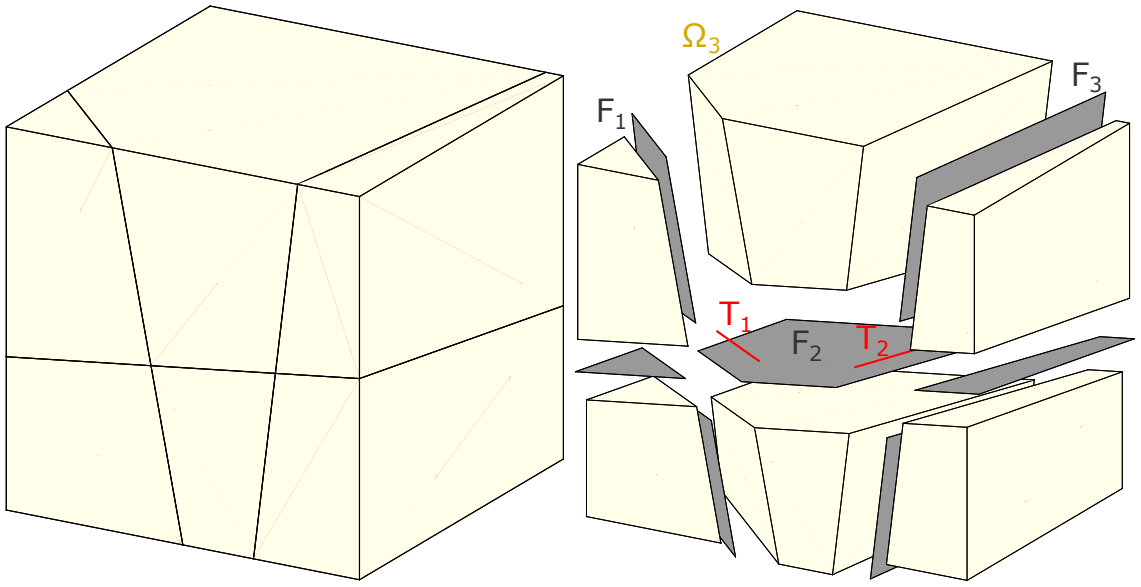


FIGURE 4.1.1: A cube and 3 fractures (left). Exploded mesh for visualization purposes (right)

4.2 Computation of stiffness matrix

In the following, it is assumed that numerical computations of integrals of known functions over 2D and 3D polytopes can be performed. The most straightforward approach is to divide a polygon into triangles, or a polyhedra into tetrahedrals and use standard Gaussian integration. Alternatives are to consider cubature [47], algebraic integration by parts or heuristic methods (like Montecarlo

integration). The procedure for computing the local matrices needed to obtain the discrete linear system (3.1.12) is explained next, with emphasis on its implementation in 3D. First we consider a basis of $(\mathbb{P}_k(E))^d = \{\boldsymbol{g}_\alpha\}_{\alpha=1,\dots,dn_{k,d}} = \left\{\boldsymbol{g}_\beta^\nabla\right\}_{\beta=1,\dots,n_{k,\nabla}} \cup \left\{\boldsymbol{g}_\gamma^\oplus\right\}_{\gamma=1,\dots,dn_{k,\oplus}}$. The local basis functions of the space $V_{k,h}^E$ will be denoted by $\{\boldsymbol{\varphi}_i\}_{i=1,\dots,n_{dof}^E}$, where n_{dof}^E is the number of DOFs for the flux variable of the element (See 3.1.1). A set of operators $\mathrm{DOF}_i : V_{k,h}^E \to \mathbb{R}$ are defined as $\mathrm{DOF}_i(\cdot)$:=evaluating $(\cdot)$ at the i-th DOF.

4.2.1 Local matrices

Firstly, several matrices will be defined, whose usefulness will become apparent later.

Matrices G and G^ν

Component-wise, $G \in \mathbb{R}^{dn_{k,d}\times dn_{k,d}}$ is defined as the product of the elements in the basis of $(\mathbb{P}_k(E))^d$

$$[G]_{pq} = \int_E \boldsymbol{g}_p \cdot \boldsymbol{g}_q \ \mathrm{dV}$$

and can be computed directly. Using the basis for $(\mathbb{P}_k(E))^d$, G can be split into

$$G = \begin{bmatrix} G^{\nabla\nabla} & G^{\nabla\oplus} \\ G^{\oplus\nabla} & G^{\oplus\oplus} \end{bmatrix}. \tag{4.2.1}$$

In the case when $\nu \neq \mathcal{I}_{d\times d}$, *i.e.*he operator is not the Laplacian, we define G^ν with the same size as G by

$$[G^\nu]_{pq} = \int_E \nu\boldsymbol{g}_p \cdot \boldsymbol{g}_q \ \mathrm{dV}.$$

Matrices H, $H^\#$ and H^c

For RT-VEM, $H \in \mathbb{R}^{n_{k,d}\times n_{k,d}}$ and $H \in \mathbb{R}^{n_{k-1,d}\times n_{k-1,d}}$ for BDM-VEM. It is the product of known polynomials over E

$$[H]_{rt} = \int_E m_r m_t \ \mathrm{dV} \tag{4.2.2}$$

and can be computed directly. For the case with reaction term $(c(\boldsymbol{x}) \neq 0)$,

$$[H^c]_{rt} = \int_E c m_r m_t \ \mathrm{dV} \tag{4.2.3}$$

with the same size as H. Similarly, $H^{\#} \in \mathbb{R}^{n_{k+1,\nabla} \times n_{k,d}}$ for RT-VEM and $H^{\#} \in \mathbb{R}^{n_{k+1,\nabla} \times n_{k-1,d}}$ for BDM-VEM. It is also the product of known polynomials over E

$$[H^{\#}]_{rt} = \int_E m_{r+1} m_t \text{ dV} \tag{4.2.4}$$

and can be computed directly. Some of the entries in H are repeated in $H^{\#}$.

Matrix W

Matrix $W \in \mathbb{R}^{n_{k,d} \times n_{dof}^E}$ for RT-VEM $W \in \mathbb{R}^{n_{k-1,d} \times n_{dof}^E}$ for BDM-VEM also involves computations with 'virtual' shape functions and is defined as

$$[W]_{ri} = \int_E m_r (\nabla \cdot \boldsymbol{\varphi}_i) \text{ dV} = -\int_E \nabla m_r \cdot \boldsymbol{\varphi}_i \text{ dV} + \int_{\partial E} (\boldsymbol{\varphi}_i \cdot \hat{\boldsymbol{n}}_E) m_r \text{ dS} \tag{4.2.5}$$

where integration by parts was used, so that

$$[W_1]_{ri} = -\int_E \nabla m_r \cdot \boldsymbol{\varphi}_i \text{ dV} \tag{4.2.6}$$

$$[W_2]_{ri} = \int_{\partial E} (\boldsymbol{\varphi}_i \cdot \hat{\boldsymbol{n}}_E) m_r \text{ dS} \tag{4.2.7}$$

and $W = W_1 + W_2$. Since $\nabla m_r \in (\nabla \mathbb{P}_k^E)$, W_1 can be obtained immediately without computations. Namely, $\nabla m_r = \boldsymbol{g}_{r-1}$ so that recalling type *ii* DOFs in (3.1.3) and noting that ∇m_r is of order $k-1$ for RT-VEM and $k-2$ for BDM-VEM, we have that $[W]_{ri} = -|E| \text{DOF}_{r-1}(\boldsymbol{\varphi}_i) = -|E|$ if $(i = r-1)$ and 0 otherwise. Regarding matrix W_2, once again the term is computable recalling that DOFs of type i completely defines $(\boldsymbol{\varphi}_i \cdot \hat{\boldsymbol{n}}_E)$ on ∂E and m_r is known.

Matrix V

It is useful to store the values of $\nabla \cdot \boldsymbol{\varphi}_i \in \mathbb{P}_k(E)$. The i-th column of $V = H^{-1}W$ contains the coefficients of the polynomial decomposition of $\nabla \cdot \boldsymbol{\varphi}_i \in \mathbb{P}_{k-1}(E)$. For RT-VEM, $V \in \mathbb{R}^{n_{k,d} \times n_{dof}^E}$ and $V \in \mathbb{R}^{n_{k-1,d} \times n_{dof}}$ for BDM-VEM.

Matrix U

When a non-zero convection term is present $(\boldsymbol{\beta} \neq \{0,0\})$, we define $U \in \mathbb{R}^{dn_{k,d} \times n_{k,d}^E}$ as

$$[U]_{\alpha r} = \int_E (\boldsymbol{\beta} \cdot \boldsymbol{g}_\alpha) \, m_r \text{ dV}.s \tag{4.2.8}$$

Matrix B

$B \in \mathbb{R}^{dn^E_{k,d} \times n^E_{dof}}$ is crucial for the computation since it involves integrating 'virtual' shape functions, which is a priori not possible. Its definition is

$$[B]_{\alpha i} = \int_E \boldsymbol{g}_\alpha \cdot \boldsymbol{\varphi}_i \, \mathrm{dV}, \tag{4.2.9}$$

which can be split into $B = \begin{bmatrix} B^\nabla \\ B^\oplus \end{bmatrix}$ where

$$\begin{aligned} [B^\nabla]_{\beta i} &= \int_E \boldsymbol{g}^\nabla_\beta \cdot \boldsymbol{\varphi}_i \, \mathrm{dV}, \\ [B^\oplus]_{\gamma i} &= \int_E \boldsymbol{g}^\oplus_\gamma \cdot \boldsymbol{\varphi}_i \, \mathrm{dV}, \end{aligned} \tag{4.2.10}$$

$B^\nabla \in \mathbb{R}^{n^E_{k,\nabla} \times n^E_{dof}}$ and $B^\oplus \in \mathbb{R}^{n^E_{k,\oplus} \times n^E_{dof}}$. $B^\oplus$ can be computed exactly noting that its expression is precisely the DOFs of type *iii* in (3.1.3), so that $[B^\oplus]_{i\alpha} = |E|\mathrm{DOF}_\gamma(\boldsymbol{\varphi}_i) = |E|$ if i corresponds to γ and 0 otherwise. This means that in general, the components of $B^\oplus$, which are associated with DOFs of type *iii*, do not require any explicit calculations and are thus cheap computationally speaking.

On the other hand, B^∇ cannot be computed directly since $\boldsymbol{g}_\beta$ is a polynomial of degree k in $(\mathbb{P}_k(E))^d$ when $\hat{n}_{k-1,d}+1 \leq \beta \leq n_{k,d}$ for RT-VEM and $\hat{n}_{k-2,d}+1 \leq \beta \leq n_{k-1,d}$ for BDM-VEM. Using integration by parts

$$[B^\nabla]_{\beta i} = \int_E m_{\beta+1}(\nabla \cdot \boldsymbol{\varphi}_i) \, \mathrm{dV} + \int_{\partial E} m_{\beta+1}(\boldsymbol{\varphi}_i \cdot \hat{\boldsymbol{n}}_E) \, \mathrm{dS} := [B^\nabla_1]_{\beta i} + [B^\nabla_2]_{\beta i}. \tag{4.2.11}$$

where $\nabla m_{\beta+1} = \boldsymbol{g}^\nabla_\beta$. On one hand, B^∇_2 is directly computable from the DOFs of type i since it involves integration of a known polynomial over the faces of the element. On the other hand, B^∇_1 could be computable if the expression of $(\nabla \cdot \boldsymbol{\varphi})_i$ were known. But the polynomial decomposition of the divergence of a VEM basis function $\boldsymbol{\varphi}_i$ are exactly the columns of $V = H^{-1}W$, already computed. Finally, recalling also (4.2.4), the following relationship is obtained for all basis functions:

$$B^\nabla_1 = -H^\# V = -H^\#(H^{-1}W) = -H^\# H^{-1}(W_1 + W_2). \tag{4.2.12}$$

Matrix D

Since $(\mathbb{P}_k(E))^d \subset V^E_{k,h}$, it is possible to express the projector as seen from $V^E_{k,h} \to V^E_{k,h}$, instead of

$V_{k,h}^E \to (\mathbb{P}_k(E))^d$. For that purpose, matrix $D \in \mathbb{R}^{n_{dof}^E \times dn_{k,d}}$ is defined as

$$[D]_{i\alpha} = \mathrm{DOF}_i(\boldsymbol{g}_\alpha). \tag{4.2.13}$$

4.2.2 Computation of the projector

The L^2 projector $\boldsymbol{\Pi}_k^0 : V_{k,h}^E \to (\mathbb{P}_k(E))^d$ from (3.1.4) is defined as the solution of the following linear system:

$$\int_E \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h \cdot \boldsymbol{g} \ \mathrm{dV} = \int_E \boldsymbol{v}_h \cdot \boldsymbol{g} \ \mathrm{dV} \quad \forall \boldsymbol{g} \in (\mathbb{P}_k(E))^d, \quad \boldsymbol{v}_h \in V_{k,h}^E. \tag{4.2.14}$$

Specifically, for a basis function $\boldsymbol{\varphi}_i \in V_{k,h}^E$ it will be now shown how to compute this projection. First, $\boldsymbol{\Pi}_k^0(\boldsymbol{\varphi}_i)$ is expressed as a polynomial using the decompositions of $(\mathbb{P}_k(E))^d$ shown previously:

$$\Pi_k^0(\boldsymbol{\varphi}_i) = \sum_{\alpha=1}^{dn_{k,d}} t_\alpha^i \{\boldsymbol{g}_\alpha\} = \sum_{\beta=1}^{n_{k,\nabla}} t_\beta^i \{\boldsymbol{g}_\beta^\nabla\} \cup \sum_{\gamma=1}^{dn_{k,\oplus}} t_\gamma^i \{\boldsymbol{g}_\gamma^\oplus\} \tag{4.2.15}$$

where $\{t_\alpha^i\}_{\alpha=1,\dots,dn_{k,d}} = \{t_\beta^i\}_{\beta=1,\dots,n_{k,\nabla}} \cup \{t_\gamma^i\}_{\gamma=1,\dots,dn_{k,\oplus}}$ are the coefficients expressing the combination of the polynomial basis for the projection of $\boldsymbol{\varphi}_i$. Replacing in (4.2.14) and decomposing $\boldsymbol{g}$ the following linear system is obtained:

$$\begin{cases} \displaystyle\sum_{\beta=1}^{n_{k,\nabla}} t_\beta^i \int_E \boldsymbol{g}_\beta^\nabla \cdot \boldsymbol{g}_\delta^\nabla \ \mathrm{dV} + \sum_{\gamma=1}^{dn_{k,\oplus}} t_\gamma^i \int_E \boldsymbol{g}_\gamma^\oplus \cdot \boldsymbol{g}_\delta^\nabla \ \mathrm{dV} = \int_E \boldsymbol{\varphi}_i \cdot \boldsymbol{g}_\delta^\nabla \ \mathrm{dV} \\ \displaystyle\sum_{\beta=1}^{n_{k,\nabla}} t_\beta^i \int_E \boldsymbol{g}_\beta^\nabla \cdot \boldsymbol{g}_\lambda^\oplus \ \mathrm{dV} + \sum_{\gamma=1}^{dn_{k,\oplus}} t_\gamma^i \int_E \boldsymbol{g}_\gamma^\oplus \cdot \boldsymbol{g}_\lambda^\oplus \ \mathrm{dV} = \int_E \boldsymbol{\varphi}_i \cdot \boldsymbol{g}_\lambda^\oplus \ \mathrm{dV} \end{cases} \tag{4.2.16}$$

which, in view of (4.2.1) and (4.2.10), can be rewritten as

$$G t_\alpha^i = [B]_{\alpha i} \longrightarrow G = \begin{bmatrix} G^{\nabla\nabla} & G^{\nabla\oplus} \\ G^{\oplus\nabla} & G^{\oplus\oplus} \end{bmatrix} \begin{bmatrix} \{t_\beta^i\} \\ \{t_\gamma^i\} \end{bmatrix} = \begin{bmatrix} B^\nabla \\ B^\oplus \end{bmatrix} \tag{4.2.17}$$

so that $\{t^i\} = G^{-1}[B]_{\cdot i}$(column i of B). Collecting all the coefficients for $i = 1, \dots, n_{dof}^E$ we can define the projection matrix $\hat{\Pi}_k^0 \in \mathbb{R}^{dn_{k,d} \times n_{dof}^E}$ of the operator acting from $V_{k,h}^E$ to $(\mathbb{P}_k(E))^d$ as:

$$\hat{\Pi}_k^0 = \begin{bmatrix} t^1 & \dots & t^{n_{dof}^E} \end{bmatrix} \longrightarrow \hat{\Pi}_k^0 = G^{-1}B. \tag{4.2.18}$$

In order to obtain the matrix expression of the operator acting on $V_{k,h}^E$ into itself, we begin by expressing a polynomial $\boldsymbol{g}_\alpha$ as

$$\boldsymbol{g}_\alpha = \sum_{I=1}^{n_{dof}^E} \mathrm{DOF}_I(\boldsymbol{g}_\alpha)\boldsymbol{\varphi}_I,$$

so that replacing this equation in (4.2.14) yields

$$\Pi_k^0(\boldsymbol{\varphi}_i) = \sum_{\alpha=1}^{dn_{k,d}} t_\alpha^i \boldsymbol{g}_\alpha = \sum_{\alpha=1}^{dn_{k,d}} t_\alpha^i \left(\sum_{I=1}^{n_{dof}^E} \mathrm{DOF}_I(\boldsymbol{g}_\alpha)\boldsymbol{\varphi}_I \right) = \sum_{I=1}^{n_{dof}^E} \left(\sum_{\alpha=1}^{dn_{k,d}} t_\alpha^i \mathrm{DOF}_I(\boldsymbol{g}_\alpha) \right) \boldsymbol{\varphi}_I$$

which can be expressed in terms of D (4.2.13) computed earlier noting that $\mathrm{DOF}_I(\boldsymbol{g}_\alpha) = [D]_{I\alpha}$, giving $\Pi_k^0 = D\hat{\Pi}_k^0$. The flow of the local matrices required for computing the projector is clarified in Figure 4.2.1. A priori, all matrices on the Figure need to be computed to discretize the complete second order elliptic equation. However, the dashed lines represent matrices whose computation can be avoided if the respective terms are not present in the problem.

4.2.3 Stiffness matrices

We will now stablish the matrix implementation of the discrete equations (3.1.12).

Diffusion term: a_h^E

The terms in the bilinear form (3.1.8)

$$\begin{aligned} a_h^E(\boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j) &= (\nu \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_i, \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_j)_E + S^E(\boldsymbol{\varphi}_i - \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_i, \boldsymbol{\varphi}_j - \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_j) \\ &:= (K_c^a)_{[ij]} + (K_s^a)_{[ij]} \end{aligned}$$

In terms of the already computed matrices, we have

$$\begin{aligned} K_c^a &= \left[\hat{\Pi}_k^0\right]^\intercal G^\nu \hat{\Pi}_k^0 \\ K_s^a &= \overline{\nu}|E|(\mathcal{I} - \boldsymbol{\Pi}_k^0)^\intercal (\mathcal{I} - \boldsymbol{\Pi}_k^0) \end{aligned}$$

where $\mathcal{I}$ is the $n_{dof}^E \times n_{dof}^E$ identity matrix.

Divergence term: $(P_h, \nabla \cdot \boldsymbol{v}_h)_{2,\Omega}$

This term does not use the projector and it has already been computed as W.

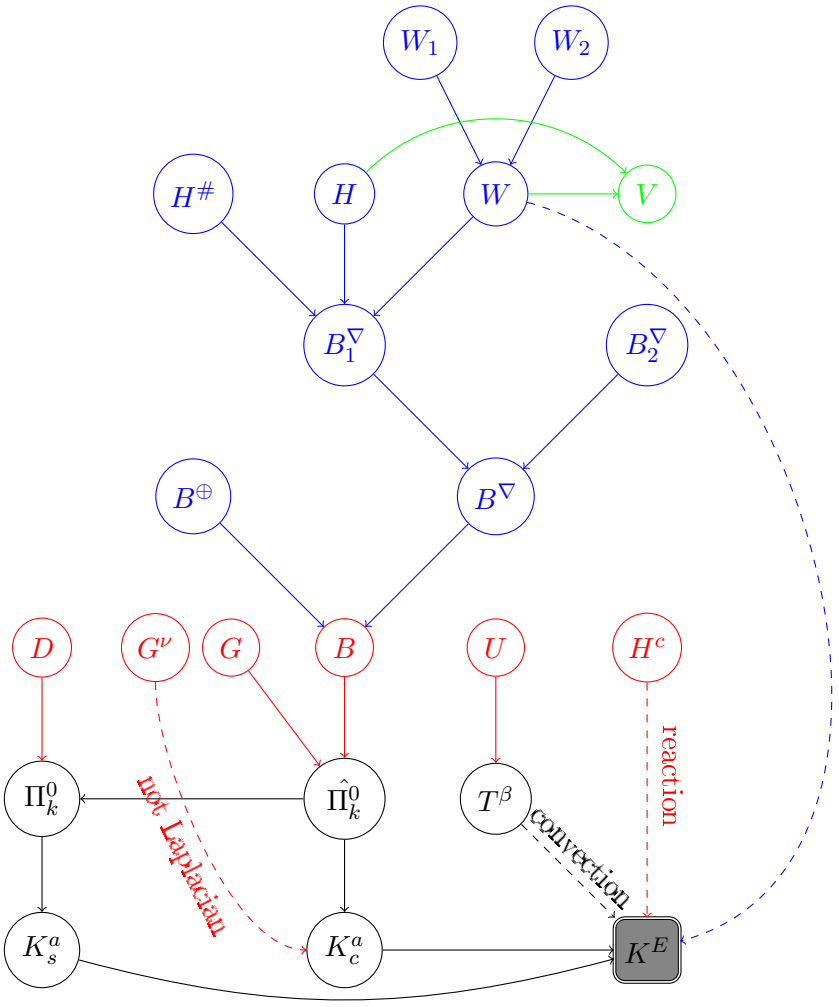


FIGURE 4.2.1: Flow chart of matrix computations

Convection term: $(\boldsymbol{\beta} \cdot \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h, P_h)_{2,\Omega}$

In the case that $\boldsymbol{\beta} \neq (0,0)$, we approximate $(\boldsymbol{\beta} \cdot \boldsymbol{v}_h, P_h)_{2,\Omega} \approx (\boldsymbol{\beta} \cdot \boldsymbol{\Pi}_k^0 \boldsymbol{v}_h, P_h)_{2,\Omega}$, so that

$$[T]_{ir}^{\beta} = \int_E \left(\boldsymbol{\beta} \cdot \Pi_k^0 \boldsymbol{\varphi}_i\right) m_r \, \mathrm{dV} = \sum_{\alpha=1}^{dn_{k,d}} t_\alpha^i \int_E (\boldsymbol{\beta} \cdot \boldsymbol{g}_\alpha) \, m_r \, \mathrm{dV}, \tag{4.2.19}$$

which collecting all basis functions and expressing it in matrix form using (4.2.8) becomes

$$T^{\beta} = \left[\hat{\Pi}_k^0\right]^{\intercal} U. \tag{4.2.20}$$

Reaction term: $(c \; P_h, Q_h)_{2,\Omega}$

This term is already computed as H^c.

Final stiffness matrix

Finally, the local stiffness matrix K^E on an element E is given by:

$$K^E := \begin{bmatrix} K_c^a + K_s^a & -W^\intercal - T^\beta \\ W & H^c \end{bmatrix}, \tag{4.2.21}$$

with size $H \in \mathbb{R}^{(n_{dof}^E + n_{k,d}) \times (n_{dof}^E + n_{k,d})}$ for RT-VE and $H \in \mathbb{R}^{(n_{dof}^E + n_{k-1,d}) \times (n_{dof}^E + n_{k-1,d})}$ for BDM-VEM. Note that for a pure Darcy flow problem, $T^\beta = H^c = \mathbf{0}$

4.3 Imposing conditions on the Degrees of Freedom

4.3.1 DOFs definition in the presence of fractures and traces

For 3D elements that are intersected by planar fractures, a jump will appear in the flux between former adjacent faces, since now some of the flux leaving one face may enter the 2D element as a source term. In order to capture this phenomenon, DOFs of type i associated with a face on a fracture must be doubled, so that this jump can be represented. Similarly, in the case of intersecting fractures whose intersection defines a trace, an edge of the trace will have 4 times as many DOFs. The reason for this is that each of the 2 fractures defining the trace will have its DOFs doubled. For examples clarifying this point see Figures 4.3.1 and 4.3.2, where RT-VEM of order 0 elements were chosen for simplicity, as they have only one DOF of type i per face/edge, although conceptually there is no difference for any order. In the first case, the fracture results in a duplication of the type i DOFs on the face coinciding with the fracture (blue DOFs 1 and 2 in the Figure). In the second case, a trace segment doubles the flux DOFs on each fracture, so that 4 flux DOFs are present in the end (DOFs 1 to 4).

4.3.2 3D systems

Let $\boldsymbol{u}^3 = \left\{\boldsymbol{u}_1^3, ..., \boldsymbol{u}_{U_{dof_{3D}}}^3\right\}$, $Q^3 = \left\{Q_1^3, ..., Q_{P_{dof_{3D}}}^3\right\}$ be the column vectors of flux DOFs and pressure DOFS, collected in $h^{3D} := (\boldsymbol{u}^3, Q)$ and recalling $f^3 \in \mathbb{R}^{ndof_{3D}}$ as the load values of the 3D domain (including terms arising from non-homogeneous boundary conditions). In Ω_3, the stiffness

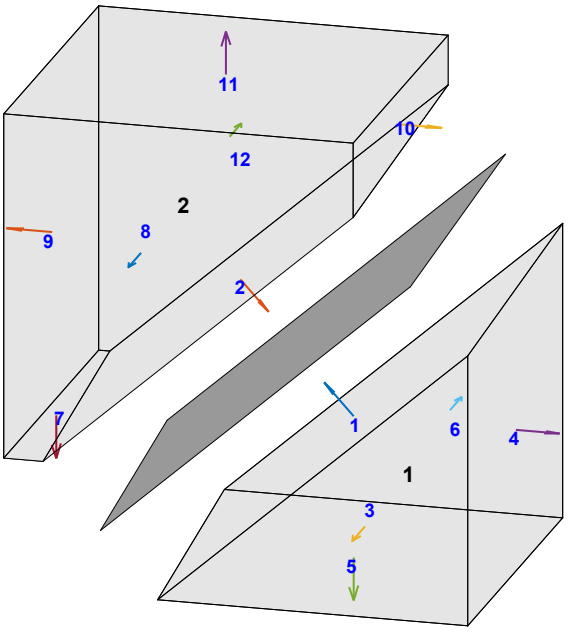


FIGURE 4.3.1: Duplication of DOFs in faces in the presence of a fracture

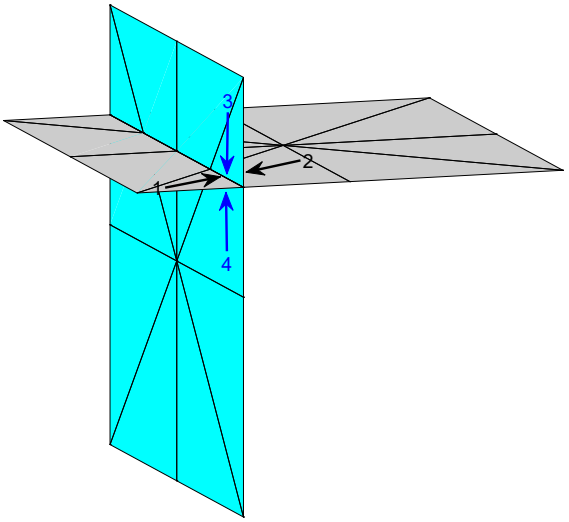


FIGURE 4.3.2: Duplication of DOFs on fracture edges in the presence of traces

matrix after assembly of the local matrices is

$$K^3 := \left(\begin{array}{c|c} A^{3D} & -\left(D^{3D}\right)^{\mathsf{T}} \\ \hline D^{3D} & \mathbf{0} \end{array}\right) \begin{matrix} \} \boldsymbol{u}^3 \\ \} Q^3 \end{matrix} \qquad (4.3.1)$$

$$\underbrace{\phantom{A^{3D}}}_{\boldsymbol{u}^3} \quad \underbrace{\phantom{-\left(D^{3D}\right)^{\mathsf{T}}}}_{Q^3}$$

where A^{3D} and D^{3D} are the matrices arising from the bilinear forms. Namely $[A^{3D}]_{ij} = (\nu \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_i, \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_j)_{2,\Omega_3}$ and $[D^{3D}]_{pj} = (P_p, \nabla \cdot \boldsymbol{\varphi}_j)_{2,\Omega_3}$, with $i,j = 1, ..., U_{dof_{3D}}$ and $p = U_{dof_{3D}} + 1, ..., , U_{dof_{3D}} + P_{dof_{3D}}$ (arising from (2.1.5), see Section 4).

4.3.3 DFN system

For every fracture F_r, with $r = 1, ..., N_f$, we call U_{dof_r} the number of DOFs for the velocity space and P_{dof_r} the number of DOFs for the pressure space of fracture F_r. We define $ndof_r := U_{dof_r} + P_{dof_r}$ as the total number of DOFs from the flux and the pressure variable, so that the stiffness matrix $K^r \in \mathbb{R}^{ndof_r \times ndof_r}$ for each fractures can be obtained. Recalling (4.2.21) and after assembling the elementary stiffness matrices following standard FEM procedures, the structure of the stiffness matrix for a fracture F_r for the pure Darcy problem is:

$$K^r := \left(\begin{array}{c|c} A^r & -(D^r)^\intercal \\ \hline D^r & \mathbf{0} \end{array}\right) \begin{array}{l} \left.\vphantom{\begin{array}{c} A^r \\ D^r \end{array}}\right\} \boldsymbol{u}^{2,r} \\ \left.\vphantom{\begin{array}{c} A^r \\ D^r \end{array}}\right\} Q^{2,r} \end{array} \qquad (4.3.2)$$

$\boldsymbol{u}^{2,r}$ $\quad$ $Q^{2,r}$

where A and D are the matrices arising from the bilinear forms. Namely $[A^r]_{ij} = (\nu^{2,r} \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_i, \boldsymbol{\Pi}_k^0 \boldsymbol{\varphi}_j)_{2,F_r}$, and $[D^r]_{sj} = (Q_s, \nabla \cdot \boldsymbol{\varphi}_j)_{2,F_r}$, with $i,j = 1, ..., U_{dof_r}$ and $s = U_{dof_r} + 1, ..., , U_{dof_r} + P_{dof_r}$ (see (2.1.5) and Chapter 2). The column vectors $\boldsymbol{u}^{2,r} = \left\{\boldsymbol{u}_1^{2,r}, ..., \boldsymbol{u}_{U_{dof_r}}^{2,r}\right\}$, $Q^{2,r} = \left\{Q_1^{2,r}, ..., Q_{P_{dof_r}}^{2,r}\right\}$ and $f_r \in \mathbb{R}^{ndof_r}$ are the vectors of flux DOFs, pressure DOFS and load values (including terms arising from non-homogeneous boundary conditions) respectively. They are collected in $h_r := (\boldsymbol{u}_r, Q_r)$ as the vector of values of the DOFs of the complete discrete solution on fracture F_r. We note that the matrix K^r is singular for fractures with pure Neumann boundary conditions whenever $\gamma = 0$ (no reaction term). For the complete DFN we have:

$$K^{DFN} = \begin{pmatrix} K^1 & 0 & \cdots & 0 \\ 0 & K^2 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \cdots & K^{N_f} \end{pmatrix}, \; f^{DFN} = \begin{pmatrix} f_1 \\ \vdots \\ \vdots \\ f_{N_f} \end{pmatrix} \text{ and } h^{DFN} = \begin{pmatrix} h_1 \\ \vdots \\ \vdots \\ h_{N_f} \end{pmatrix},$$

and the final linear system is

$$K^{DFN} h^{DFN} = f^{DFN} \tag{4.3.3}$$

At this point there is no connection between the discrete problems at each fracture. As explained in Section 2.2, there are constraints that arise from the coupling between domains and must be satisfied for the problem to be physically consistent. Namely, these are the continuity of pressure (or its corresponding jump for non-infinite 2D/3D permeability) and conservation of mass. In the case of a primal discretization, only the continuity of pressure is imposed (see [15]), since the balance of mass is not necessarily preserved for C^0 elements. For the mixed case, continuity of pressure is practically never satisfied since the discrete spaces for the pressure are not globally continuous, but only in L^2. However, since the discretization of the flux variable is globally in H(div), flux balance is perfectly preserved.

The linear system for the complete DFN is obtained by imposing matching conditions for the DOF on the traces that guarantee flux/mass balance on the traces. In the primal formulation [15] Lagrange multipliers are introduced to enforce equality between DOFs on nodes on the same trace but on different fractures that occupy the same point in space. On the other hand, 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 trace, *i.e.* $\boldsymbol{u}_h \cdot \hat{\boldsymbol{n}}_T^{2,r} \in \mathbb{P}_k(e)$ where $\hat{\boldsymbol{n}}_{T_t}^{2,r}$ is the normal of the edge $e \subset T$ on the plane of the fracture F_r. 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) and assuming for now that the indexed by $t = 1, ..., \ldots, N_{et}$(total number of edges on traces). We have that for each t we can define the only two indices $s(t), j(t)$ such that $e_t \in F_{s(t)}$ and $e_t \in F_{j(t)}$. In other words, only two the fractures whose intersetion includes the edge (e_t). The edge e_t will have $2(k+1)$ DOF on fracture $F_{s(t)}$ and the same for fracture $F_{j(t)}$, because edges on trace have twice as many DOF as internal edges (as explained in 4.3.1). We will call the former $D_{t,s}$ and the latter $D_{t,j}$, and $\#D_{t,s} = \#D_{t,s} = 2(k+1)$. These set of DOF can be split, 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 $D_{t,s} = L_{t,s} \cup R_{t,s}$ (analogously for $D_{t,j}$), which collect the DOF to the left and to the right of the trace respectively. We have that $\#L_{t,s} = \#R_{t,s} = k+1$. Therefore, for each edge we will establish $k+1$ conditions that will link

the pointwise values of the DOF from $F_{s(t)}$ and $F_{j(t)}$ using Lagrange multipliers. Then, for each t, we define the row vectors $\mathfrak{T}_{t_q} \in \mathbb{R}^{\sum_r ndof_r}$ as:

$$\mathfrak{T}_{t_q} := \begin{pmatrix} & & L_{t,s}(q) & & R_{t,s}(q) & & L_{t,j}(q) & & R_{t,j}(q) & \\ 0 & \cdots & \xi_{s,j} & \cdots & \xi_{s,j} & \cdots & \xi_{s,j} & \cdots & \xi_{s,j} & \cdots & 0 \end{pmatrix},$$

with $q = 1, ..., k+1$, and $L_{t,s}(q)$ is the q^{th} DOF on edge e_t in fracture $F_{s(t)}$ located to the left of the trace (analogously for the others). The numbers $\xi_{s,j} > 0$ represent a measure of permeability in the flux exchange between fractures F_s and F_j but since there is no inclusion of 1D elements in the model, they are all equal to 1 (*i.e.* perfect flux balance). The sets of DOF are numbered in such a way that for the same q, $L_{t,s}(q)$, $R_{t,s}(q)$, $L_{t,j}(q)$ and $R_{t,j}(q)$ all represent pointwise values on the same point on the trace (albeit on two different fractures). To clarify concepts beyond notation, there will be a new equation and a Lagrange multiplier for each pointwise flux value, imposing that the sum of all 4 flux contributions on that point be zero. In the weak form, it is like taking dirac-delta functions on the weak form for strong pointwise equality.

Finally, we set $\mathfrak{T} \in \mathbb{R}^{N_t(k+1)\times\sum_r ndof_r}$ as the matrix:

$$\mathfrak{T}^{DFN} = \begin{pmatrix} \mathfrak{T}_{1_1} \\ \vdots \\ \mathfrak{T}_{1_{k+1}} \\ \mathfrak{T}_{2_1} \\ \vdots \\ \mathfrak{T}_{N_{t\,k+1}} \end{pmatrix},$$

and thus the final linear system for a pure DFN problem is:

$$\begin{bmatrix} K^{DFN} & \mathfrak{T}^{DFN^T} \\ \mathfrak{T}^{DFN} & 0 \end{bmatrix} \begin{bmatrix} h^{DFN} \\ \lambda^{DFN} \end{bmatrix} = \begin{bmatrix} f^{DFN} \\ 0 \end{bmatrix}. \tag{4.3.4}$$

The solution for system with this structure falls into the study of domain decomposition methods for mixed finite elements. We refer the reader to [6, 8].

4.3.4 Hybrid systems

The procedures described in the previous sections for introducing coupling conditions on pure DFN problems and between 2D and 3D elements can be readily combined into a single global problem. Furthermore, the coupling conditions between fractures act on edge DOFs of the flux variables while 2D-3D coupling stablishes conditions between flux DOFs on faces with internal 2D DOFs. So that, in fact, the coupling conditions are 'decoupled' in the final global system. The complete system with coupling conditions between 2D and 3D elements as well as between fractures is obtained:

$$\begin{bmatrix} K^{3D} + C^{3D/3D} & C^{3D/2D\intercal} & 0 \\ C^{3D/2D} & K^{DFN} & \mathfrak{T}^{DFN\intercal} \\ 0 & \mathfrak{T}^{DFN} & 0 \end{bmatrix} \begin{bmatrix} h^{3D} \\ h^{DFN} \\ \lambda^{DFN} \end{bmatrix} = \begin{bmatrix} f^{3D} \\ f^{DFN} \\ 0 \end{bmatrix},$$

where the $C^{3D/3D}$, $C^{3D/2D}$ and $\mathfrak{T}^{DFN}$ are the coupling matrices (see (2.2.3)). Specifically,

$$[C^{3D/3D}]_{ij} = \sum_{r=1}^{N_f} (\nu^r \boldsymbol{\varphi}_i \cdot \hat{\boldsymbol{n}}^{3,r}, \boldsymbol{\varphi}_j \cdot \hat{\boldsymbol{n}}^{3,r})_{2,F_r} \qquad [C^{3D/2D}]_{sj} = \sum_{r=1}^{N_f} ([\![Q_s, \boldsymbol{\varphi}_j \cdot \hat{\boldsymbol{n}}^{3,r}]\!])_{2,F_r}$$

As an example, we present the final matrix for the linear system associated with a problem comprised of a 3D matrix and a 2 fracture DFN, whose geometry is given in Figure 4.3.3. It is discretized by RT1-VEM elements. In Figure 4.3.4 the blue dots indicate non-null components of the matrix, and they are associated with the 3D and 2D DOFs, for the pressure and the flux variable. There are 912 3D flux DOFs, 256 3D pressure DOFs, 216 2D flux DOFs, 96 2D pressure DOFs and 8 DOFs for the lagrange multipliers at traces.

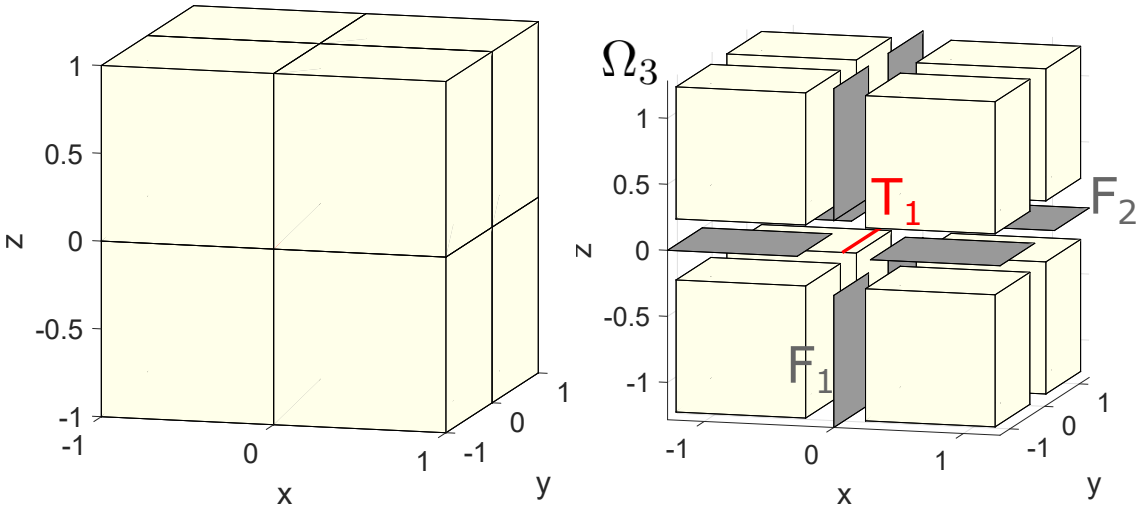


FIGURE 4.3.3: A cube and 2 fractures (left). Exploded mesh for visualization purposes (right)

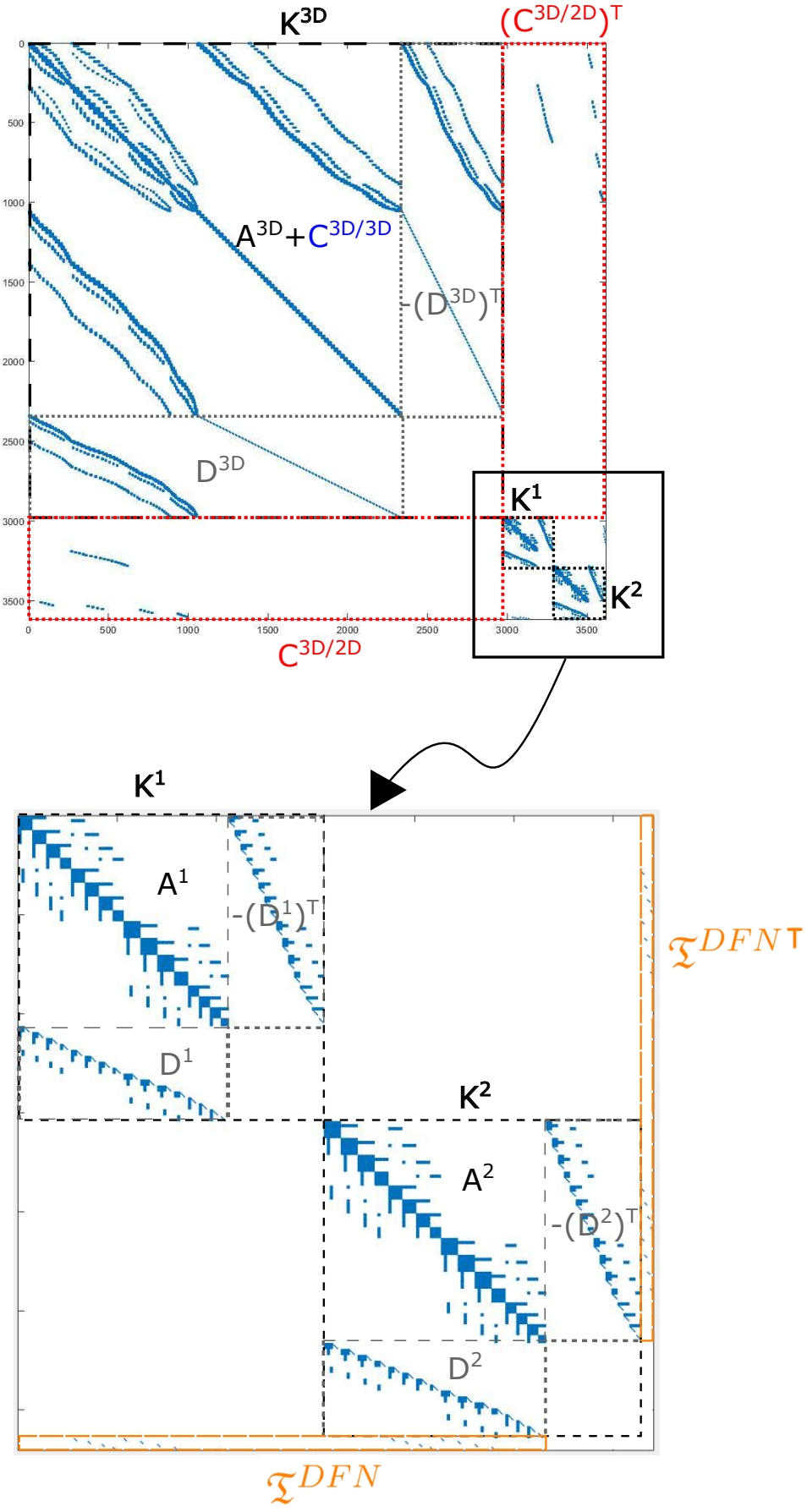


FIGURE 4.3.4: Linear system for a hybrid problem with a 2 fracture DFN

Chapter 5

Numerical results

This section contains the most relevant contribution of this thesis, where a thorough numerical treatment of the proposed methodologies is presented. Every result and graphic put forward here has been made with an in-house code in the Matlab programming language. Firstly, a set of pure 3D problems is proposed. Then, pure DFN problems are analised. Finally, full problems combining 2D-3D interaction as well as networks of fractures are studied. All problems will be considered to have been previously *nondimensionalized*, meaning that the corresponding input data had their physical units removed and units have been adjusted by a suitable substitution of variables to simplify and parameterize problems.

5.1 3D problems

In order to test the correctness of the method, a set of 'Patch-test' and benchmark problems are proposed, were the error in the discrete solutions is compared with the known exact solutions. For the error in the flux variable $\boldsymbol{u}_h$, since it is not explicitly known inside an element, the projection $\Pi_k^0 \boldsymbol{u}_h$ is used.

5.1.1 Patch test

By the very definition of the VEM space, it is guaranteed that, for a discretization order k, $\mathcal{P}_k(E) \subset \mathcal{V}_k(E)$. In other words, any polynomial in three coordinates (x,y,z) on an element E of degree at most k is included in the discrete VEM space on any element $E \in \mathcal{T}_h$. The purpose of this subsection

Element	$\#\mathcal{Q}_k(E)$	Solution					
		5	$x-y+z$	z^2+xy	x^3-xyz	$x^2z^2-y^4$	xy^2z^2
RT0	1	$\sim\epsilon$	2.8284	3.3810	1.1597	4.8280	0.2365
BDM1	1	$\sim\epsilon$	2.8284	1.2649	1.1996	1.0210	0.3263
RT1	4	$\sim\epsilon$	$\sim\epsilon$	1.2649	0.7738	1.0210	0.3731
BDM2	4	$\sim\epsilon$	$\sim\epsilon$	1.2649	0.6922	0.9518	0.2716
RT2	10	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.6922	0.6168	0.2716
BDM3	10	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.6922	0.3311	0.4286
RT3	20	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.3311	0.5446
BDM4	20	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.3311	0.1452
RT4	35	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.1452
BDM5	35	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.1452
RT5	56	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$

TABLE 5.1.1: $||P-P_h||_{\mathrm{L}^2}$ for patch test problems

Element	$\#\mathcal{V}_k(E)$	Solution					
		5	$x-y+z$	z^2+xy	x^3-xyz	$x^2z^2-y^4$	xy^2z^2
RT0	6	$\sim\epsilon$	$\sim\epsilon$	3.2660	0.9445	1.3887	0.7285
BDM1	21	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.9428	1.3064	0.8432
RT1	24	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.9428	1.3064	0.8432
BDM2	50	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	1.1383	0.5622
RT2	56	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	1.1383	0.5622
BDM3	95	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.8036
RT3	105	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	0.8036
BDM4	159	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
RT4	174	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
BDM5	266	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
RT5	266	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$

TABLE 5.1.2: $||\boldsymbol{u}-\Pi_k^0\boldsymbol{u}_h||_{\mathrm{L}^2}$ for patch test problems

is therefore to show that when the solution of the problem is a polynomial of degree at most k, *i.e.* 'patch test', the method captures the exact solution within numerical precision. For all cases, problem data will be $\boldsymbol{a}=\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}$, $\boldsymbol{b}=(0,0,0)$ and $\boldsymbol{c}=0$, while the mesh will be comprised of a single hexahedral element over the domain $\Omega=[-1,1]^3$, as seen in Figure 5.1.1. The symbol $\sim\epsilon$ will be used to denote a number that is zero within numerical precision. Results for errors in p, $\Pi_0\boldsymbol{u}$ and $\mathrm{div}(\boldsymbol{u})$ are presented in Tables 5.1.1 to 5.1.3.

Element	$\#\mathcal{Q}_k(E)$	Solution					
		5	$x-y+z$	z^2+xy	x^3-xyz	$x^2z^2-y^4$	xy^2z^2
RT0	1	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	9.7980	10.1988	2.4347
BDM1	1	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	9.7980	10.3966	2.5762
RT1	4	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	10.3966	1.3771
BDM2	4	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	10.3966	1.3771
RT2	10	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	1.3771
BDM3	10	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	1.3771
RT3	20	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
BDM4	20	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
RT4	35	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
BDM5	35	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$
RT5	56	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$	$\sim\epsilon$

TABLE 5.1.3: $||\mathrm{div}(\boldsymbol{u}) - \mathrm{div}(\boldsymbol{u}_h)||_{\mathrm{L}^2}$ for patch test problems

5.1.2 Convergence tests

A verification of the expected convergence rates is presented in the following. A series of regular hexagonal meshes is used (Figure 5.1.1) in order to prevent numerical error stemming from computing rotations of faces, and mesh parameter h represents the length of the diagonal of an element. Since the mesh is regular and the data is constant, it is only necessary to compute the stiffness matrix once since it will be the same for any element. This saves an enormous amount of computation time and thus it is possible to obtain results for element of high interpolation order ($k \geq 3$).

5.1.3 Polynomial solution

For this problem, the solution is given by: $P(x,y,z) = z^6$ with data

$$\boldsymbol{a} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \boldsymbol{b} = (0,0,0)\,, \quad \boldsymbol{c} = 0$$

so that the load term is $f = -30z^4$. Convergence rates are given in Figure 5.1.2 by RT-VEM and BDM-VEM for orders 0 to 5 and 1 to 5 respectively. It can be seen that in all cases that the expected convergence rates are obtained.

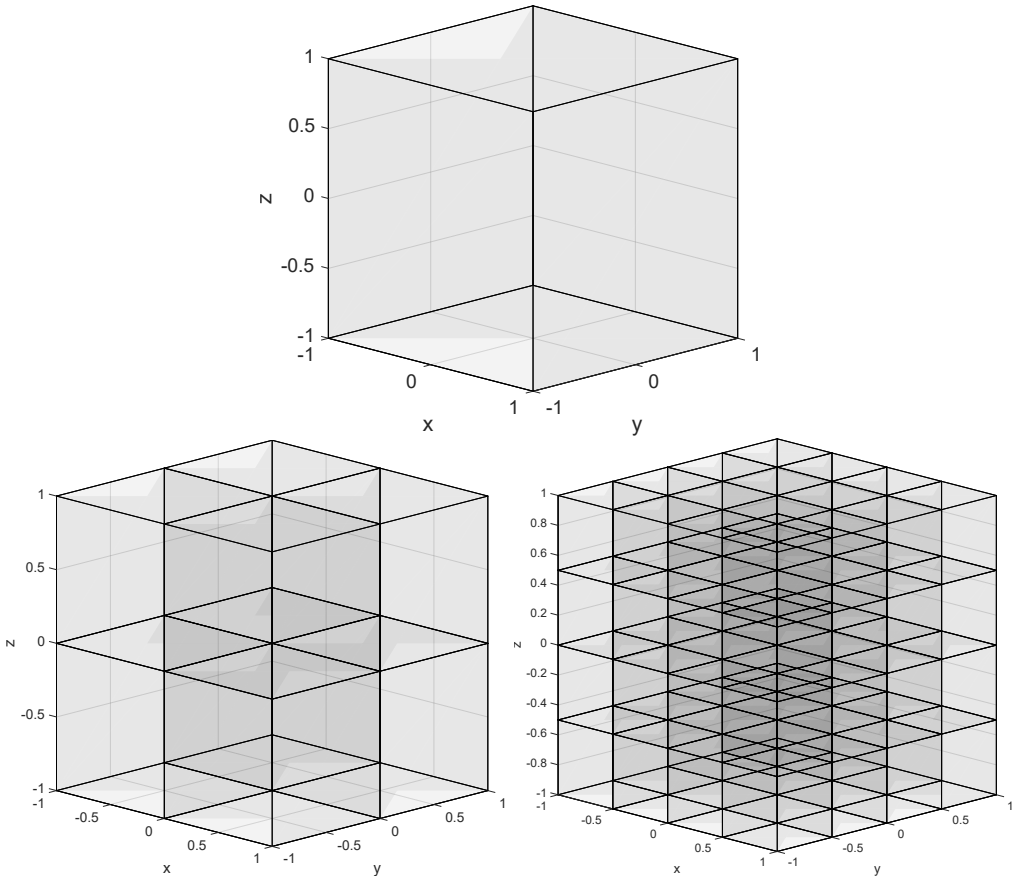


FIGURE 5.1.1: Regular hexahedral meshes

5.1.4 A diffusion-convection problem with non constant coefficients

This problem involves a smooth non-polynomial solution and non constant coefficients over $\Omega = [-1,1]^3$, which is very common problem in subsurface flow. Data for this problem is:

$$P(x,y,z) = x^2yz + sin(\pi x)sin(\pi y)sin(\pi z) + 2, \tag{5.1.1}$$

$$\boldsymbol{a}(x,y,z) = \begin{pmatrix} 1+z^2 & -xy & xz \\ -xy & 1+x^2 & -yz \\ xz & -yz & 1+y^2 \end{pmatrix} \tag{5.1.2}$$

$$\boldsymbol{b}(x,y,z) = (x,y,z) \tag{5.1.3}$$

$$\boldsymbol{c}(x,y,z) = 0. \tag{5.1.4}$$

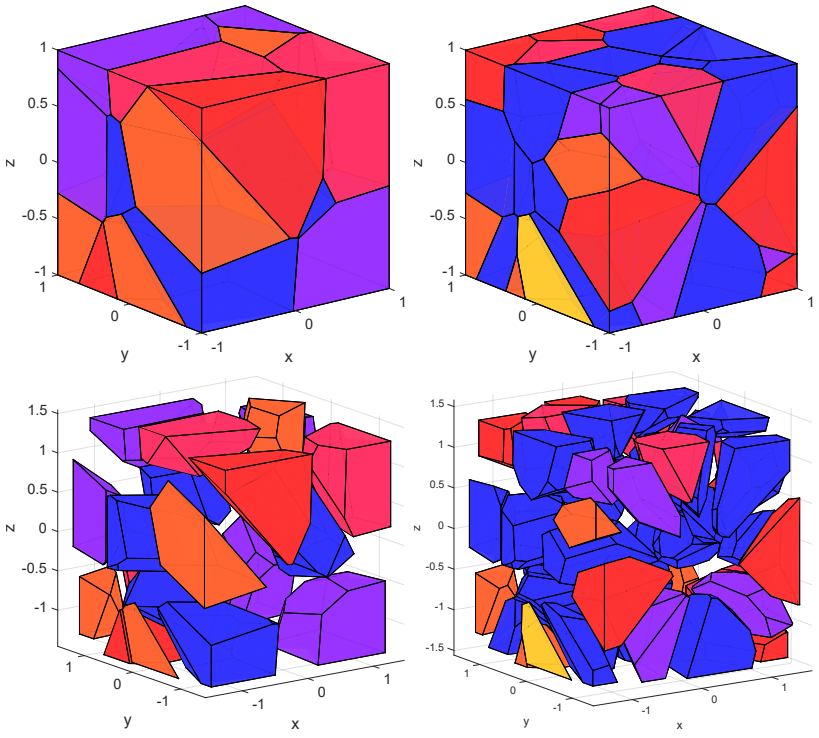


FIGURE 5.1.3: Voronoi meshes of 20 and 50 elements

The loading term is obtained by replacing the data in (2.1.1). Random Voronoi meshes (Figure 5.1.3) of increasing number of elements are used to verify convergence rates for RT-VEM and BDM-VEM of orders 0 to 4 and 1 to 4 respectively, where color indicates number of faces (blue represents $n_f \geq 10$.). These are given Figure 5.1.4. It can be seen that a finer mesh is require to obtain the optimal rates of asymptotic convergence, which is expected for several reasons: randomly generated meshes mean that the mesh parameter h, which is dominated by one single element with the largest diameter, may not decrease uniformly with element number and due to this a refined mesh may be coarser than its predecessor mesh in some areas. Furthermore, non constant coefficients make the load term more complex, which requires higher accuracy when approximating it by numerical integration. In addition, since the coefficients are non constant and the mesh is not uniform, it becomes mandatory to fully compute the stiffness matrix for each single element, in contrast with the previous example. Finally, 3D computations are always much more expensive than 2D ones. For these reasons and the fact that higher orders are rarely used in actual applications, convergence is only shown for elements with $k \leq 2$ (Figure 5.1.4). Results show convergence rates approximating the expected values with some oscillations, with some elements underperforming and others showing superconvergence. Finer meshes are likely to produce a more stable convergence rate, more similar to the expected theoretical value.

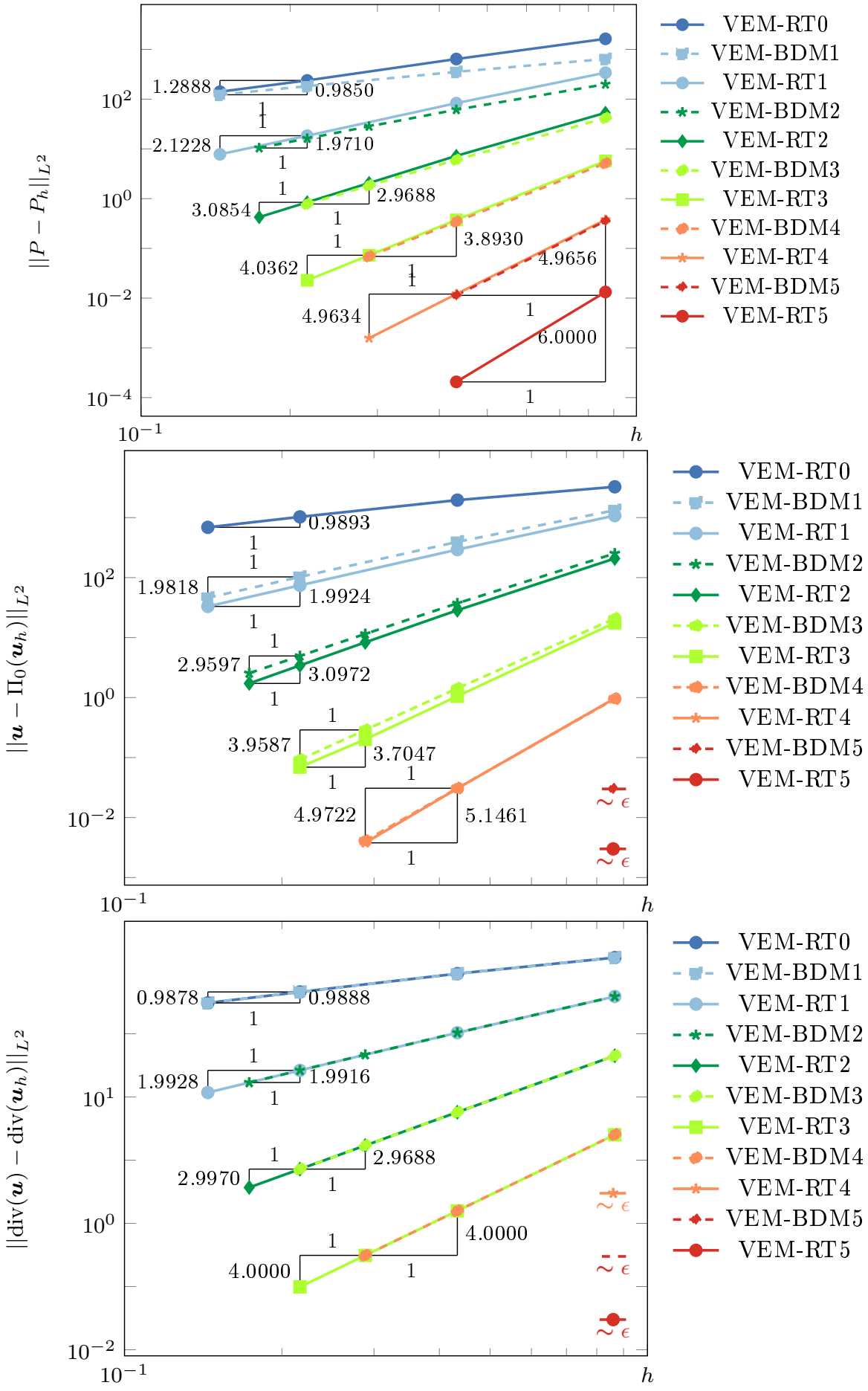


FIGURE 5.1.2: Convergence rates for a problem with polynomial solution

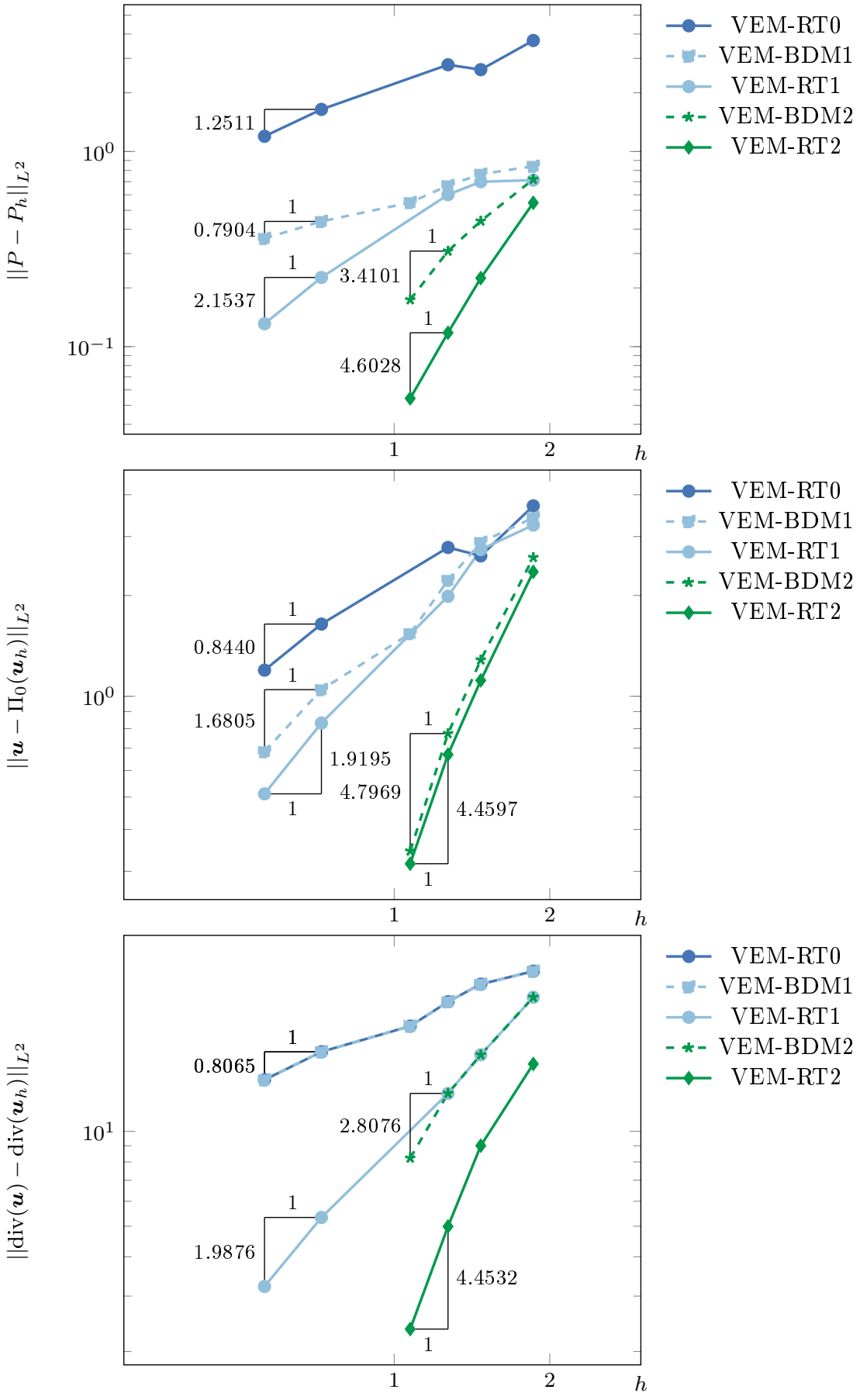


FIGURE 5.1.4: Convergence rates for a problem with non-polynomial smooth solution and variable coefficients

5.2 Pure DFN problems

In this section we consider problems in which only the planar fracture network is taken into account. Only a brief presentation is provided. For a more complete treatment of this setting refer to [16], were 2D convergence rates for benchmark problems were obtained.

5.2.1 DFN6: Exact solution

This problem consists of a 6 fracture network, where an incoming unitary flow per unit length is applied on the left side fracture 1 and a null pressure on the right side of fracture 5. Due to the geometry, the problem has an exact solution which is easily obtainable considering that the DFN can be thought of as a series of 1 dimensional flow pipes. Therefore, it is expected that the solution will be piecewise linear within each fracture with constant velocity field, so that the problem can be solved exactly with VEM elements which are at least linear in the pressure and constant in the velocity (*e.g.* BDM1-VEM,RT1-VEM). Figure 5.2.1 shows the geometry and mesh of the problem with the pressure head solution, while 5.2.2 shows the obtained velocity field and boundary conditions. Since the network is small, it is useful to provide a graphic showing the path that the flux takes through the network (Figure 5.2.3) where each arrow represents the exchange between fractures. As mentioned before, the mixed formulation has the advantage of strongly imposing perfect mass conservation, and it can be seen that the total incoming and outgoing flux is the same since there are no sinks nor sources inside the network.

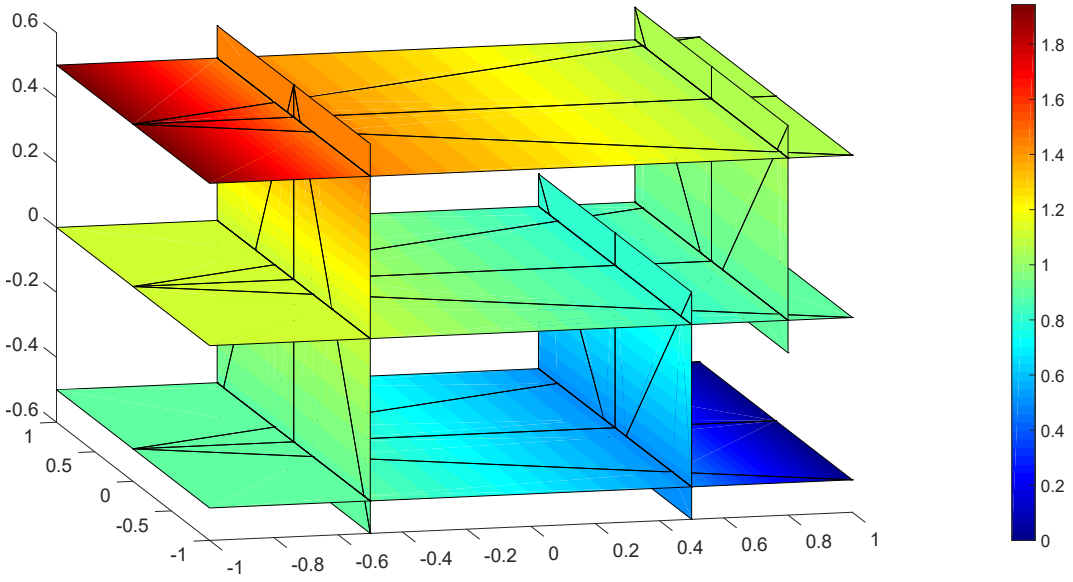


FIGURE 5.2.1: Geometry and pressure head solution of the problem DFN6

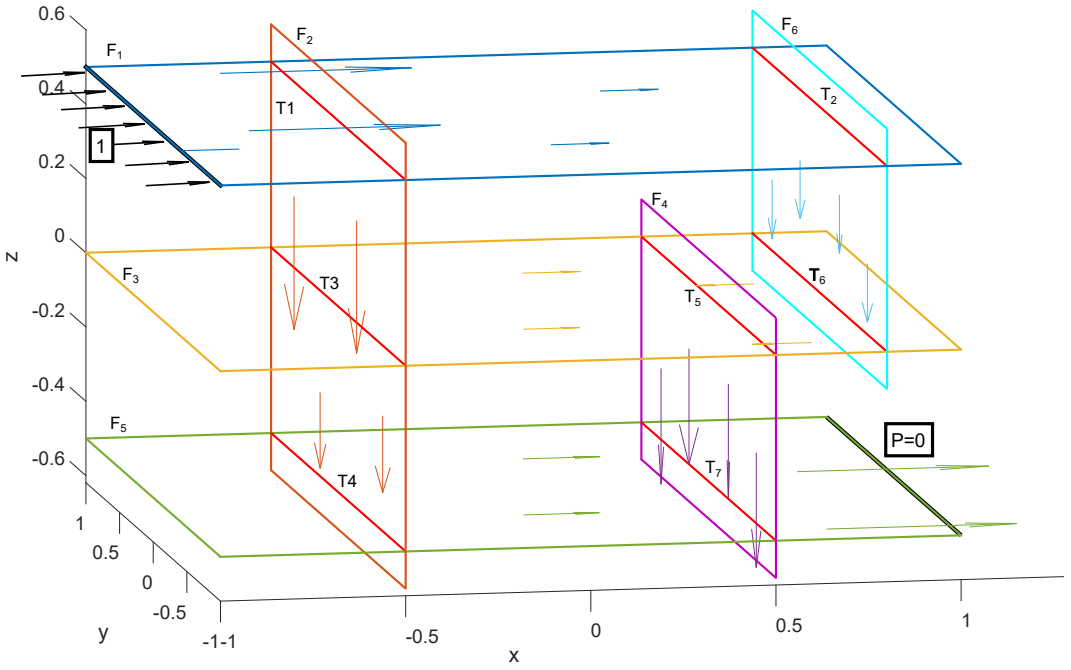


FIGURE 5.2.2: Global velocity field

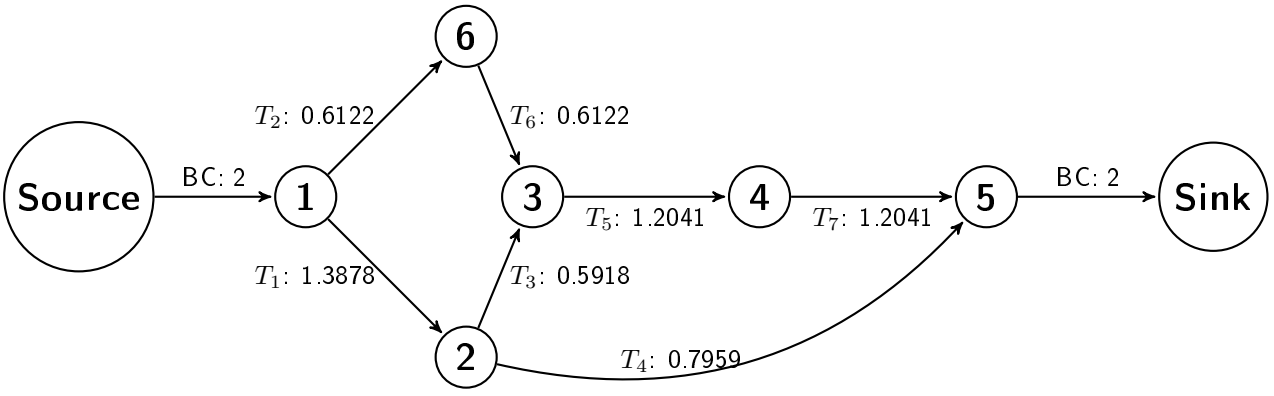


FIGURE 5.2.3: Flux path on a 6 fracture DFN

5.2.2 Complex DFN

This 36-fracture network contains one source fracture where a net incoming flux of 1000 is applied. All other boundaries are considered isolated except for a single sink border, where pressure is set to 0. The diffusion coefficient is set to 1. The intersection between fractures gives rise to 65 traces, and remembering that the discretization is globally in H(div), mass conservation is satisfied within numerical accuracy. The discrete solution of the pressure head obtained with order 1 RT-VEM is presented in Figure 5.2.4 and in Figure 5.2.5 for the velocity field. Table 5.2.1 shows the flux exchange for the whole network.

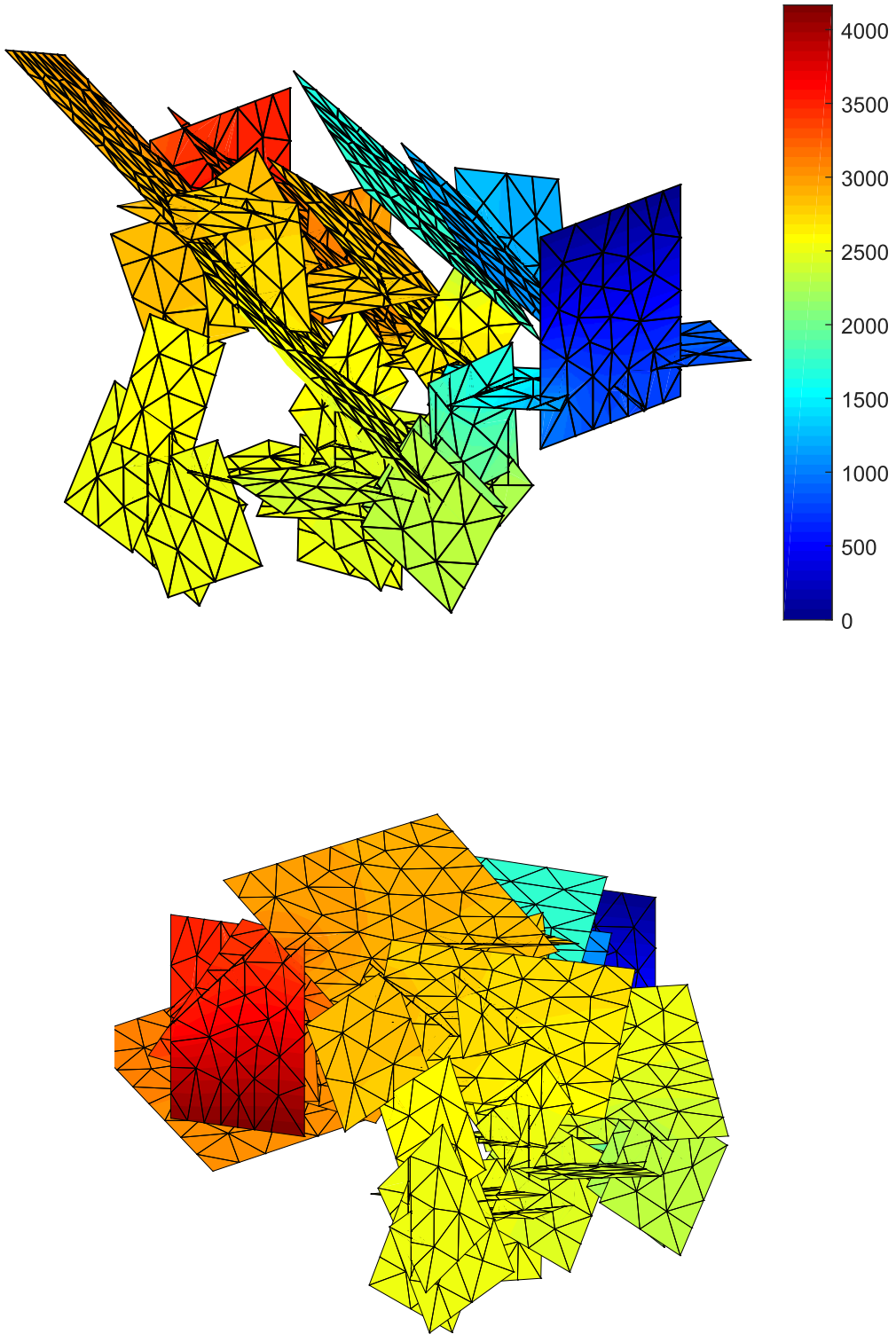


FIGURE 5.2.4: Pressure head of a pure DFN problem with 36 fractures

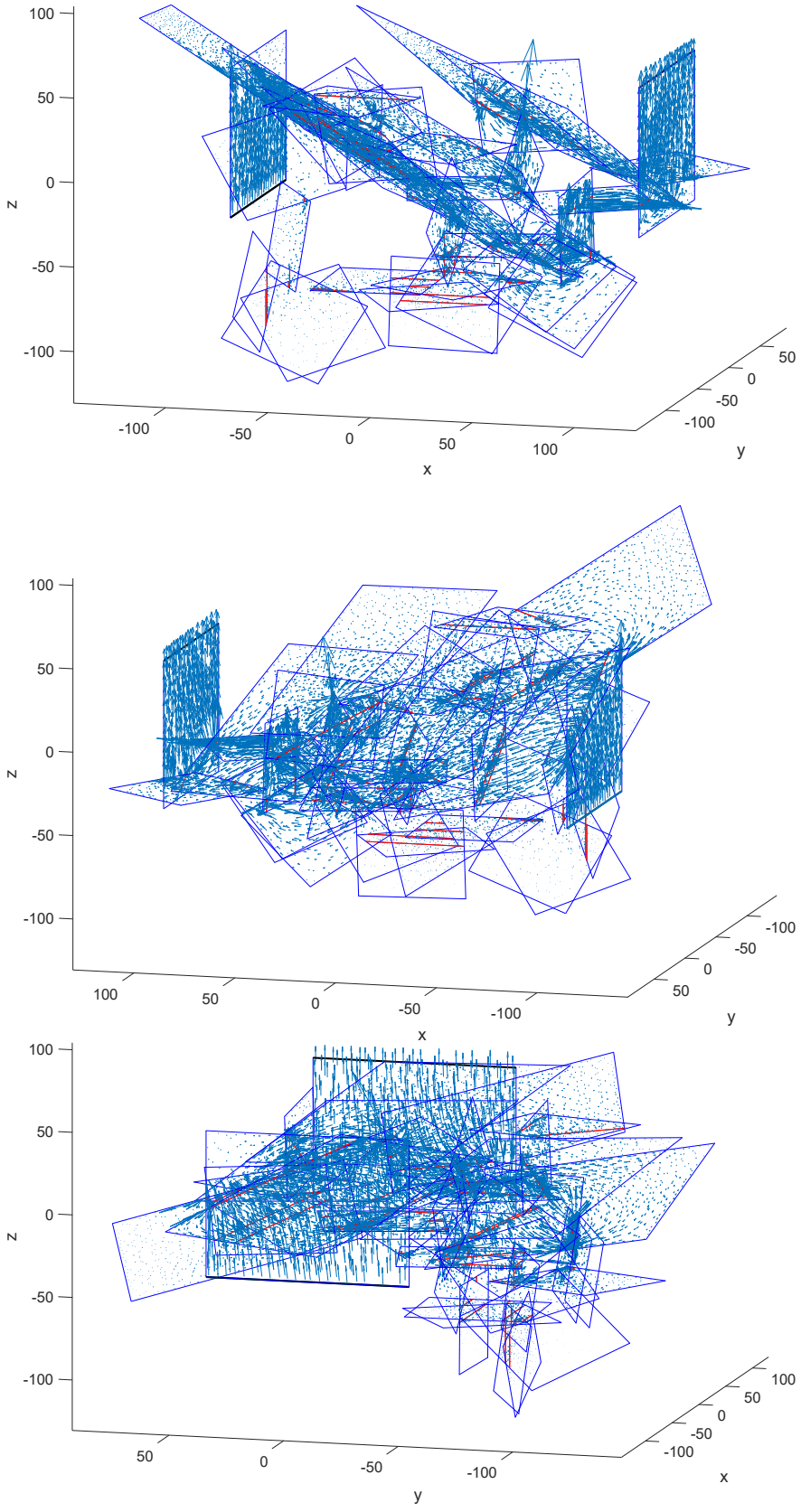


FIGURE 5.2.5: Different perspectives of the solution for the velocity field of a pure DFN problem with 36 fractures

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17	F18	F19	F20	F21	F22	F23	F24	F25	F26	F27	F28	F29	F30	F31	F32	F33	F34	F35	F36	$\sum$
T1	74.6	0	0	0	0	0	0	0	0	-74.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T2	621.5	0	0	0	0	0	0	0	0	0	0	-621.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T3	303.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-303.9	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T4	0	-176.1	176.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T5	0	-199.9	0	199.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T6	0	-624	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	624	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T7	0	0	-176.1	176.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T8	0	0	0	-376	376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T9	0	0	0	0	-376	376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T10	0	0	0	0	0	-376	376	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T11	0	0	0	0	0	0	-368.6	368.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T12	0	0	0	0	0	0	-76.1	0	76.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T13	0	0	0	0	0	0	68.7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-68.7	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T14	0	0	0	0	0	0	0	-101.6	101.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T15	0	0	0	0	0	0	0	-172.8	0	0	172.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T16	0	0	0	0	0	0	0	-94.2	0	0	0	0	0	94.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T17	0	0	0	0	0	0	0	0	-133.9	133.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T18	0	0	0	0	0	0	0	0	-171.1	0	0	171.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T19	0	0	0	0	0	0	0	0	44.6	0	0	0	0	0	-44.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T20	0	0	0	0	0	0	0	0	82.8	0	0	0	0	0	0	-82.8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T21	0	0	0	0	0	0	0	0	0	-59.4	59.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T22	0	0	0	0	0	0	0	0	0	0	-232.2	232.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T23	0	0	0	0	0	0	0	0	0	0	0	115.6	-115.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T24	0	0	0	0	0	0	0	0	0	0	0	102.6	0	-102.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T25	0	0	0	0	0	0	0	0	0	0	0	0	29.6	-29.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T26	0	0	0	0	0	0	0	0	0	0	0	0	192.1	0	-192.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T27	0	0	0	0	0	0	0	0	0	0	0	0	-106	0	0	0	0	0	0	0	0	0	0	0	0	0	106	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T28	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T29	0	0	0	0	0	0	0	0	0	0	0	0	0	39	0	0	0	0	0	-39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	226.2	-226.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27.4	0	0	0	0	0	0	0	0	0	0	-27.4	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-17.9	0	0	0	0	0	0	0	0	0	0	0	0	0	17.9	0	0	0	0	0	0	0	$\sim \epsilon$
T33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	53.3	-53.3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	27.1	0	-27.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	152.6	0	0	0	-152.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	77.1	0	0	0	0	0	0	0	0	-77.1	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.2	0	$\sim \epsilon$
T38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	86.9	-86.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.9	0	0	0	0	0	0	0	-1.9	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-18.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.5	0	0	$\sim \epsilon$
T41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-17.1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.1	$\sim \epsilon$
T42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.9	-6.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	107.1	0	0	0	0	0	-107.1	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6.9	-6.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47.2	-47.2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	300.5	0	0	0	-300.5	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21.4	0	0	0	0	-21.4	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-170.7	0	0	0	0	0	170.7	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T49	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	216.4	-216.4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-169.2	0	0	0	169.2	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	624	-624	0	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-407.6	0	407.6	0	0	0	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-60.9	60.9	0	0	0	0	0	0	0	0	0	$\sim \epsilon$
T54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-82.3	0	82.3	0	0	0	0	0	0	0	0	$\sim \epsilon$
T55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	82.3	-82.3	0	0	0	0	0	0	0	0	$\sim \epsilon$
T56	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	54.6	0	-54.6	0	0	0	0	0	0	0	$\sim \epsilon$
T57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	36.7	-36.7	0	0	0	0	0	0	$\sim \epsilon$
T58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	-10	0	0	0	0	0	$\sim \epsilon$
T59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26.7	0	0	-26.7	0	0	0	$\sim \epsilon$
T60	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8.6	8.6	0	0	0	0	$\sim \epsilon$
T61	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.6	0	0	-18.6	0	0	$\sim \epsilon$
T62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-8.6	8.6	0	0	0	$\sim \epsilon$
T63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.1	-18.1	0	0	$\sim \epsilon$
T64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18.2	-18.2	0	$\sim \epsilon$
T65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17.1	-17.1	$\sim \epsilon$
$\sum$	1000	-1000	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$	$\sim \epsilon$

TABLE 5.2.1: Flux exchange for DFN36 problem

5.3 Combining 2D and 3D elements

The main results for this chapter involve the computation of the complete pressure and velocity fields across the whole domain which comprises 3D elements and planar fractures.

5.3.1 Domains with a single fracture

5.3.1.1 Benchmark problem with exact solution

In order to assess the method in the presence of 3D-2D interaction, a benchmark problem with exact solution is analysed. The 3D domain, Ω_3, is the cube $[-1,1]^3$ while the domain for the single fracture is $F_1 = \{(x,y,z) : z = 0.5, -1 \leq x \leq 1, -1 \leq y \leq 1\}$. The problem will be a pure Darcy flow, *i.e.* constant (unitary) diffusion coefficient with neither convection nor reaction term for both matrix and fracture. The problem geometry is presented in Figure 5.3.1. The global solution for

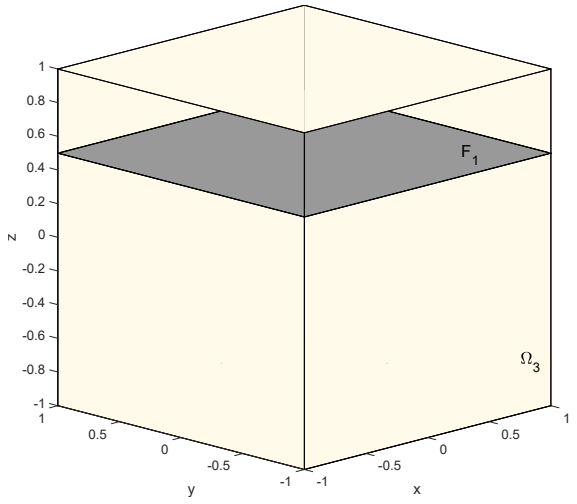


FIGURE 5.3.1: Geometry for a cube with a single fracture

the pressure was chosen to be of the lowest possible polynomial order. In fact, any exact solution of a problem with non-zero flux exchange between 2D and 3D elements will be of order at least 3, since the simplest possible jump in flux (constant jump) gives rise to a second order solution in the fracture, but has to be increased in order by one in the normal direction to include jump in the normal flux in Ω_3. The formula for the pressure is:

$$P(x,y,z) = \begin{cases} \frac{2}{3}(1-x^2)(z+1) & -1 \leq z < 0.5 \\ (1-x^2) - (z-\frac{1}{2})(\frac{4}{3}+\frac{2}{3}x^2) & 0.5 \leq z \leq 1 \end{cases}$$

which is continuous in the whole domain, corresponding to an infinite permeability between matrix and fracture. The velocity variable is

$$\boldsymbol{u}(x,y,z) = \begin{cases} \{-\frac{4}{3}x(1+z), 0, \frac{2}{3}(1-x^2)\} & -1 < z < 0.5 \\ \{-2x - \frac{4}{3}x(z-\frac{1}{2}), 0, -\frac{4}{3} - \frac{2}{3}x^2)\} & 0.5 < z \leq 1 \end{cases}$$

At $z = 0.5$, the vertical component of the velocity variable is not continuous. In fact,

$$\begin{aligned} [\![\boldsymbol{u}(x,y,z)\cdot(0,0,1)]\!]_{z=\frac{1}{2}} &= \left\{-\frac{4}{3}x(1+\frac{1}{2}), 0, \frac{2}{3}(1-x^2)\right\}\cdot\{(0,0,1)\} + \\ &\quad \left\{-2x - \frac{4}{3}x(\frac{1}{2}-\frac{1}{2}), 0, -\frac{4}{3} - \frac{2}{3}x^2\right\}\cdot\{(0,0,-1)\} \\ &= \frac{2}{3}(1-x^2) - (-\frac{4}{3} - \frac{2}{3}x^2) = 2 \end{aligned}$$

This is coherent with the solution of the problem in F_1, which is given by $P_{F_1}(x,y) = 1 - x^2$, since its loading term representing the incoming mass contribution from the 3D domain is $-\Delta P_{F_1}(x,y) = 2 = [\![\boldsymbol{u}(x,y,z)\cdot(0,0,1)]\!]_{z=\frac{1}{2}}$. Non-homogeneous Dirichlet boundary conditions are applied on the borders of the 3D and 2D domains, as well as the corresponding loading term for Ω_3. Note that the load term for F_1 arises from the coupling between dimensions and there is no need to provide it. The exact and discrete solution for the pressure head represented as slices of the domain including the fracture plane are compared in Figure 5.3.2 for order RT3-VEM, showing that the solution is attained within numerical accuracy, in bot the 3D domain as well as in the fracture.

5.3.1.2 Unidirectional flow

The geometry of the problem consists of the rock matrix $\Omega_3 = \{(x,y,z) : -2 \leq x \leq 2, -1 \leq y \leq 1, -1 \leq z \leq 1\}$ and the single fracture $F_1 = \{(x,y,z) : x = 0\}$, as shown in Figure 5.3.3 for a mesh of 20 polyhedral elements before the cut. An incoming flux of $1/4$ is applied to the left boundary while pressure is fixed at 0 for the right boundary. All other boundaries are isolated (no normal flux) so that the total flux entering is through the left boundary is one quarter of the area of the face, yielding 1. Unitary permeability is considered for both the matrix and the fracture. However, due to the expected result of a unidirectional flow from left to right, no flux field will be present in the fracture since flux will completely pass through it. Finally, the pressure of the problem will be dependent on η_1, the normal permeability between F_1 and Ω_3. For this case, $\eta_1 = 2$ is taken so a jump in pressure is expected on either side of the fracture since global continuity of pressure is only imposed when the permeability is arbitrarily high. The problem was solved using order 1 elements, since they capture

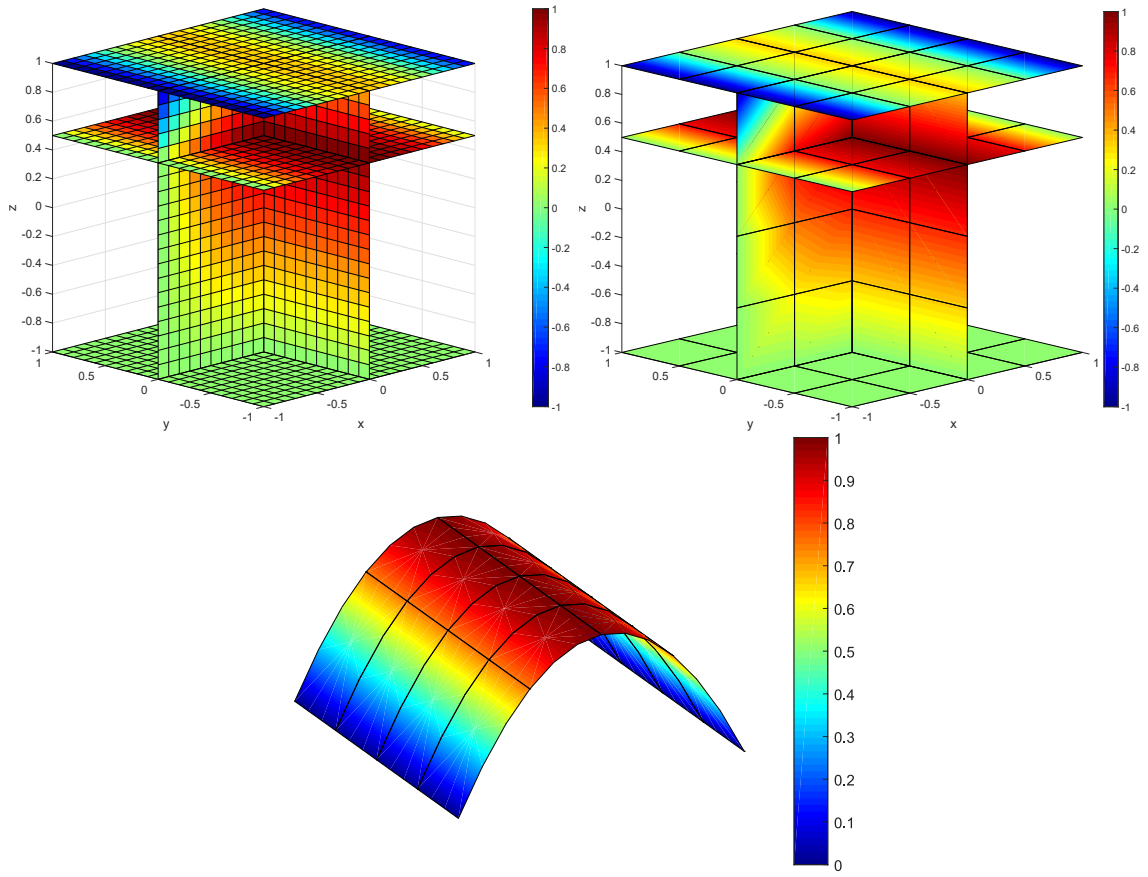


FIGURE 5.3.2: Exact and discrete solution for a cube with a single fracture (above) and discrete solution on F_1 (below)

the exact solution given that the pressure variation is linear. Pressure results are given in Figure 5.3.4, where the mesh was exploded to show the constant pressure on F_1, indicating that there is no 2-dimensional flow present. The jump in pressure is consistent with the coupling conditions (2.2.2) since it is satisfied that

$$1 = (\boldsymbol{u}^3_+ \cdot \hat{\boldsymbol{n}}^1)|L_{Face}| = \eta_1(P^3_+|_{F_1} - P^{2,1}) = 2(1.5 - 1) = 1$$
$$-1 = (\boldsymbol{u}^3_- \cdot \hat{\boldsymbol{n}}^1)|L_{Face}| = \eta_1(P^3_-|_{F_1} - P^{2,1}) = 2(0.5 - 1) = -1.$$

5.3.2 Complex embedded DFN

5.3.2.1 Matrix+DFN6

This problem revisits the DFN presented in Section 5.2.1, but now including the interaction with the surrounding rock matrix whose geometry is $\Omega_3 = [-1, 1]^3$. The same boundary conditions are

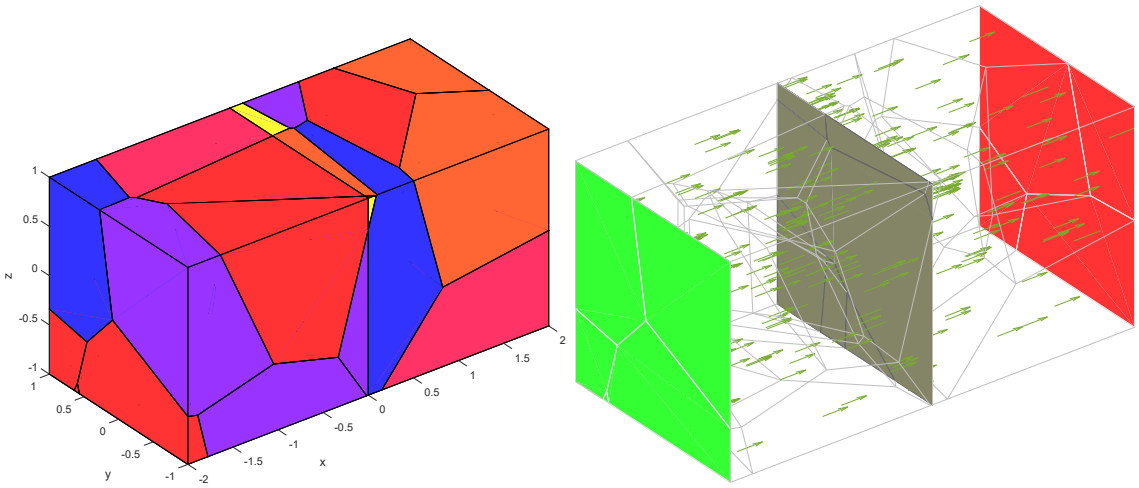


FIGURE 5.3.3: Geometry and mesh for the Unidirectional flow problem (left). Velocity field (right)

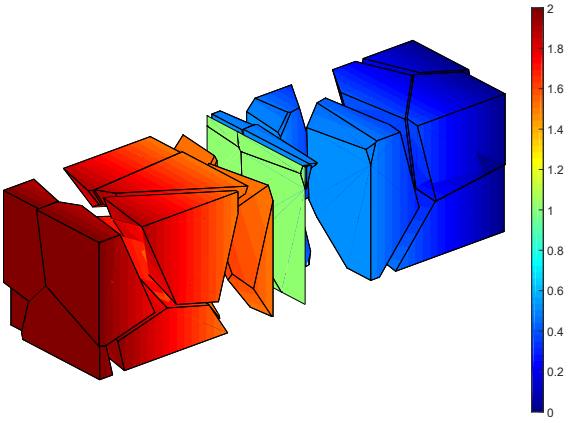


FIGURE 5.3.4: Pressure field

imposed on the system, namely, a incoming flux on the source fracture and a null pressure on the sink fracture. The boundary of the rock matrix is considered fully isolated. The permeability data is 1 for all fractures and the rock permeability is 0.1, *i.e.* $\frac{\kappa^{3D}}{\kappa^{2D}} = 0.1$. This is a very different condition from considering impervious rock matrix as before. The normal permeability η_r will be set to $1/2$ for $r = 1, ...6$, so that a jump in pressure is visible between fractures and rock matrix. The problem was solved using RT2-VEM elements with 160 3D elements with 7824 DOFs and 116 2D elements with 2341 DOFs, showing how much more demanding is a hybrid problem with respect to a pure DFN problem from a computational point of view.

The pressure and velocity fields are shown in Figures 5.3.5 and 5.3.6 respectively. The first graphic had some elements removed to show the underlying DFN, while in the second it can be seen that there is a flux leaving the fractures (blue arrows), that is due to the somewhat comparable permeability of the matrix that provides new paths from source to sink boundaries. Once again the flux chart is provided (Figure 5.3.7) where now the flux exchange between matrix and fractures is

represented. It can be concluded that the influence of the rock matrix in the analysis is considerable, and should not be disregarded when the permeabilities are only one order of magnitude apart.

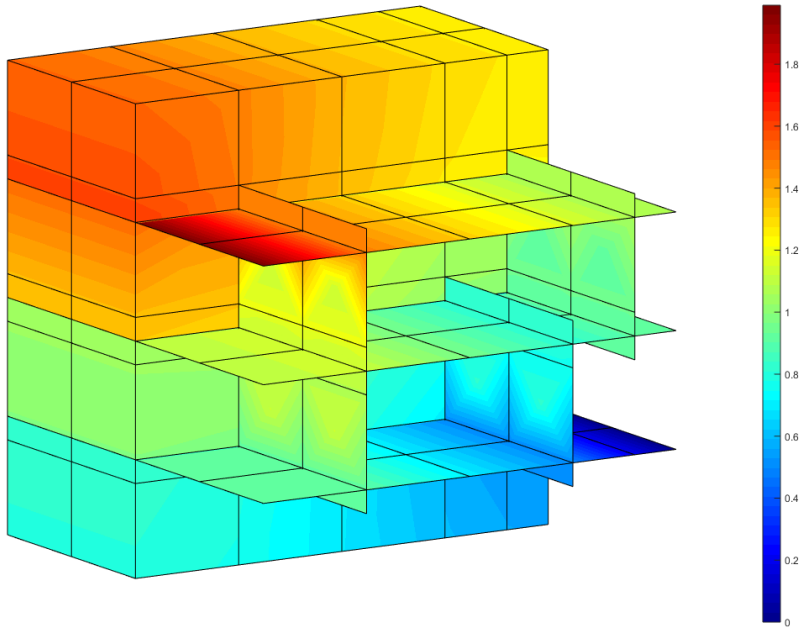


FIGURE 5.3.5: Pressure head solution for the problem Matrix+DFN6

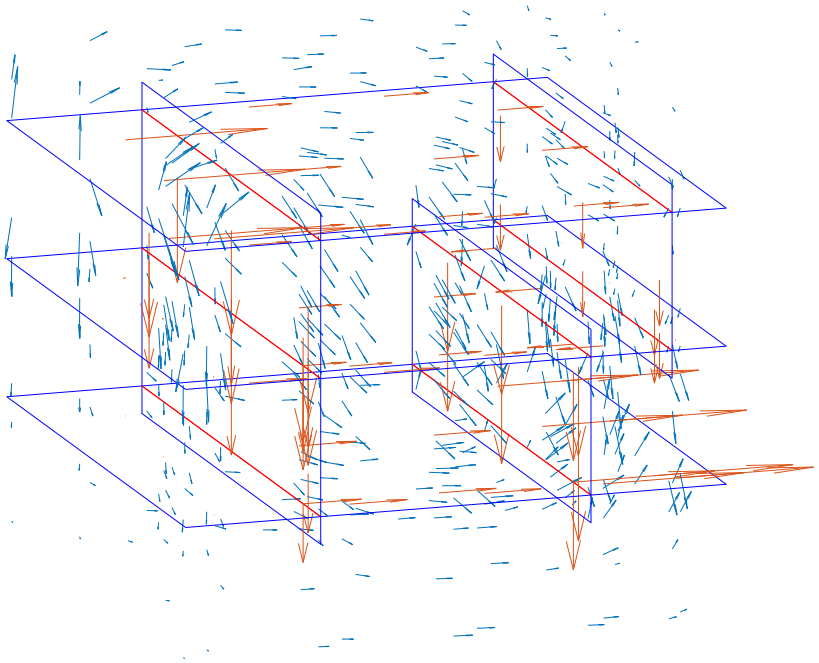


FIGURE 5.3.6: Global velocity field for the problem Matrix+DFN6

5.3.2.2 Matrix+DFN10

In this problem, a DFN of 10 fractures and 38 traces is embedded into a matrix whose domain is $\Omega_3 = [-1, 1]^3$ as shown in Figure 5.3.8 where fractures are depicted in blue. The pressure is

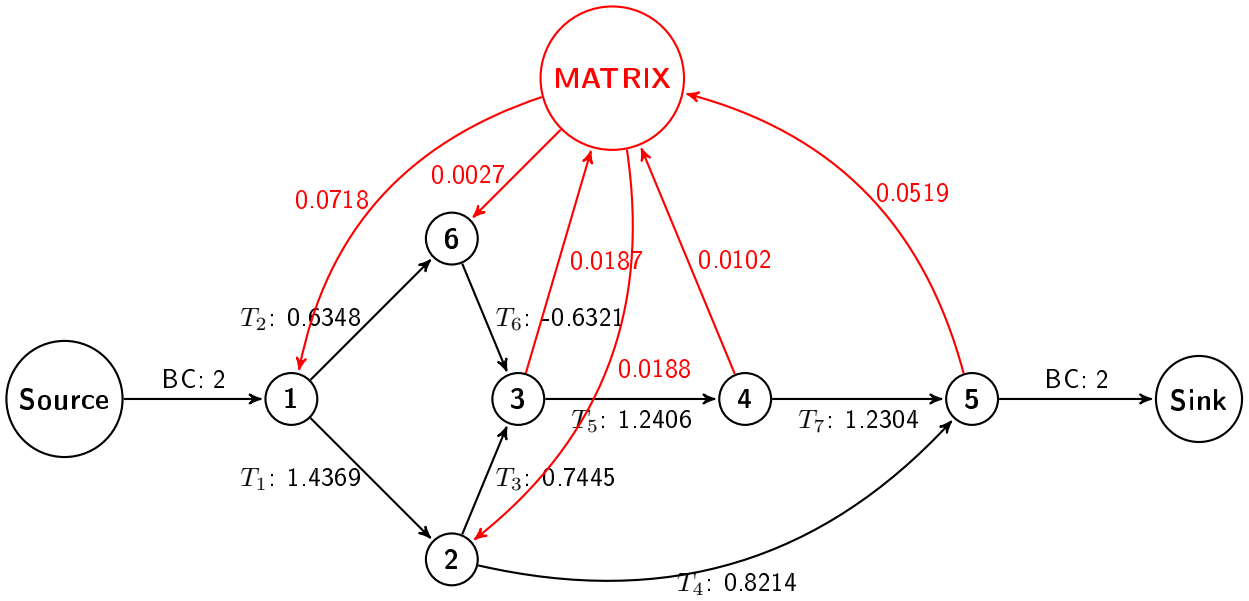


FIGURE 5.3.7: Flux path on a 6 fracture DFN embedded in the rock matrix, with $\frac{\kappa^{3D}}{\kappa^{2D}} = 0.1$

prescribed as $P = 1$ on $z = 1$, $P = 0$ on $z = -1$ and isolated boundaries for the other 4 faces. The starting point for the discretization is a polyhedral mesh comprised of 100 elements, which, after 10 successive cuts, one for each fracture, results in the final mesh (Figure 5.3.9). 10 can be seen as a very low number of fractures to discretize a network, which can amount to thousands of fractures. However, even this relatively contained example, the demand for computational power is very high. Not only that, the production of this globally conforming mesh that includes the planes is very demanding as well, and the resulting mesh contains a plethora of very badly shaped elements with many undesirable features such as: small angles, faces and edges, large discrepancies in size between adjacent elements, collapsing nodes, etc. It would be much more expensive to solve this problem with a conforming Finite Element mesh made up of the usual shapes (tetrahedra, piramids, wedges, and hexahedra) that will require very small element sizes to provide an acceptable mesh. However, VEM has been shown to be very robust to mesh distortion and the versatility of allowing arbitrary polyhedral shapes greatly simplifies the meshing process.

For the study of the problem,the settings considered are: a ratio between permeabilities of $\frac{\kappa^{3D}}{\kappa^{2D}} = (10)^{-7}$, and $\eta_r = 10$ for $r = 1, ..., 10$, so that there will be a jump in pressure across fractures and the flux will be mainly DFN-dominated, except in the zones where the flux has no other alternative to arrive to the bottom than through the matrix. The problem was solved using RT0-VEM, RT1-VEM and RT2-VEM, whose details of discretization are provided in Table 5.3.1.

In the next Figures the results of the analysis are presented: The global pressure and the

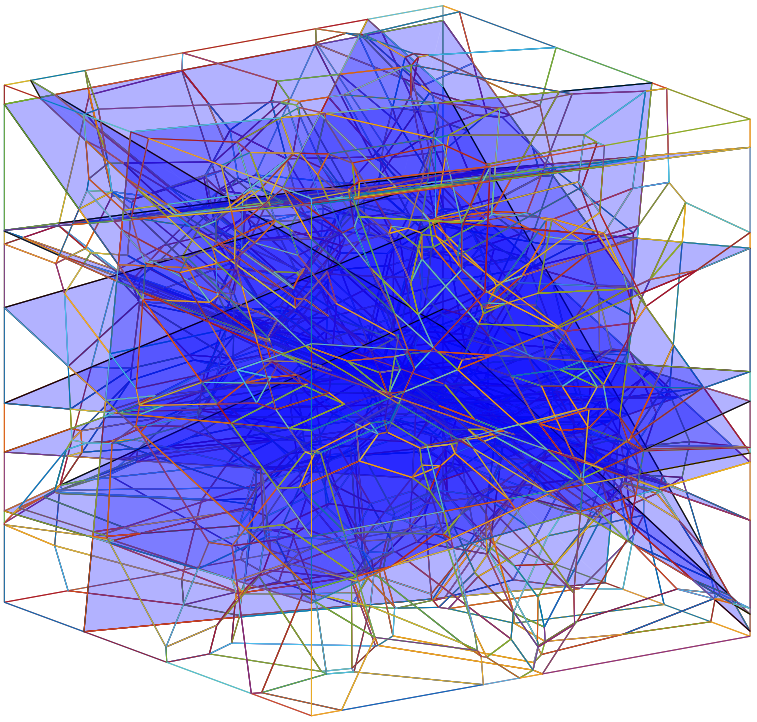


FIGURE 5.3.8: 10 fracture DFN with embedded DFN

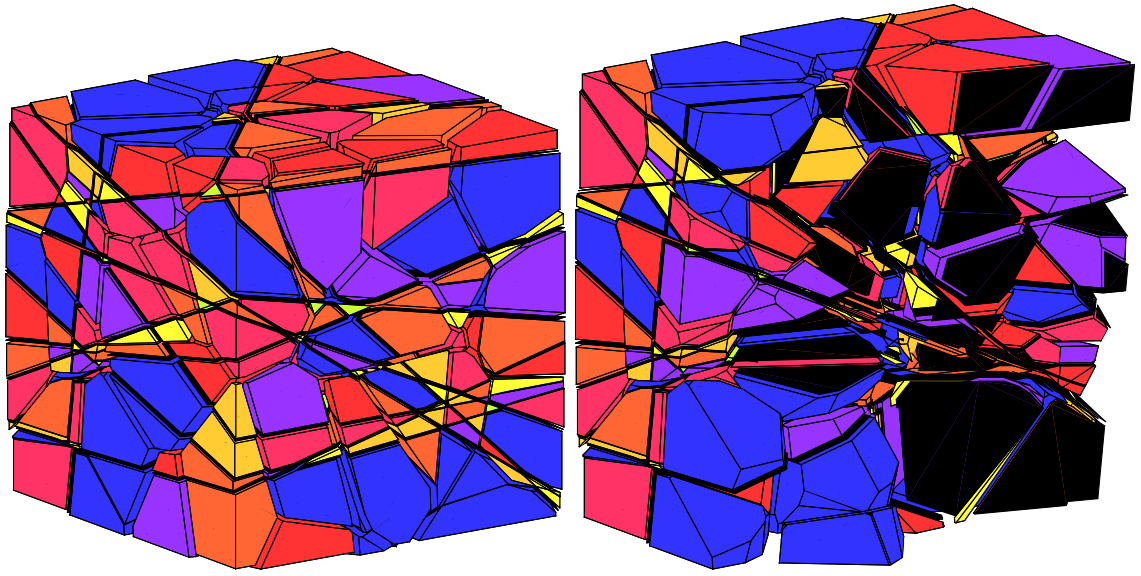


FIGURE 5.3.9: Final VEM mesh for the problem Matrix+DFN10, exploded and sliced for clarity. Fracture planes are shown in black

velocity field are shown in 5.3.10 while the pressure and (normalized) velocity field for a particular fracture (F_6) in 5.3.11, for RT2-VEM. The latter Figure clearly shows the jump in the co-normal derivative of the solution across a trace, showing flux exchange between fractures.

Some conclusions can be taken from this problem: the solution of the problem is highly dependent on the fracture distribution, as expected. In fact, flux is greatest on fractures that

Element Type	#Elements		#3D DOFs		#2D DOFs	
	3D	2D	Flux	Pressure	Flux	Pressure
RT0-VEM	748	1016	3648	748	2941	1016
RT1-VEM	748	1016	15432	2992	8930	3048
RT2-VEM	748	1016	36848	7480	16951	6096

TABLE 5.3.1: Data for the discretization for the problem Matrix+DFN10

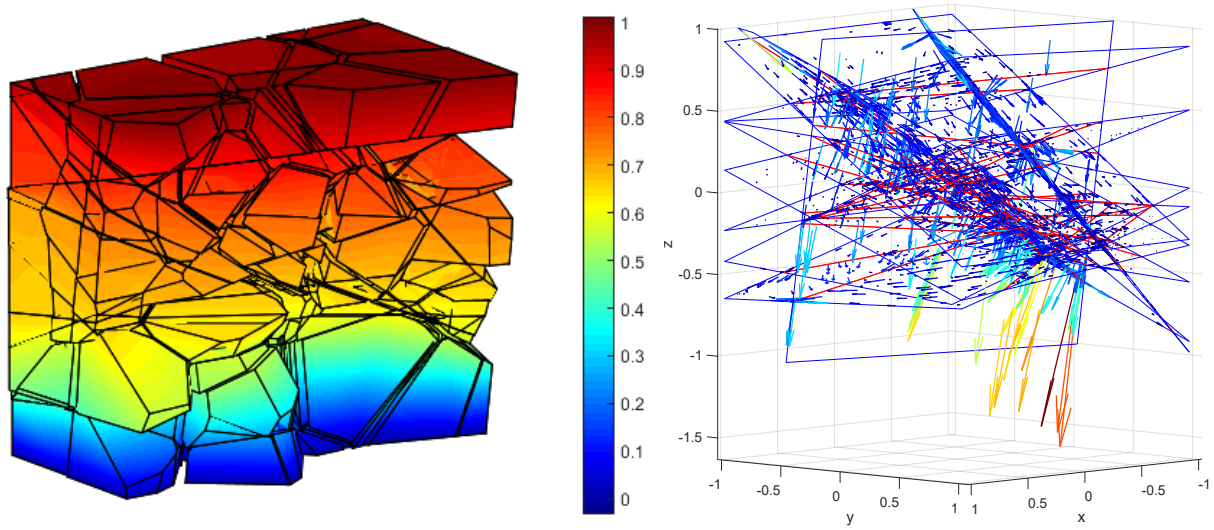


FIGURE 5.3.10: Global Pressure head (left) and velocity field (right) solution

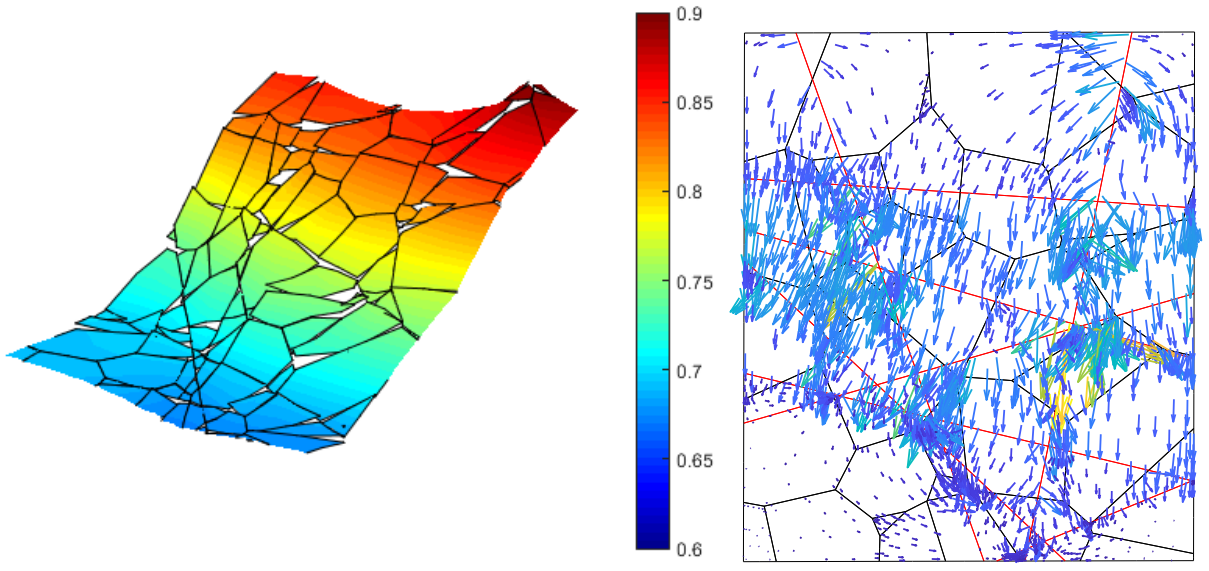


FIGURE 5.3.11: Pressure head (above) and velocity field (below) solution for F_6

are almost vertical and provide the least-effort path between boundaries with prescribed pressure conditions. The relative permeability between rock matrix and fractures will influence the results considerably, unlike the orthogonal permeability η that does not seem to influence the results very much besides penalizing the lack of continuity of pressure between 3D and fractures. In addition, taking into account the rock matrix in the model makes it significantly more expensive to solve, and

should be avoided whenever the rock permeability is low enough such that the problem is mainly DFN-dominated. Furthermore, the method was very robust and is affected very little by mesh quality. Finally, global conservation of flux provides very reliable results for the flux variable, even for the lowest order elements.

Chapter 6

Conclusions

A mixed Virtual Element formulation for 2D and 3D domains for solving elliptic partial differential equations was explored in this work. A summary of the main theoretical results as well as the main ideas of the method were included. Details about its implementation were given a thorough treatment for various arbitrary discretization order. Many results were provided, including patch-tests and benchmark problems for the pure 3D case and examples of pure 2D DFN. Furthermore, numerical experiments involving the combination of 3D and 2D elements representing embedded fractures in a solid matrix have shown the viability of the method. In particular, whereas the primal formulation guarantees continuity of pressure head, the mixed formulation has the property of being completely conservative, which is a more desirable characteristic in flux computations. Some generalizations of the approach are clearly possible and most definitively within easy reach. For instance, including 1D elements to represent flux on traces as well as requiring flux balance on the points defined by traces intersection. It should also be straightforward to generalize the approach to second order elliptic problems. Additionally, time-dependent problems should pose no problem and follow standard procedure once the mass and stiffness matrix for the VEM discretization are obtained. From a computational point of view, the method shows potential for parallelization, since each dimension of the problem can be computed independently (even in parallel themselves) and the coupling between dimensions can be added once the respective stiffness matrices have been computed. In conclusion, the Mixed Virtual Element method shows a lot of promise for the study of geometrically complex hibrid dimensional flow problem.

Appendix A

A.1 Basis for polynomials spaces

Tables A.1.1 and A.1.2 list all polynomials comprising the space of gradients and its complement. Figure A.1.1 is an schematic representation of Pascal's pyramid, which contains the ordering of the polynomials used in the implementation of the method. $\overline{x}$, $\overline{y}$ and $\overline{z}$ should be understood as $\frac{(x-x_0)}{d}$, $\frac{(y-y_0)}{d}$ and $\frac{(z-z_0)}{d}$, where the subscript 0 indicates the barycenter of the element a d its diameter.

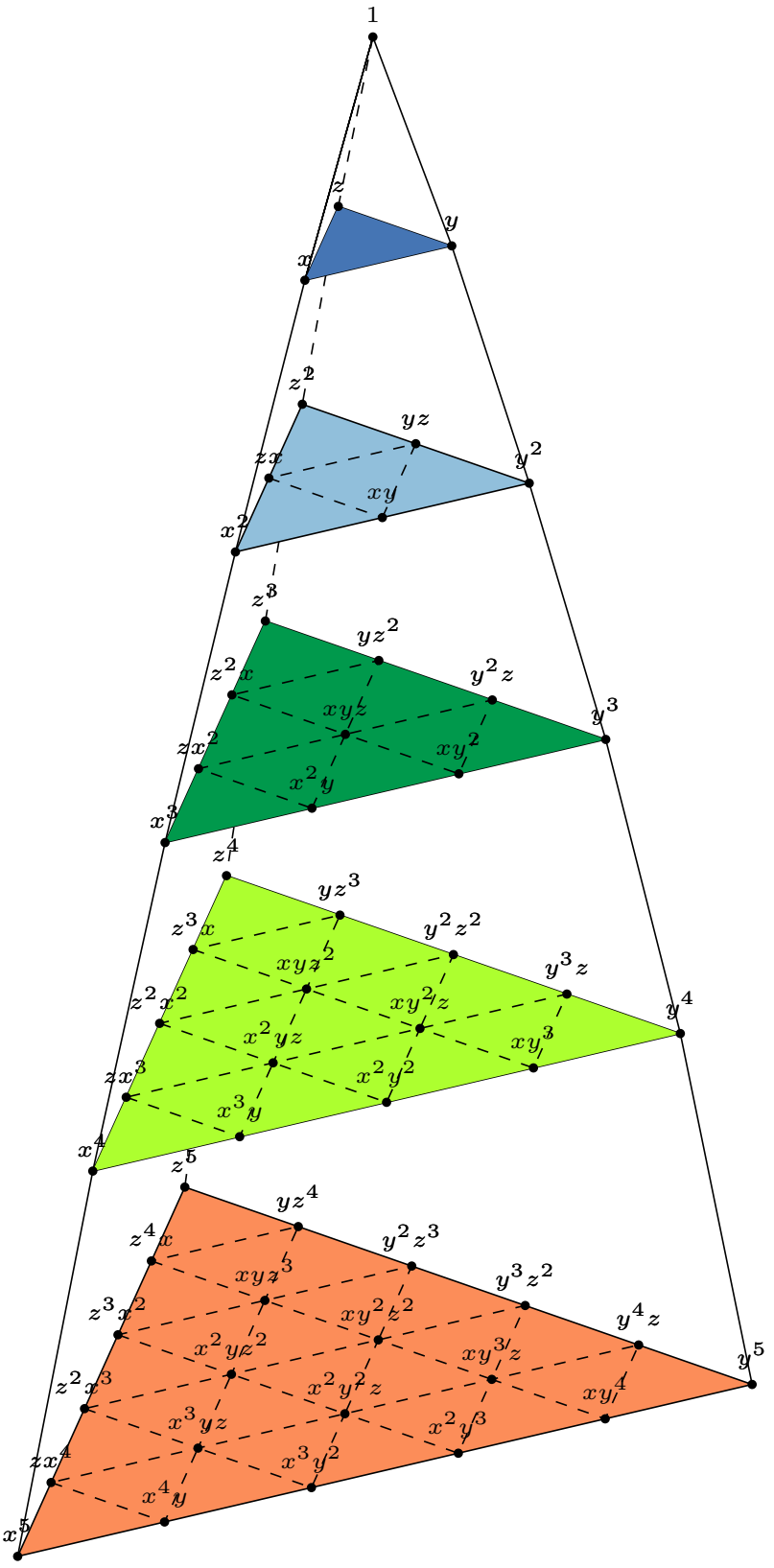


FIGURE A.1.1: Pascal's pyramid

TABLE A.1.1: Decomposition of spaces $\mathcal{G}_{0,1,2,3}$

k	#	$\nabla\mathbb{P}_{k+1}$	$(\nabla\mathbb{P}_{k+1})^{\oplus}$
0	1	$(1,0,0)$	
$\frac{1}{d^E}$	2	$(0,1,0)$	
	3	$(0,0,1)$	
1	4	$(2x,0,0)$	
$\frac{1}{(d^E)^2}$	5	$(y,x,0)$	
	6	$(0,2y,0)$	
	7	$(0,z,y)$	
	8	$(0,0,2z)$	
	9	$(z,0,x)$	
$\frac{1}{d^E}$	10		$(y,-x,0)$
	11		$(z,0,-x)$
	12		$(0,z,-y)$
2	13	$\big(3x^2,0,0\big)$	
$\frac{1}{(d^E)^3}$	14	$\big(2xy,x^2,0\big)$	
	15	$\big(y^2,2xy,0\big)$	
	16	$\big(0,3y^2,0\big)$	
	17	$\big(0,2yz,y^2\big)$	
	18	$\big(0,z^2,2yz\big)$	
	19	$\big(0,0,3z^2\big)$	
	20	$\big(z^2,0,2xz\big)$	
	21	$\big(2xz,0,x^2\big)$	
	22	(yz,xz,xy)	
$\frac{1}{(d^E)^2}$	23		$\big(xy,-2x^2,0\big)$
	24		$\big(-2y^2,-xy,0\big)$
	25		$(yz,0,-xy)$
	26		$\big(0,yz,-2y^2\big)$
	27		$\big(0,2z^2,yz\big)$
	28		$(xz,0,-2x)$
	29		$(0,xz,-xy)$
	30		$\big(-2z^2,0,xz\big)$

k	#	$\nabla\mathbb{P}_{k+1}$	$(\nabla\mathbb{P}_{k+1})^{\oplus}$
3	31	$\big(4x^3,0,0\big)$	
$\frac{1}{(d^E)^4}$	32	$\big(3x^2y,x^3,0\big)$	
	33	$\big(2xy^2,2x^2y,0\big)$	
	34	$\big(y^3,3xy^2,0\big)$	
	35	$\big(0,4y^3,0\big)$	
	36	$\big(0,3y^2z,y^3\big)$	
	37	$\big(0,2yz^2,2y^2z\big)$	
	38	$\big(0,z^3,3yz^2\big)$	
	39	$\big(0,0,4z^3\big)$	
	40	$\big(z^3,0,3xz^2\big)$	
	41	$\big(2xz^2,0,2x^2z\big)$	
	42	$\big(3x^2z,x^2z,x^3\big)$	
	43	$\big(2xyz,x^2z,x^2y\big)$	
	44	$\big(y^2z,2xyz,xy^2\big)$	
	45	$\big(yz^2,xz^2,2xyz\big)$	
$\frac{1}{(d^E)^3}$	46		$\big(x^2y,-3x^3,0\big)$
	47		$\big(xy^2,-x^2y,0\big)$
	48		$\big(y^3,\frac{-1}{3}xy^2,0\big)$
	49		$\big(y^2z,0,-xy^2\big)$
	50		$\big(0,y^2z,-3y^3\big)$
	51		$\big(0,yz^2,-y^2z\big)$
	52		$\big(0,z^3,\frac{-1}{3}yz^2\big)$
	53		$\big(-yz^2,xz^2,0\big)$
	54		$\big(-3z^3,0,xz^2\big)$
	55		$\big(x^2z,0,-3x^3\big)$
	56		$\big(0,x^2z,-x^2y\big)$
	57		$\big(-xz^2,0,x^2z\big)$
	58		$\big(xyz,0,-2x^2y\big)$
	59		$\big(0,xyz,-2xy^2\big)$
	60		$\big(-2yz^2,0,xyz\big)$

TABLE A.1.2: Decomposition of spaces $\mathcal{G}_{4,5}$

k	#	$\nabla\mathbb{P}_{k+1}$	$(\nabla\mathbb{P}_{k+1})^{\oplus}$
4	61	$\left(5x^4, 0, 0\right)$	
$\frac{1}{(d^E)^5}$	62	$\left(4x^3y, x^4, 0\right)$	
	63	$\left(3x^2y^2, 2x^3y, 0\right)$	
	64	$\left(2xy^3, 3x^2y^2, 0\right)$	
	65	$\left(y^4, 4xy^3, 0\right)$	
	66	$\left(0, 5y^4, 0\right)$	
	67	$\left(0, 4y^3z, y^4\right)$	
	68	$\left(0, 3y^2z^2, 2y^3z\right)$	
	69	$\left(0, 2yz^3, 3y^2z^2\right)$	
	70	$\left(0, z^4, 4yz^3\right)$	
	71	$\left(0, 0, 5z^4\right)$	
	72	$\left(z^4, 0, 4xz^3\right)$	
	73	$\left(2xz^3, 0, 3x^2z^2\right)$	
	74	$\left(3x^2z^2, 0, 2x^3z\right)$	
	75	$\left(4x^3z, 0, x^4\right)$	
	76	$\left(3x^2yz, x^3z, x^3y\right)$	
	77	$\left(2xy^2z, 2x^2yz, x^2y^2\right)$	
	78	$\left(y^3z, 3xy^2z, xy^3\right)$	
	79	$\left(y^2z^2, 2xyz^2, 2xy^2\right)$	
	80	$\left(yz^3, xz^3, 3xyz^2\right)$	
	81	$\left(2xyz^2, x^2z^2, 2x^2yz\right)$	
$\frac{1}{(d^E)^4}$	82		$\left(-3yz^3, 0, xyz^2\right)$
	83		$\left(-2yz^2, xyz^2, 0\right)$
	84		$\left(xyz^2, -2x^2z^2, 0\right)$
	85		$\left(-2y^2z^2, 0, xyz^2\right)$
	86		$\left(0, xy^2z, -3xy^3\right)$
	87		$\left(xy^2z, 0, -2x^2y^2\right)$
	88		$\left(0, -2x^2z^2, -2x^2y^2\right)$
	89		$\left(0, x^2yz, x^2yz\right)$
	90		$\left(x^2yz, 0, -2x^2y^2\right)$
	91		$\left(-2x^2z^2, 0, 3x^3z\right)$
	92		$\left(0, x^3z, -3x^3y\right)$
	93		$\left(x^3z, 0, -4x^4\right)$
	94		$\left(-3xz^3, 0, 2x^2z^2\right)$
	95		$\left(-4z^4, 0, xz^3\right)$
	96		$\left(-yz^3, xz^3, 0\right)$
	97		$\left(0, 4z^4, -yz^3\right)$
	98		$\left(0, 3yz^3, -2y^2z^2\right)$
	99		$\left(0, 2y^2z^2, -3y^3z\right)$
	100		$\left(0, y^3z, -4y^4\right)$
	101		$\left(y^3z, 0, -3xy^3\right)$
	102		$\left(4y^4, -xy^3, 0\right)$
	103		$\left(3xy^3, -2x^2y^2, 0\right)$
	104		$\left(2x^2y^2, -3x^3y, 0\right)$
	105		$\left(x^3y, -4x^4, 0\right)$

k	#	$\nabla\mathbb{P}_{k+1}$	$(\nabla\mathbb{P}_{k+1})^{\oplus}$
5	106	$(6x^5, 0, 0)$	
$\frac{1}{(d^E)^6}$	107	$(5x^4y, x^5, 0)$	
	108	$(4x^3y^2, 2yx^4, 0)$	
	109	$(3x^2y^3, 3x^3y^2, 0)$	
	110	$(2xy^4, 4x^2y^3, 0)$	
	111	$(y^5, 5xy^4, 0)$	
	112	$(0, 6y^5, 0)$	
	113	$(0, 5y^4z, y^5)$	
	114	$(0, 4y^3z^2, 2zy^4)$	
	115	$(0, 3y^2z^3, 3y^3z^2)$	
	116	$(0, 2yz^4, 4y^2z^3)$	
	117	$(0, z^5, 5yz^4))$	
	118	$(0, 0, 6z^5)$	
	119	$(z^5, 0, 5xz^4)$	
	120	$(2xz^4, 0, 4x^2z^3)$	
	121	$(3x^2z^3, 0, 3x^3z^2)$	
	122	$(4x^3z^2, 0, 2zx^4)$	
	123	$(5x^4z, 0, x^5)$	
	124	$(4x^3yz, x^4z, x^4y)$	
	125	$(3x^2y^2z, 2yx^3z, x^3y^2)$	
	126	$(2xy^3z, 3x^2y^2z, x^2y^3)$	
	127	$(y^4z, 4xy^3z, xy^4)$	
	128	$(y^3z^2, 3xy^2z^2, 2zxy^3)$	
	129	$(y^2z^3, 2yxz^3, 3xy^2z^2)$	
	130	$(yz^4, xz^4, 4xyz^3)$	
	131	$(2xyz^3, x^2z^3, 3x^2yz^2)$	
	132	$(3x^2yz^2, x^3z^2, 2zx^3y)$	
	133	$(2xy^2z^2, 2yx^2z^2, 2zx^2y^2)$	
$\frac{1}{(d^E)^5}$	134		$(0, -3x^2z^3, x^2yz^2)$
	135		$(0, x^2yz^2, -x^2y^2z)$
	136		$(x^2yz^2, -3x^3z^2, 0)$
	137		$(-4yz^4, 0, xyz^3)$
	138		$(-2y^2z^3, xyz^3, 0)$
	139		$(xyz^3, -2x^2z^3, 0)$
	140		$(-3y^2z^3, 0, xy^2z^2)$
	141		$(-3y^3z^2, xy^2z^2, 0)$
	142		$(xy^2z^2, 0, -x^2y^2z)$
	143		$(-2y^3z^2, 0, xy^3z)$
	144		$(0, xy^3z, -4xy^4)$
	145		$(xy^3z, 0, -2x^2y^3)$
	146		$(0, x^2y^2z, -3x^2y^3)$
	147		$(x^2y^2z, 0, -3x^3y^2)$
	148		$(0, -2x^3z^2, x^3yz)$
	149		$(0, x^3yz, -2x^3y^2)$
	150		$(x^3yz, 0, -4x^4y)$
	151		$(-x^3z^2, 0, 2x^4z)$
	152		$(0, x^4z, -x^4y)$
	153		$(x^4z, 0, -5x^5)$
	154		$(-x^2z^3, 0, x^3z^2)$
	155		$(-2xz^4, 0, x^2z^3)$
	156		$(-5z^5, 0, xz^4)$
	157		$(-yz^4, xz^4, 0)$
	158		$(0, 5z^5, -yz^4)$
	159		$(0, 2yz^4, -y^2z^3)$
	160		$(0, y^2z^3, -y^3z^2)$
	161		$(0, y^3z^2, -2y^4z)$
	162		$(0, y^4z, -5y^5)$
	163		$(y^4z, 0, -xy^4)$
	164		$(5y^5, -xy^4, 0)$
	165		$(2xy^4, -x^2y^3, 0)$
	166		$(x^2y^3, -x^3y^2, 0)$
	167		$(x^3y^2, -2x^4y, 0)$
	168		$(x^4y, -5x^5, 0)$

Bibliography

[1] I. Aavatsmark, *An introduction to multipoint flux approximations for quadrilateral grids*, Computational Geosciences, 6 (2002), pp. 405–432.

[2] O. Al-Hinai, S. Srinivasan, and M. F. Wheeler, *Domain decomposition for flow in porous media with fractures*, in SPE Reservoir Simulation Symposium 23-25 February 2013, Houston, Texas, USA, Society of Petroleum Engineers, 2015.

[3] O. Al-Hinai, S. Srinivasan, and M. F. Wheeler, *Mimetic finite differences for flow in fractures from microseismic data*, in SPE Reservoir Simulation Symposium, Society of Petroleum Engineers, 2015.

[4] P. Angot, F. Boyer, and F. Hubert, *Asymptotic and numerical modelling of flows in fractured porous media*, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 43 (2009), pp. 239–275.

[5] P. F. Antonietti, L. Formaggia, A. Scotti, M. Verani, and N. Verzott, *Mimetic finite difference approximation of flows in fractured porous media*, ESAIM: Mathematical Modelling and Numerical Analysis, 50 (2016), pp. 809–832.

[6] 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.

[7] E. Artioli, S. de Miranda, C. Lovadina, and L. Patruno, *A stress/displacement Virtual Element method for plane elasticity problems*, Computer Methods in Applied Mechanics and Engineering, 325 (2017), pp. 155 – 174.

[8] 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.

[9] L. Beirão da Veiga, F. Brezzi, A. Cangiani, G. Manzini, L. D. Marini, and A. Russo, *Basic principles of Virtual Element methods*, Math. Models and Methods in Applied Sciences, (2013).

[10] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *Mixed virtual element methods for general second order elliptic problems on polygonal meshes*, ESAIM: Mathematical Modelling and Numerical Analysis, 50 (2016), pp. 727–747.

[11] 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.

[12] L. Beirão da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *H(div) and H(curl)-conforming virtual element methods*, Numer. Math., 133 (2016), pp. 303–332.

[13] M. F. Benedetto, S. Berrone, and A. Borio, *The Virtual Element Method for underground flow simulations in fractured media*, in Advances in Discretization Methods, vol. 12 of SEMA SIMAI Springer Series, Springer International Publishing, Switzerland, 2016, pp. 167–186.

[14] M. F. Benedetto, S. Berrone, S. Pieraccini, and S. Scialò, *The virtual element method for discrete fracture network simulations*, Comput. Methods Appl. Mech. Engrg., 280 (2014), pp. 135 – 156.

[15] M. F. Benedetto, S. Berrone, and S. Scialò, *A globally conforming method for solving flow in discrete fracture networks using the virtual element method*, Finite Elements in Analysis and Design, 109 (2016), pp. 23 – 36.

[16] M. F. Benedetto, A. Borio, and S. Scialò, *Mixed virtual elements for discrete fracture network simulations*, Finite Elements in Analysis and Design, 134 (2017), pp. 55 – 67.

[17] S. Berrone, A. Borio, C. Fidelibus, S. Pieraccini, S. Scialò, and F. Vicini, *Advanced computation of steady-state fluid flow in discrete fracture-matrix models: Fem–bem and vem–vem fracture-block coupling*, GEM - International Journal on Geomathematics, 9 (2018), pp. 377–399.

[18] S. Berrone, C. Canuto, S. Pieraccini, and S. Scialò, *Uncertainty quantification in discrete fracture network models: stochastic fracture transmissivity*, Comput. Math. Appl., 70 (2015), pp. 603–623.

[19] S. Berrone, S. Pieraccini, and S. Scialò, *Flow simulations in porous media with immersed intersecting fractures*, Journal of Computational Physics, 345 (2017), pp. 768 – 791.

[20] 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.

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

[22] ——, *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, and S. Scialò, *Non-stationary transport phenomena in networks of fractures: Effective simulations and stochastic analysis*, Computer Methods in Applied Mechanics and Engineering, 315 (2017), pp. 1098 – 1112.

[24] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Element Methods and Applications*, no. 44 in Springer Series in Computational Mathematics, Springer-Verlag Berlin Heidelberg, 2013.

[25] W. Boon, J. Nordbotten, and I. Yotov, *Robust discretization of flow in fractured porous media*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 2203–2233.

[26] K. Brenner, M. Groza, C. Guichard, G. Lebeau, and R. Masson, *Gradient discretization of hybrid dimensional Darcy flows in fractured porous media*, Numerische Mathematik, 134 (2016), pp. 569–609.

[27] K. Brenner, J. Hennicker, R. Masson, and P. Samier, *Gradient discretization of hybrid-dimensional Darcy flow in fractured porous media with discontinuous pressures at matrix-fracture interfaces*, IMA Journal of Numerical Analysis, (2016).

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

[29] E. Cáceres and G. N. Gatica, *A mixed virtual element method for the pseudostress–velocity formulation of the stokes problem*, IMA Journal of Numerical Analysis, 37 (2017), pp. 296–331.

[30] E. Cáceres, G. N. Gatica, and F. A. Sequeira, *A mixed virtual element method for the brinkman problem*, Mathematical Models and Methods in Applied Sciences, 27 (2017), pp. 707–743.

[31] ——, *A mixed virtual element method for quasi-newtonian stokes flows*, SIAM Journal on Numerical Analysis, 56 (2018), pp. 317–343.

[32] F. A. Chave, D. A. Di Pietro, and L. Formaggia, *A Hybrid High-Order method for Darcy flows in fractured porous media* , SIAM Journal on Scientific Computing, 40 (2018), pp. A1063–A1094.

[33] L. B. da Veiga, F. Brezzi, L. D. Marini, and A. Russo, *Virtual Element Implementation for General Elliptic Equations*, Springer International Publishing, Cham, 2016, pp. 39–71.

[34] D´Angelo, Carlo and Scotti, Anna, *A mixed finite element method for darcy flow in fractured porous media with non-matching grids*, ESAIM: M2AN, 46 (2012), pp. 465–489.

[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. Douglas and J. E. Roberts, *Global estimates for mixed methods for second order elliptic equations*, Mathematics of Computation, 44 (1985), pp. 39–52.

[37] J. Douglas Jr. and J. E. Roberts, *Global estimates for mixed methods for second order elliptic equations*, Math. Comp., 44 (1985), pp. 39–52.

[38] I. Faille, A. Fumagalli, J. Jaffré, and J. E. Roberts, *Model reduction and discretization using hybrid finite volumes for flow in porous media containing faults*, Computational Geosciences, 20 (2016), pp. 317–339.

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

[40] L. Formaggia, A. Fumagalli, A. Scotti, and P. Ruffo, *A reduced model for Darcy's problem in networks of fractures*, ESAIM: Mathematical Modelling and Numerical Analysis, 48 (2014), pp. 1089–1116.

[41] Formaggia, Luca, Scotti, Anna, and Sottocasa, Federica, *Analysis of a mimetic finite difference approximation of flows in fractured porous media*, ESAIM: M2AN, 52 (2018), pp. 595–630.

[42] N. Frih, V. Martin, J. E. Roberts, and A. Saâda, *Modeling fractures as interfaces with nonmatching grids*, Computational Geosciences, 16 (2012), pp. 1043–1060.

[43] A. Fumagalli and E. Keilegavlen, *Dual virtual element methods for discrete fracture matrix models*, arXiv preprint arXiv:1711.01818, (2017).

[44] A. Fumagalli and E. Keilegavlen, *Dual virtual element method for discrete fractures networks*, SIAM J. Scientific Computing, 40 (2018).

[45] A. Fumagalli, L. Pasquale, S. Zonca, and S. Micheletti, *An upscaling procedure for fractured reservoirs with embedded grids*, Water Resources Research, 52 (2015), pp. 6506–6525.

[46] A. Fumagalli and A. Scotti, *A numerical method for two-phase flow in fractured porous media with non-matching grids*, Advances in Water Resources, 62 (2013), pp. 454 – 464.

[47] M. Gentile, A. Sommariva, and M. Vianello, *Polynomial interpolation and cubature over polygons*, Journal of Computational and Applied Mathematics, 235 (2011), pp. 5232 – 5239.

[48] K. Lipnikov, G. Manzini, and M. Shashkov, *Mimetic finite difference method*, Journal of Computational Physics, 257 (2014), pp. 1163–1227.

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

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

[51] B. Nœtinger, *A quasi steady state method for solving transient Darcy flow in complex 3D fractured networks accounting for matrix to fracture flow*, J. Comput. Phys., 283 (2015), pp. 205–223.

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

[53] J. M. Nordbotten, W. M. Boon, A. Fumagalli, E. Keilegavlen, and R. Helmig, *Unified approach to discretization of flow in fractured porous media*, 2018.

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

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

[56] V. Reichenberger, H. Jakobs, P. Bastian, and R. Helmig, *A mixed-dimensional finite volume method for two-phase flow in fractured porous media*, Advances in Water Resources, 29 (2006), pp. 1020 – 1036.

[57] C. Rycroft, *Voro++: A three-dimensional voronoi cell library in c++*, (2009).

[58] T. Sandve, I. Berre, and J. Nordbotten, *An efficient multi-point flux approximation method for discrete fracture–matrix simulations*, Journal of Computational Physics, 231 (2012), pp. 3784 – 3800.

[59] N. Schwenck, B. Flemisch, R. Helmig, and B. I. Wohlmuth, *Dimensionally reduced flow models in fractured porous media: crossings and boundaries*, Computational Geosciences, 19 (2015), pp. 1219–1230.

[60] 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.