A posteriori error estimation and modeling of unsaturated flow in fractured porous media

Jhabriel Daniel Varela Estigarribia

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2022

UNIVERSITY OF BERGEN

A posteriori error estimation and modeling of unsaturated flow in fractured porous media

Jhabriel Daniel Varela Estigarribia

Thesis for the degree of Philosophiae Doctor (PhD)
at the University of Bergen

Date of defense: 17.06.2022

Year: 2022

Title: A posteriori error estimation and modeling of unsaturated flow in fractured porous media

Name: Jhabriel Daniel Varela Estigarribia

Print: Skipnes Kommunikasjon / University of Bergen

Preface

This dissertation is submitted as a partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) at the University of Bergen (UiB). The advisory committee consisted of Eirik Keilegavlen (UiB), Jan Martin Nordbotten (UiB), and Florin Adrian Radu (UiB).

The PhD project has been financially supported by VISTA, a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Equinor.

Acknowledgments

First and foremost, I would like to thank my supervisors Eirik Keilegavlen, Jan Martin Nordbotten, and Florin Adrian Radu for these three awesome years (five years including the Masters in the case of Eirik and Jan). I was immensely lucky to have you as advisers, and I will always be extremely thankful for that. Even when on occasions I doubt it myself, you still believed in me and, time and again, that kept me going. If I ever got half as good as you in advising, I would consider myself a great tutor.

This journey had many ups and downs, specially considering my engineering background and the ambitious goals we had in terms of working with “formal” mathematics. I cannot even begin to imagine how much patient this process of “let us try to turn this chemical engineer into a mathematician” must have demanded from my supervisors. So, once again, Eirik, Jan, and Florin, let me say that I am extremely humbled for all the resiliency and patience you have invested in me. Do not get me wrong, I am far from being a mathematician, but I think we can all agree that not many engineers can discuss the mixed-dimensional Poincaré inequality.

I would like to thank my wife **Patricia** next (yes, bold is necessary). To put it simply, I would not have done it without you. Thank you for remaining strong when you were in Paraguay, and I was in Norway. Thank you for being with me all those nights when I was trying to understand a proof or my code simply would not work. Thank you for always telling me that everything will eventually work out, even though I was hopeless. I was incredibly lucky to found you, and I am super excited to build a family with you.

I also thank the Porous Media Group (PMG) in general. One cannot simply ask for a better group to pursue a Ph.D. degree. Full of incredible academics, but more importantly, good people (and I mean *good* in the purest sense of the word). A group where you will always find someone offering help when you are stuck, or simply a coffee (*or a beer*) to clear (*or blurry*) your mind. But in all seriousness, PMG is a place where you can really feel this selfless sense of camaraderie, which is not easily taken for granted in most competitive environments.

Now is the time for my friends, both in Paraguay and in Norway. Here, I want to first mention David Landa. I met David in a NUPUS trip in 2017 and we “clicked” almost immediately. He visited me in Paraguay, we lived in Fantoft for several years and for almost a year we were roommates. As the awesome friend he is, he helped me numerous times in many aspects of my life, and for that, I am very grateful. Gracias mi amigo!. Continuing with my Fantoft friends, I would also like to thank Arooj, Dennis,

Shifat, Fahim, Robert, Isa, Birk, Tor, Sonia, and Penne, for the many barbecues, parties, and gatherings we enjoyed together. To those belonging to my work environment, let me also thank Erlend, Elyes, Alessio, Wietse, Davide, Runar, Ingeborg, Mats, Max, Ana, Michael, Anita, Sauenn, Hau, Ivar, Jakub, Omar, Veljko, and Shin. In the Paraguayan side, I would like to thank Renato, Fabi, Gustavo, Adri, Enzo, Hugui, Anto, Jimmy, Vinchi, Nelly, Eli, Ale, José, and Nubia.

Finally, I would like to thank my family: my dad Nelson, my mom María Mercedes, and my brother Adriel. Even if we are on different continents, you were (and are) always there for me. Always. Trough thick and thin.

Abstract

This doctoral thesis focuses on three topics: (1) modeling of unsaturated flow in fractured porous media, (2) *a posteriori* error estimation for mixed-dimensional elliptic equations, and (3) contributions to open-source software for complex multiphysics processes in porous media.

In our first contribution, following a Discrete-Fracture Matrix (DFM) approach, we propose a model where Richards' equation governs the water flow in the matrix, whereas fractures are represented as lower-dimensional open channels, naturally providing a capillary barrier to the water flow. Therefore, water in the matrix is only allowed to imbibe the fracture if the capillary barrier is overcome. When this occurs, we assume that the water inside the fracture flows downwards without resistance and, therefore, is instantaneously at hydrostatic equilibrium. This assumption can be justifiable for fractures with sufficiently large apertures, where capillary forces play no role. Mathematically, our model can be classified as a coupled PDE-ODE system of equations with variational inequalities, in which each fracture is considered a potential seepage face.

Our second contribution deals with error estimation for mixed-dimensional (mD) elliptic equations, which, in particular, model single-phase flow in fractured porous media. Here, based on the theory of functional *a posteriori* error estimates, we derive guaranteed upper bounds for the mD primal and mD dual variables, and two-sided bounds for the mD primal-dual pair. Moreover, we improve the standard results of the functional approach by proposing four ways of estimating the residual errors based on the conservation properties of the approximations, that is, (1) no conservation, (2) subdomain conservation, (3) local conservation, and (4) pointwise conservation. This results in sharper and fully-computable bounds when mass is conserved either locally or exactly. To our knowledge, to date, no error estimates have been available for fracture networks, including fracture intersections and floating subdomains.

Our last contribution is related to the development of open-source software. First, we present the implementation of a new multipoint finite-volume-based module for unsaturated poroelasticity, compatible with the Matlab Reservoir Simulation Toolbox (MRST). Second, we present a new Python-based simulation framework for multiphysics processes in fractured porous media, named PorePy. PorePy, by design, is particularly well-suited for handling mixed-dimensional geometries, and thus optimal for DFM models. The first two contributions discussed above were implemented in PorePy.

Abstrakt

Denne avhandlingen tar for seg tre emner: (1) modellering av flyt i umettet porøst medium med sprekker, (2) a posteriori feilestimater for blandet-dimensjonale elliptiske ligninger, og (3) bidrag til åpen kildekode for komplekse multifysikk-prosesser i porøse medier.

I det første bidraget anvender vi en Discrete-Fracture Matrix (DFM) (Diskret-Sprekk Matrise) metode til å sette opp en modell hvor Richard´s ligning modellerer vannflyt i matrisen, og sprekkene representeres som lavere-dimensjonale åpne kanaler, som naturlig virker som kapillærbarrierer til vann-flyten. Derfor vil vann i matrisen kun få tilgang til sprekken når kapillærbarrieren blir brutt. Når det inntreffer, antar vi at vannet i sprekken flyter nedover uten motstand, og at hydrostatisk ekvilibrium derfor inntreffer øyeblikkelig. Slike antakelser kan rettferdiggjøres for sprekker med tilstrekkelig stor apertur (åpning), hvor kapillærkrefter ikke har noen innvirkning. Fra et matematisk standpunkt kan modellen klassifiseres som en sammenkoblet PDE-ODE med variasjonelle ulikheter hvor hver sprekk behandles som en filtreringsfase.

Det andre bidraget tar for seg feilestimater for blandet-dimensjonale elliptiske ligninger, som modellerer en-fase flyt i porøse medier med sprekker. Her anvender vi teorien for “funksjonal a posteriori feilestimater” til å finne øvre skranker for primær og dual variablene, samt øvre og nedre skranker for primær-dual paret. Dessuten viser vi at vi kan forbedre standardresultatene fra “funksjonal a posteriori feilestimater” ved å foreslå fire måte å estimere residualfeilen basert på bevaringsegenskapene til diskretiseringen. De fire forskjellige bevaringsegenskapene er; ingen bevaringsegenskap, underdomene bevaring, lokal bevaring og punktvis bevaring. Dette fører til skarpere skranker som er mulige å beregne når masse er bevart enten lokalt, eller eksakt. Vi kjenner ikke til andre tilgjengelige feilestimater for sprekknettverk som inkluderer snitt av sprekker og sprekkrender som ligger innenfor domenets rand.

Det siste bidraget omhandler utvikling av åpen kildekode. Først presenterer vi implementeringen av en multipunktfluks-basert modul for flyt i umettet deformerbart porøst medium som er kompatibelt med “Matlab Reservoir Simulation Toolbox” (MRST). I tillegg presenterer vi et nytt Python-basert rammeverk for simulering av multifysikkprosesser i porøse medier med sprekker, som heter PorePy. Dette rammeverket er designet for å håndtere geometrier med blandede dimensjoner og er derfor optimalt for DFM modeller. De to første bidragene i avhandlingen (nevnt over) er implementert i PorePy.

Outline

This work is made up of two parts. The first part provides the mathematical background for the thesis, whereas the second contains the scientific production.

Part I, containing five chapters, is organized as follows. Chapter 1 provides an overview of the thesis. We focus on the available current models and error estimation techniques, analyze their limitations, and motivate the need for our contributions. In Chapter 2, we introduce standard mathematical models describing flow and deformation processes in nonfractured porous media. In Chapter 3, we extend the flow equations presented in the preceding chapter to account for fractures in the domain. Chapter 4 deals with the fundamentals of *a posteriori* error techniques of the functional type. Finally, in Chapter 5, we present a summary of the papers and propose further extensions of the current work.

Part II contains the scientific results, consisting of the following four papers:

Paper A Varela, J., Gasda, S. E., Keilegavlen, E., and Nordbotten, J. M. (2021). A Finite-Volume-Based Module for Unsaturated Poroelasticity. In K. Lie and O. Møyner (editors), *Advanced Modeling with the MATLAB Reservoir Simulation Toolbox*, 515–548. Cambridge: Cambridge University Press. doi: 10.1017/9781009019781.019.

Paper B Keilegavlen E., Berge R., Fumagalli A., Starnoni M., Stefansson I., Varela J., Berre I. (2021). Porepy: an open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences* **25**(**1**), 243–265. doi: 10.1007/s10596-020-10002-5.

Paper C Varela J., Ahmed E., Keilegavlen E., Nordbotten J. M., Radu, F. A. (2021). *A posteriori* error estimates for hierarchical mixed-dimensional elliptic equations. *Submitted to the Journal of Numerical Mathematics*.

Paper D Varela J., Keilegavlen E., Nordbotten J. M., Radu, F. A. (2022). A model of unsaturated flow in the presence of fractures acting as capillary barriers. *In preparation*.

Contents

Part I

Scientific Background

Chapter 1

Introduction

This thesis deals with modeling of unsaturated flow and error estimation. The first topic refers to the derivation of a system of differential equations describing the simultaneous flow of water and air in a porous medium. The second topic refers to the mathematical procedure in which the difference between the exact solution to a mathematical problem and some given function (most of the time derived from approximations to the model problem) is estimated. Both topics are highly relevant to numerous applications, ranging from nuclear waste storage to the design of efficient simulators.

Understanding how water flows in unsaturated systems (see Figure 2.2) is important for several processes such as groundwater management, aquifer remediation, agriculture, evaporation and evapotranspiration, transport of nutrients, nuclear waster storage, design and operation of dams, structural analysis, and the mining industry [137]. The flow in partially saturated media is often described by Richards' equation [114], which is a simplification of the two-phase flow equations [33] based on the assumption of inviscid air [90, 91]. To this date, Richards' equation remains one of the most challenging equations to solve [45]. In fact, a great deal of effort has been invested to provide robust, stable, economical and mass-conservative solutions [6, 31, 56, 60, 72, 94, 123, 143]. Many of these topics are still a matter of ongoing research.

Although unsaturated systems have been extensively studied in the case of non-fractured domains, the unsaturated flow in fractured porous media is far less mature. The study of unsaturated fractured systems started to receive more attention in the early 80s, when the U.S. Department of Energy considered the possibility of storing spent nuclear fuel and high-level radioactive waste in the Yucca mountain, Nevada. The Yucca mountain is made up of alternating layers of highly fractured anisotropic volcanic tuff [119]. The fact that there is an aquifer below the intended storage repository [46], raised concerns for the potential leakage of radioactive waste into the groundwater system. This motivated the development of new conceptual models to better understand the dynamics of unsaturated flow in the presence of fractures [138].

Models for fractured flow can generally be classified into two groups [16, 48]: (1) equivalent continuum models, and (2) models representing the fractures explicitly. In the first group, matrix and fractures are represented either as a single continuum, or as different continua coupled via transfer terms. In the second group, fractures are represented explicitly as separate lower-dimensional geometric objects embedded in the matrix. The link between matrix and fractures takes place via coupling conditions. This thesis employs the second approach. The reader is referred to Section 3.1 and the references therein for a more detailed discussion about the two approaches.

In the context of unsaturated fractured flow, examples of continuum-type models include: dual-porosity models [28, 65], dual-permeability models [62, 116], and more general multiscale multi-continua models [120]. Models with explicit representation of fractures, such as Discrete Fracture Matrix (DFM) models, were also successfully derived and applied. In [122], Therrien and Sudicky proposed a DFM model based on Richards' equation in the matrix and an extension of the cubic law to the partially saturated case in the fracture. Applications of unsaturated flow and transport, including fractured rocks in the context of aquifer pollution [77] and mining [2] were later based on this model. More recently, [69] employed a DFM model for saturated-unsaturated seepage analysis in the presence of fractures, and [61] proposed a DFM model in which Richard's equation is used in both the matrix and the fractures, and the resulting set of equations is solved using mixed hybrid finite element methods in space, a high-order integration scheme in time, and a novel mass lumping technique.

Although proven useful, the above-mentioned DFM models were derived as extensions of the saturated case, and as such, they lacked a formal mathematical ground. This issue was recently circumvented in [71] and [64], where the authors, using rigorous and formal upscaling, respectively, derived a catalog of reduced models depending on the porosity and permeability ratios between the fracture and the matrix.

Indeed, the previously mentioned DMF models correspond to the case where fractures are capable of storing and conducting water at comparable time scales as in the matrix. In this work, however, we are interested in a less studied type of setting, which takes place when the fracture permeability is so large compared to the matrix that capillary effects are essentially absent. This allows us to assume that water moves downwards at infinite speed inside the fracture, and hydrostatic equilibrium is achieved instantaneously. However, we remark that this assumption is only valid for fractures with sufficiently large apertures where capillary forces play no role.

In this context, our representation of fractures in unsaturated systems consists of idealized open channels. The fractures will then act as capillary barriers, diverging the flow of water in the matrix tangentially to the fracture. Only when near-saturation conditions at the fracture-matrix interface are achieved (and the capillary barrier is overcome) is water allowed to imbibe the fracture. Depending on the wetting conditions, fractures can then potentially serve as zero-resistance flow paths.

Mathematically, the previously described system can be classified as a coupled PDE-ODE system with variational inequalities. The PDE part corresponds to Richards' equation in the matrix, the ODE part to a water volume balance in the fractures, and the variational inequalities are included to model the conditional imbibition and drainage scenarios, where each fracture can be seen as a potential seepage face. Naturally, solving these types of systems is non-trivial, primarily due to the large gradients in heads and the lost of continuity of the water phase in the matrix/fracture interface. Paper D presents this model, including its generalization to fracture networks and solution strategies.

In general, the vast majority of mathematical models cannot be solved exactly, and therefore one has to resort to numerical approximations. Currently, a large array of numerical methods are available, e.g.: Finite Difference Method (FDM) [37, 67], Finite Element Method (FEM) [25, 27, 41], Mimetic Finite Difference Method (MFDM) [8, 36], Virtual Element Method (VEM) [13], Mixed Finite Element Method (MFEM) [21, 52], Mixed Virtual Element Method (MVEM) [50, 51], Discontinuous Galerkin (DG) [38, 115], Cell-Centered Finite Volume Method (CCFVM) [1, 58, 81, 82], just to name some of the most popular discretization techniques in porous media.

Provided an approximate solution to a model is available, two natural questions arise [136]: (1) How close is our approximation to the *true* solution? (2) Where are the errors localized? If we count with the exact solution, this can be easily accomplished by measuring the difference between the exact and the approximate solutions in a suitable norm —-in this case, however, we would not need an approximation in the first place. The more realistic situation corresponds to the case where the exact solution is not available and one can only hope for an estimation of the error.

The error between the exact and approximate solutions can be estimated *a priori* or *a posteriori*. The former provides information about the *rate* at which the error will decay, and as such it can be known before obtaining the approximate solution. The latter, which is the one considered in this thesis, is employed after the approximation has been obtained, and aims at providing a quantitative measure of the difference between the approximate and the exact solutions in terms of known quantities, e.g., the approximations themselves, the domain of interest, material properties, external sources, etc. Indeed, a good estimator should [136]: (1) be directly computable from the approximate solution, (2) go to zero as the computational effort increases, (3) be independent of the material parameters, and (4) be computationally inexpensive.

In this thesis, we are interested in derive *a posteriori* error estimates for approximations of the set of equations that model the incompressible single-phase flow in fractured porous media [23, 83]. More generally, we are interested in the types of models that can be represented as hierarchical mixed-dimensional elliptic equations [22], where matrices, fractures, intersections between fractures, and intersections between intersections can be hierarchically represented as different geometric objects of codimension one. For the 3d case, this corresponds to a matrix composed of simply connected 3d domains, fractures are simply connected planar 2d surfaces, the intersection between these fractures are 1d segments, and the intersection between fracture intersections are 0d points (see, e.g., Figure 3.3 for a two-dimensional example).

Although *a posteriori* error techniques for mono-dimensional domains are well-established (see the introductory part of Chapter 4 and the references therein), error bounds for mixed-dimensional geometries are far more scarce. Indeed, the ones available exist in the context of mortar/multiscale methods [14, 89, 140, 142] and fractured porous media [32, 53, 78] with far less geometric generality than the ones presented here.

The lack of *a posteriori* error bounds for mixed-dimensional elliptic problems is due to the inherent difficulty associated with working with several subdomains, some of them intersecting and some potentially floating. Indeed, deriving such bounds can rapidly become cumbersome and highly technical. These issues can be circumvented by deriving the bounds based on the theory of functional *a posteriori* error estimates [105] (see also Chapter 4) together with the introduction of a compact mixed-dimensional notation [22].

Although the bounds obtained with standard functional *a posteriori* error estimates provide guaranteed upper bounds, they require constants (in particular, the mixed-dimensional Poincaré constant) that are often hard to determine. To avoid such constrain, we exploit the fact that Poincaré-type inequalities imply weighted norms [88, 95], unlocking the possibility to derive sharper bounds when mass is conserved at the subdomain-level, grid-level, or exactly. We refer to Paper C for further details.

Ultimately, approximations to mathematical models, as well as *a posteriori* error bounds, need to be implemented in a computational framework. As science advanced and more complex processes required to be analyzed using simulation tools, the demand for more robust simulation frameworks also increased.

This is the case of the Matlab Reservoir Simulation Toolbox (MRST) [70] and the Python-based multiphysics simulation framework PorePy [57]. Both programs are open source, rapid prototyping-oriented alternatives for simulating complex processes in porous media. MRST is primarily oriented towards multiphase multicomponent processes on monodimensional domains, whereas PorePy is particularly well-suited for DFM-type processes on mixed-dimensional geometries.

The fact that both software are designed as rapid prototyping tools allows the possibility of extending existing models with relatively little effort. Two aspects are key to achieve this: (1) access to standard discretization techniques and (2) an automatic differentiation (AD) framework.

Indeed, this thesis greatly benefits from the rapid-prototyping capabilities of MRST and PorePy. In particular, we can mention the following instances:

Paper A: Extension of the equations of linear poroelasticity (Section 2.3) to the nonlinear unsaturated case (Section 2.4) based on multipoint finite volume discretizations.

Paper C: Possibility of effortlessly validating the derived error bounds for three families of locally mass-conservative approximations; MFEM, MVEM, and CCFVM.

Paper D: Extension of the implemented monodimensional Richards' equation from Paper A, to the fractured case, accounting for the ODE coupling.

We remark that all implemented models (including validations and numerical examples) obtained as a result of this thesis are completely open source. The reader is referred to Section 1.1 for the links to the respective repositories.

1.1 Main contributions

In the following, we summarize the main contributions of this doctoral thesis:

Mathematical modeling: A new model for unsaturated fractured flow has been proposed. Contrary to standard models where Richards' equation is used to model both the flow in the matrix and in the fractures, we consider the fractures to be open channels. The fractures act as capillary barriers for the water flow in the matrix, which is only allowed to imbibe the fractures if such barrier is overcome. The resulting model is a coupled PDE-ODE system where each fracture represents a potential seepage face. The reader is referred to Paper D for further details. The implementation of the model, together with conceptual simulations are available through `unsat-frac`, a PorePy extension package available at `https://github.com/jhabriel/unsat-frac`.

***A posteriori* error estimation:** *A posteriori* error estimates for mixed-dimensional elliptic equations were derived. In particular, our bounds can be applied to approximations to the set of equations modeling the incompressible single-phase flow in fracture networks. To the best of our knowledge, error bounds were only available for a single fracture embedded in a matrix (or various nonintersecting fractures) or approximations to mortar-type models with considerable less geometric generality compared to what we have obtained here. Thus, until now, error bounds for generic fracture networks (including intersecting fractures and floating subdomains) have essentially not been available. The reader is referred to Paper C for further details. We also refer to the PorePy extension package `mdestimates`, available at `https://github.com/jhabriel/mixdim-estimates`.

Open-source software for complex processes: `fv-unsat`, an open-source multipoint finite-volume-based solver for unsaturated flow and unsaturated poroelasticity compatible with MRST was developed. Paper A contains the complete description of the module. The software is available as an add-on module to MRST, see also `https://github.com/pmgbergen/fv-unsat`.

Moreover, we proposed a new Python-based simulation framework for multiphysics problems in fractured porous media named PorePy; see Paper B and `https://github.com/pmgbergen/porepy`. Since Paper B is a multi-authored effort, we would like to emphasize the specific contributions of this thesis: (1) implementation and writing of the validation of the poroelastic code for Mandel's problem (Section 5.2), (2) generation of figures in IPE [118], (3) generation of code snippets using the LaTeX package `listings` [54], and (4) writing and coordination of the electronic supplementary material.

Chapter 2

Flow and deformation in non-fractured porous media

This chapter is devoted to introducing the governing equations in non-fractured domains. We start with the description of the flow equations for single-phase and unsaturated systems and later incorporate deformation effects. The models presented in this chapter are particularly relevant for Paper A and Section 5.2 of Paper B. However, the reader will find that the models from Section 2.1 and Section 2.2 are later extended in Chapter 3, and that the exposition from Chapter 4 is based on the model from Section 2.1.

Throughout this chapter, we consider a single open bounded domain $\Omega \subset \mathbb{R}^n$, $n \in \{2, 3\}$, with a smooth boundary $\partial\Omega$. We let $\mathbf{x} = [x_1, \dots, x_n] \in \Omega$ denote the vector of spatial coordinates, with $z = x_n$ denoting the vertical coordinate considered positive, pointing upwards. For time-dependent problems, we let t represent the time variable, $T > 0$ the final time, and $(0, T)$ the time interval of interest.

Our exposition will not consider derivations. The interested reader, however, is referred to [35, 117] and the references therein for further details.

2.1 Single phase flow

Let us start by stating the steady-state mass conservation equation for an incompressible fluid in a nondeformable porous medium [11]:

$$\nabla \cdot \left(\phi \mathbf{v}_{f,s}\right) = \frac{\dot{m}_f}{\rho_f}, \qquad \text{in } \Omega. \tag{2.1}$$

Here, ϕ is the porosity of the porous medium, ρ_f is the density of the fluid, $\mathbf{v}_{f,s}$ is the velocity of the fluid relative to the solid particles, and $\dot{m}_f$ is the external rate of addition or subtraction of the fluid mass.

The product between ϕ and $\mathbf{v}_{f,s}$ is given by Darcy's law, which, in the absence of

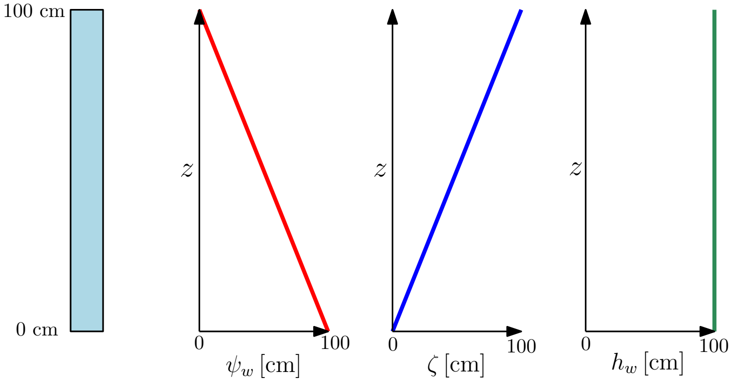


Figure 2.1: Pressure, elevation, and hydraulic heads for a saturated water column in hydrostatic equilibrium.

inertial effects, is given by

$$\mathbf{q}_f = \phi \mathbf{v}_{f,s} = -\frac{\mathbf{k}_s}{\mu_f}(\nabla p_f - \rho_f \mathbf{g}), \qquad \text{in } \Omega, \tag{2.2}$$

where $\mathbf{q}_f$ is the Darcy velocity or specific discharge of the fluid f, $\mathbf{k}_s$ is a symmetric positive-definite $n \times n$ tensor denoting the intrinsic permeability of the solid particles, μ_f is the fluid dynamic viscosity, ∇p_f is the gradient of the fluid pressure, and $\mathbf{g}$ is the gravitational acceleration.

In hydrogeology, it is more common to express Darcy's law (2.2) in terms of the gradients of the hydraulic heads. The hydraulic head (or total head) h_f is given by

$$h_f = \int_{p_{ref}}^{p_f} \frac{dp'}{\rho_f g} - \int_{z_{ref}}^{z} \frac{\mathbf{g} \cdot \mathbf{e}_n}{g} dz', \tag{2.3}$$

where p_{ref} and z_{ref} are reference values of p_f and z, and $\mathbf{e}_n$ is the unit vector in the x_n-axis [90].

Note that the density of the fluid ρ_f in (2.3) could depend on the pressure of the fluid, the temperature of the fluid and even the concentration of chemical species [91]. Even though the definition of h_f still makes perfect sense in this more general case, in practice, one only employs (2.3) when the first integral can be computed easily. In the particular case where ρ_f is constant, we have

$$h_f = \frac{p_f - p_{ref}}{\rho_f g} + (z - z_{ref}), \tag{2.4}$$

where we used $\mathbf{g} \cdot \mathbf{e}_n = -g$. The first term of (2.4) is often referred to as the pressure head ψ_f, while the second term is called the elevation head ζ. The usual practice is to set the reference pressure as the atmospheric pressure $p_{ref} = p_{atm}$ and $z_{ref} = \min_{\mathbf{x} \in \Omega} \mathbf{x} \cdot \mathbf{e}_n$.

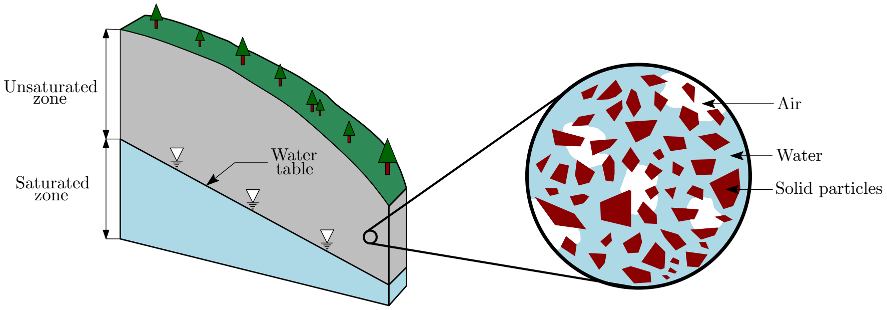


Figure 2.2: The unsaturated zone. A zoom at the pore-scale shows the three phases involved in an unsaturated system; the solid phase (solid particles) and the fluid phases (water and air).

By properly computing the derivatives of h_f [91], it is straightforward to see that Darcy's law (2.2), in the incompressible case, can be written equivalently as

$$\mathbf{q}_f = -\mathbf{K}_f \nabla h_f, \qquad \text{in } \Omega, \tag{2.5}$$

where $\mathbf{K}_f = \mathbf{k}_s \rho_f g / \mu_f$ is the hydraulic conductivity of the porous medium.

2.2 Unsaturated flow

Unsaturated flow refers to the simultaneous flow of water w and air a in a porous medium. Here, we employ the usual assumption of inviscid air. This assumption establishes that, based on the contrast in physical properties between the fluid phases (i.e., the air being three orders of magnitude less dense and two orders of magnitude less viscous than water at normal conditions), the air phase flows without any of resistance in the soil [90].

This assumption and the fact that the air in the unsaturated zone is connected to the atmosphere justifies setting the air pressure equal to the atmospheric pressure, $p_a = p_{atm}$.

A statement of mass conservation for water, assuming both water and solid particles as incompressible, can be written as

$$\frac{\partial(\phi S_w)}{\partial t} + \nabla \cdot (\phi S_w \mathbf{v}_{w,s}) = \frac{\dot{m}_f}{\rho_f}, \qquad \text{in } \Omega \times (0, T), \tag{2.6}$$

where S_w is the water saturation. The product ϕS_w is a measure of the amount of water in the void spaces of the porous medium, and therefore receives the name of water content θ_w. The quantity $\phi S_w \mathbf{v}_{w,s}$ is now given by the extended, multiphase version of Darcy's law

$$\mathbf{q}_w = \phi S_w \mathbf{v}_{w,s} = -\frac{\mathbf{k}_s k_{rw}}{\mu_w} \left(\nabla p_w - \rho_w \mathbf{g} \right), \qquad \text{in } \Omega \times (0, T), \tag{2.7}$$

where $k_{rw} \in [0,1]$ is the relative permeability of the water, incorporating the effects of the reduction in the effective pore space due to the presence of multiple phases at the pore-scale.

Analogously to the single-phase case, (2.7) can also be written in terms of hydraulic heads

$$\mathbf{q}_w = -\mathbf{K}_w k_{rw} \nabla h_w, \qquad \text{in } \Omega \times (0,T), \tag{2.8}$$

Substitution of (2.8) into (2.6) gives the well-known n-dimensional Richards' equation in mixed-form

$$\frac{\partial \theta_w}{\partial t} - \nabla \cdot \left[\mathbf{K}_w k_{rw} \nabla (\psi_w + \zeta) \right] = \frac{\dot{m}_w}{\rho_w}, \qquad \text{in } \Omega \times (0,T). \tag{2.9}$$

Although slightly confusing for those familiar with mixed-finite element methods, in this context, the name "mixed-form" simply reflects the fact that both the water content θ_w and the pressure head ψ_w appear explicitly in equation (2.9).

2.3 Biot's equations of poroelasticity

So far, we have only considered the flow of fluids under incompressible conditions. However, in some processes, it is important to account for the possibility that the porous medium will deform. Clear examples when mechanical effects must be included are the ones involving large external loads, such as the consolidation or the subsidence of soils.

Maurice A. Biot [20] proposed a general framework for representing the flow of a fluid accounting for the deformation of the porous medium assuming linear elastic properties of the solid particles. Since then, Biot's equations were extensively applied in numerous fields, including but not limited to geomechanics, tissue mechanics, material engineering, and hydrogeology.

Let us start by introducing the mechanical equations. The linear momentum conservation equation, under quasi-static conditions, is given by

$$\nabla \cdot \sigma - \left[(1-\phi)\rho_s + \phi \rho_w \right] \mathbf{g} = 0, \qquad \text{in } \Omega \times (0,T), \tag{2.10}$$

where σ is the total stress tensor and the second term denotes the body forces. For poroelastic systems, the total stress is given by the principle of effective stress

$$\sigma = \sigma_e - \alpha p_w \mathbf{I}, \qquad \text{in } \Omega \times (0,T), \tag{2.11}$$

where σ_e is the effective stress tensor, α is the Biot's coupling coefficient, and $\mathbf{I}$ is the identity tensor. Equation (2.11) states that the total stress in a poroelastic media has two contributions: (1) the effective part that causes the actual deformation of the medium, and (2) the fluid contribution. The negative sign in (2.11) follows from the convention that the tensile forces are positive and the compressive forces negative.

Substituting (2.11) into (2.10) gives the linear momentum balance equation

$$\nabla \cdot \sigma_e - \alpha \nabla p_w - \left[(1-\phi)\rho_s + \phi \rho_w \right] \mathbf{g} = 0, \qquad \text{in } \Omega \times (0,T). \tag{2.12}$$

Assuming small deformations and isotropic elastic properties for the solid, the effective stress σ_e can be related to the displacement field $\mathbf{u}$ via the generalized Hooke's law:

$$\sigma_e = \mu_s \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right) + \lambda_s (\nabla \cdot \mathbf{u})\, \mathbf{I}, \qquad \text{in } \Omega \times (0, T), \tag{2.13}$$

where λ_s and μ_s are the first and second Lamé parameters [74].

The so-called storage equation can be derived with the help of the mass conservation equations for the fluid and solid phases together with relevant constitutive relationships [68]

$$C_\varepsilon \frac{\partial p_w}{\partial t} + \alpha \frac{\partial}{\partial t} (\nabla \cdot \mathbf{u}) + \nabla \cdot \mathbf{q}_w = \frac{\dot{m}_w}{\rho_w}, \qquad \text{in } \Omega \times (0, T). \tag{2.14}$$

Here, $C_\varepsilon = (\alpha - \phi) C_s - \phi C_w$ is the storativity (a parameter depending on the compressibility of the solid particles C_s and water phase C_w) and the Darcy velocity $\mathbf{q}_w$ is given by (2.2)

Equation (2.14) suggests that there are four mechanisms that influence the flow of a fluid in a poroelastic medium: (1) compressibility effects, (2) deformation of the porous medium due to mechanical effects, (3) gradients in pressure and elevation, and (4) external sources or sinks of water.

2.4 Unsaturated poroelasticity

For some processes, such as operations of dams or soil desiccation, Biot's equations of poroelasticity are insufficient, since they are only valid for a single fluid phase saturating the entirety of the domain. Indeed, following [35, 68], it is possible to extend the equations of poroelasticity to account for the extra air phase.

The linear momentum balance and the principle of effective stress [49, 117] can be extended to the multiphase case

$$\nabla \cdot \sigma - \left[(1 - \phi)\rho_w + S_w \phi \rho_w\right] \mathbf{g} = 0, \qquad \text{in } \Omega \times (0, T), \tag{2.15}$$

$$\sigma = \sigma_e - \alpha S_w p_w \mathbf{I}, \qquad \text{in } \Omega \times (0, T), \tag{2.16}$$

where (2.15) and (2.16) differ respectively from (2.10) and (2.11) in that only a part of the total amount of fluid within the porous media contributes to the body forces and the total stress; this effect is incorporated via S_w in both equations.

Substitution of (2.16) into (2.15) results in the unsaturated momentum balance equation:

$$\nabla \cdot \sigma_e - \alpha \nabla \left(S_w p_w\right) - \left[(1 - \phi)\rho_s + S_w \phi \rho_w\right] \mathbf{g} = 0, \qquad \text{in } \Omega \times (0, T). \tag{2.17}$$

Note that since σ_e takes into account only the stresses associated with the solid skeleton, (2.13) is still valid in the unsaturated case.

Finally, we extend the storage equation (2.14) to obtain the unsaturated storage equation

$$C_{p,\varepsilon} \frac{\partial p_w}{\partial t} + C_{S,\varepsilon} \frac{\partial S_w}{\partial t} + \alpha S_w \frac{\partial}{\partial t} (\nabla \cdot \mathbf{u}) + \nabla \cdot \mathbf{q}_w = \frac{\dot{m}_w}{\rho_w}, \qquad \text{in } \Omega \times (0, T), \tag{2.18}$$

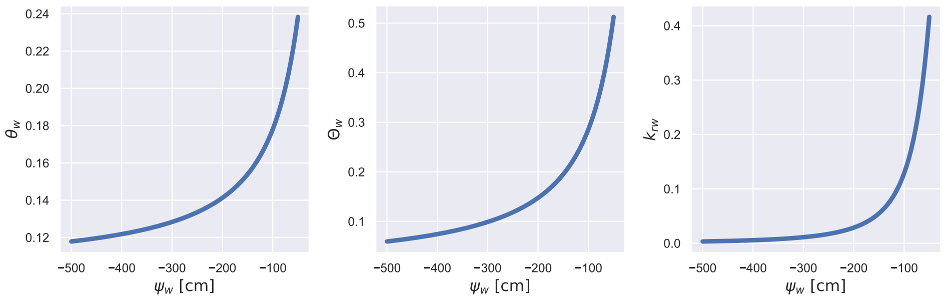


Figure 2.3: Soil-Water Retention Curves for a soil sample from New Mexico [90] with $\alpha_{vG} = 0.0335$ [1/cm], $n_{vG} = 2$, $m_{vG} = 0.5$, $\theta_{w,sat} = 0.368$, and $\theta_{w,res} = 0.102$.

where $C_{p,\varepsilon} = (\alpha - \phi)C_s S_w^2 - \phi C_w S_w$ and $C_{S,\varepsilon} = (\alpha - \phi)C_s S_w p_w + \phi$ are compressibility-like terms, and $\mathbf{q}_w$ is given by (2.7).

2.5 Soil-water retention curves

In unsaturated systems, we have to provide relations for θ_w (or S_w) and k_{rw}. This is usually achieved by means of Soil-Water Retention Curves (SWRC). The most widely used set of relations is the one proposed by van Genuchten and Mualem [79, 127]:

$$\theta_w(\psi_w) = \begin{cases} \theta_{w,res} + \left(\theta_{w,sat} - \theta_{w,\mathrm{res}}\right)\left[1 + \left(\alpha_{vG}|\psi_w|\right)^{n_{vG}}\right]^{-m_{vG}} & \psi_w < 0 \\ \theta_{w,sat} & \psi_w \geq 0 \end{cases}, \tag{2.19}$$

$$\Theta_w(\theta_w) = \frac{\theta_w - \theta_{w,res}}{\theta_{w,sat} - \theta_{w,res}}, \tag{2.20}$$

$$k_{rw}(\Theta_w) = \sqrt{\Theta_w}\left[1 - \left(1 - \Theta_w\right)^{\frac{1}{m_{vG}}}\right]^2. \tag{2.21}$$

Here, α_{vG}, n_{vG}, and m_{vG} are fitting parameters, $\theta_{w,res}$ is the residual water content, $\theta_{w,sat}$ is the water content at saturated conditions, and Θ_w is the effective saturation, i.e., the normalized water content. In Figure 2.3, we show the SWRC for a soil sample from New Mexico [90].

2.6 Boundary and initial conditions

Boundary and initial conditions have to be specified in order to close the previously presented system of equations. For a coupled flow-mechanical problem, the boundary conditions are given by

$$p_w = g_D^{\mathrm{f}}, \qquad \text{on } \partial_D^{\mathrm{f}}\Omega \times (0,T), \tag{2.22}$$

$$\mathbf{q}_w \cdot \mathbf{n} = g_N^{\mathrm{f}}, \qquad \text{on } \partial_D^{\mathrm{f}}\Omega \times (0,T), \tag{2.23}$$

$$\mathbf{u} = \mathbf{g}_D^{\mathrm{m}}, \qquad \text{on } \partial_D^{\mathrm{m}}\Omega \times (0,T), \tag{2.24}$$

$$\sigma \cdot \mathbf{n} = \mathbf{g}_N^{\mathrm{m}}, \qquad \text{on } \partial_N^{\mathrm{m}}\Omega \times (0,T), \tag{2.25}$$

where $\mathbf{n}$ denotes the normal vector facing outward on the corresponding Neumann part of the boundary [24]. We require the boundary partitions to satisfy $\overline{\partial_D^{\mathrm{f}}\Omega} \cup \overline{\partial_N^{\mathrm{f}}\Omega} = \overline{\partial_D^{\mathrm{m}}\Omega} \cup \overline{\partial_N^{\mathrm{m}}\Omega} = \partial\Omega$. Finally, the initial conditions are given by

$$p_w = g_0^{\mathrm{f}}, \qquad \text{in } \Omega \times \{0\}, \tag{2.26}$$

$$\mathbf{u} = \mathbf{g}_0^{\mathrm{m}}, \qquad \text{in } \Omega \times \{0\}. \tag{2.27}$$

where g_0^{f} and $\mathbf{g}_0^{\mathrm{m}}$ are the initial pressure and displacement distributions.

Naturally, for Richards' equation (Section 2.2) one only needs boundary conditions (2.22), (2.23), and the initial condition (2.26). For the incompressible single-phase flow model (Section 2.1) only boundary conditions have to provided, and it is required $\partial_D\Omega \neq \emptyset$ to ensure a unique solution.

Modeling evaporation processes in porous media often requires switching between boundary conditions (2.22) and (2.23) [93]. Indeed, the so-called atmospheric boundary condition imposes a Neumann boundary condition on the soil surface with the outflow set to the maximum evaporation rate E_{max} as long as the soil is sufficiently wet, otherwise, the boundary condition on the soil surface switches to a Dirichlet type, with the pressure value given by the minimum allowable pressure $p_{w,min}$ on the soil surface; which can be approximated by the psychometric conditions of the surroundings.

Let us denote the soil boundary $\partial_{top}\Omega = \{\mathbf{x} \in \partial\Omega \, : \, \max_{\mathbf{x}\in\partial\Omega} \mathbf{x} \cdot \mathbf{e}_n\} \subset \partial\Omega$. Then, the atmospheric boundary condition can be written as: On $\partial_{top}\Omega \times (0,T)$, set

$$\begin{cases} \mathbf{q}_w \cdot \mathbf{n} = E_{max}, & p_w > p_{w,min} \\ p_w = p_{w,min}, & \text{otherwise} \end{cases}. \tag{2.28}$$

The reader with an engineering background might recognize (2.28) as a system-dependent boundary condition, whereas a mathematician will classify (2.28) as a variational inequality. An interesting study case in which atmospheric boundary conditions are applied in the context of unsaturated poroelasticity can be found in Paper A.

Chapter 3

Flow in fractured porous media

In this chapter, we present the mathematical models for single-phase and unsaturated flow in fractured porous media. The former is relevant to Papers B and C, whereas the latter is relevant to Paper D.

We start with a brief discussion of the current conceptual models available for representing fractures. Then, we present the governing equations for the case of a single fracture embedded in a matrix. Extension to fracture networks are considered next, where we first introduce the mixed-dimensional decomposition of the domain of interest, and then present the models valid for fracture networks.

3.1 Representation of fractures and the different conceptual models

In the last decades, a great deal of effort have been invested in developing accurate models for representing fractures in porous media. This responds to the important role that fractures play in the extraction of oil and gas, geothermal engineering, deep nuclear waste storage, and CO_2 sequestration, just to name a few relevant applications.

The major decision that one has to make when including fractures in the model is whether to represent them implicitly or explicitly [16].

Within the first category, we find the single-continuum model [40, 73] and the multi-continuum model [9, 139]. In the former, fractures and the surrounding medium are represented as a single continuum, and their distinction is therefore achieved by assigning a different permeability where the fractures are located (including possibly orientation effects). The latter is based on the super-imposition of several mediums, each medium having its own conservation law and constitutive relationship. The most common multi-continuum model is the dual-continuum model, where the surrounding matrix and the fracture are different continua. The communication between these continua is achieved via transfer terms that enter the conservation equations; ultimately, the limitation of multi-continuum models is associated with the difficulty in obtaining such transfer terms.

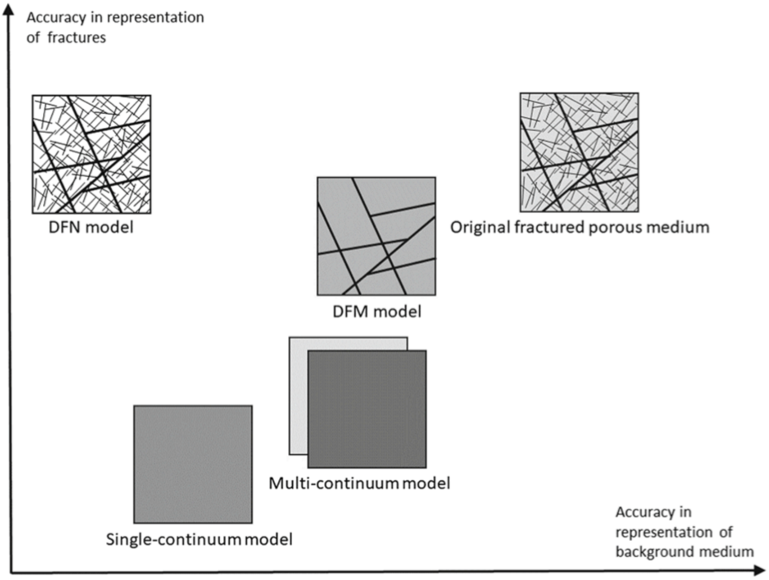

Accuracy in representation of fractures
DFN model
Original fractured porous medium
DFM model
Multi-continuum model
Single-continuum model
Accuracy in representation of background medium

Figure 3.1: Different conceptual models used for the representation of fractures. *Image borrowed from [16] under Creative Common License v.4.0.*

Unlike models that represent fractures implicitly, models that represent fractures explicitly establish a geometric distinction between the fractures and the surrounding matrix. Within this category, the two most popular models are the Discrete Fracture Network (DFN) [47] and Discrete Fracture Matrix (DFM) models [23, 76, 83]. The first represents only the fracture network and neglects the flow in the matrix (and thus effectively considers the matrix as impermeable) whereas the second considers both the flow in "relevant" fractures and in the matrix. To what extent a fracture might be considered relevant or not will depend on its size relative to the size of the matrix. In principle, all fractures could be represented explicitly, but this will not be feasible from a computational standpoint; the fractures that are not represented explicitly are generally upscaled and replaced by average quantities within the host medium [16].

Figure 3.1 from [16] presents the conceptual models and their accuracy in representing the host medium and the fractures. In this thesis, we are interested in representing the processes both in the matrix and in the fractures, thus, Papers B, C, and D employ the DFM paradigm.

3.2 Flow equations for a single fracture

Before introducing the model equations valid for generic fracture networks, we first present the simpler setting of a single fracture embedded in a matrix. We start with the equations describing the single-phase flow and then move to the unsaturated case.

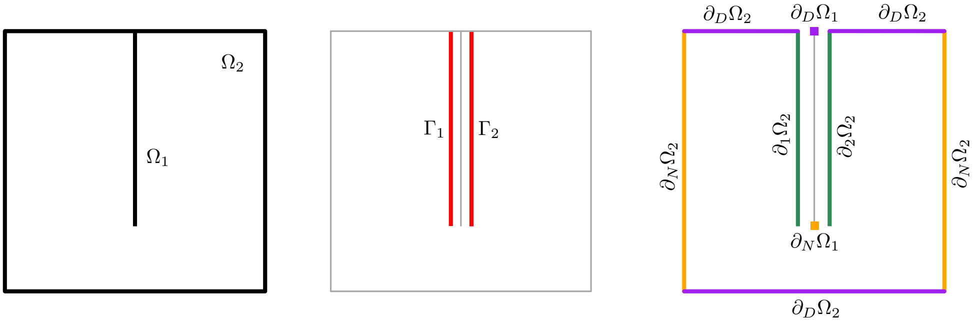


Figure 3.2: A one-dimensional fracture embedded in a two-dimensional matrix. Left: The matrix Ω_2 hosting the fracture Ω_1. Center: Interfaces Γ_1 and Γ_2 at both sides of Ω_1 establish the link with Ω_2. Right: External (Neumann and Dirichlet) and internal boundary conditions. Note that $\partial_1\Omega_2, \Gamma_1, \Omega_1, \Gamma_2$, and $\partial_2\Omega_2$ all coincide spatially. However, for illustrative purposes, they are placed in different locations.

3.2.1 Single-phase flow equations for a single fracture

Let us start by considering the case of a single one-dimensional fracture Ω_1 embedded in a two-dimensional matrix Ω_2, as shown in the left panel of Figure 3.2.

We let each side of Ω_1 be coupled with Ω_2 via the interfaces Γ_1 and Γ_2 (see the center Figure 3.2). The coupling takes place through the internal boundaries of Ω_2, namely $\partial_1\Omega_2$ and $\partial_2\Omega_2$; numbered such that the first sub-index corresponds to the interface number and the second to the matrix number (see right Figure 3.2). It is worth mentioning that, in physical space, $\partial_1\Omega_2 = \Gamma_1 = \Omega_1 = \Gamma_2 = \partial_2\Omega_2$. However, their distinction as separate entities will be key when we generalize the model to fracture networks.

We assume steady-state conditions and incompressibility of the solid and fluid phases. For the sake of compactness, we will drop the subscript denoting the water phase and define $f := \dot{m}/\rho$.

With these considerations, we present the governing equations for the matrix, fracture, and interfaces. We remark that the following set of equations are well-established in the literature [15, 23, 57, 76, 83]. In the matrix Ω_2, the governing equations are

$$\nabla_2 \cdot \mathbf{q}_2 = f_2, \qquad \text{in } \Omega_2, \tag{3.1}$$

$$\mathbf{q}_2 = -\mathbf{K}_2 \nabla_2 h_2, \qquad \text{in } \Omega_2, \tag{3.2}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = \lambda_1, \qquad \text{on } \partial_1\Omega_2, \tag{3.3}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = \lambda_2, \qquad \text{on } \partial_2\Omega_2, \tag{3.4}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = g_{N,2}, \qquad \text{on } \partial_N\Omega_2, \tag{3.5}$$

$$h_2 = g_{D,2}, \qquad \text{on } \partial_D\Omega_2. \tag{3.6}$$

We recognize equations (3.1) and (3.2) as the mass conservation equation and the single-phase Darcy's law written in terms of hydraulic head (see Section 2.1). In addition, equations (3.3) and (3.4) enforce the normal Darcy velocities on $\partial_1\Omega_2$ and $\partial_2\Omega_2$ to match the interface fluxes λ_1 and λ_2 (to be defined later), respectively. Note that the normal vector $\mathbf{n}_2$ on the internal boundaries, by construction, points from Ω_2 to Ω_1. Finally, equations (3.5) and (3.6) impose Neumann and Dirichlet boundary conditions, respectively.

In Ω_1, the governing equations are given by

$$\nabla_1 \cdot \mathbf{q}_1 - \left(\lambda_1 + \lambda_2\right) = f_1, \qquad \text{in}\, \Omega_1, \tag{3.7}$$

$$\mathbf{q}_1 = -\mathbf{K}_1 \nabla_1 h_1, \qquad \text{in}\, \Omega_1, \tag{3.8}$$

$$\mathbf{q}_1 \cdot \mathbf{n}_1 = g_{N,1}, \qquad \text{on}\, \partial_N \Omega_1, \tag{3.9}$$

$$h_1 = g_{D,1}, \qquad \text{on}\, \partial_D \Omega_1. \tag{3.10}$$

Equation (3.7) is the mass conservation equation in the fracture, where ∇_1 denotes the *del*-operator in the tangential direction and the term in parentheses represents the jump in interface fluxes onto Ω_1. The tangential velocity $\mathbf{q}_1$ is given by Darcy's law (3.8), and equations (3.9) and (3.10) establish the Neumann and Dirichlet boundary conditions at the bottom and top tips of the fracture.

The usual practice when the boundary of a fracture touches the ambient boundary is to inherit the ambient boundary value. On the other hand, on immersed tips, it is accustomed to prescribing null-flux, which is justifiable by the small mass transfer rate through the tips due to the large ratio between the fracture length and its aperture [7, 23].

The coupling between Ω_1 and Ω_2 takes place through the interface fluxes λ_1 and λ_2 on Γ_1 and Γ_2, respectively. Since interface fluxes act as coupling agents, they are also referred to as *mortar fluxes*. Following [76], we assume that mortar fluxes are linearly related to the jump in hydraulic heads, and thus follow a Darcy-type law

$$\lambda_1 = -\frac{2k_1}{a_1}\left(h_1 - h_2\right), \qquad \text{on}\, \Gamma_1, \tag{3.11}$$

$$\lambda_2 = -\frac{2k_2}{a_1}\left(h_1 - h_2\right), \qquad \text{on}\, \Gamma_2, \tag{3.12}$$

where k_1 and k_2 are the normal hydraulic conductivity on Γ_1 and Γ_2, and a_1 is the aperture of the fracture.

3.2.2 Unsaturated flow equations for a single fracture

Let us now consider the unsaturated flow in fractured porous medium. Here, we assume that the matrix can be represented as a porous media (typically some type of soil) whereas the fractures are represented as lower-dimensional thin open channels.

This allows us to describe the flow of water in the matrix using Richard's equation (see Section 2.2). However, since air-filled fractures provide a natural capillary barrier, water in the matrix can only break through the fracture if the capillary barrier is overcome. This typically occurs when saturated (or nearly saturated) conditions are present in the matrix-fracture interface.

Inside the fracture, the water travels downwards, primarily due to the action of gravity. And, since the fracture is an open channel, its velocity usually exceeds the velocity under saturated conditions in the matrix [121, 141]. Therefore, one can assume that water travels at infinite speed in the fracture, and that at any given time t, is in *instantaneous hydrostatic equilibrium*. By accepting this premise, the volumetric changes in the fracture are given by an ordinary differential equation (ODE). We note that this coupled PDE-ODE system of equations was recently obtained via formal and rigorous upscaling [64, 71].

In practice, the water inside the fracture travels at finite speed. The precise mechanism in which this occurs is not yet fully understood. However, there is experimental evidence that this takes place in the form of thin films [39, 124, 126]. It is important to mention that thanks to our mixed-dimensional decomposition (which treats matrix, fractures, and interfaces as different objects), one can, with relatively low effort, couple more complex models for the water distribution in the fractures. This is, however, beyond the scope of the current work.

In the following, we present the governing equations for a single vertical fracture Ω_1 embedded in a two-dimensional matrix Ω_2, as shown in the left Figure 3.2. Let us start by presenting the governing equations in the matrix Ω_2:

$$\frac{\partial \theta_2}{\partial t} + \nabla \cdot \mathbf{q}_2 = f_2, \qquad \text{in } \Omega_2 \times (0, T), \tag{3.13}$$

$$\mathbf{q}_2 = -\mathbf{K}_2\, k_{rw,2} \nabla h_2, \qquad \text{in } \Omega_2 \times (0, T), \tag{3.14}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = \lambda_1, \qquad \text{on } \partial_1 \Omega_2 \times (0, T), \tag{3.15}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = \lambda_2, \qquad \text{on } \partial_2 \Omega_2 \times (0, T), \tag{3.16}$$

$$\mathbf{q}_2 \cdot \mathbf{n}_2 = g_{N,2}, \qquad \text{on } \partial_N \Omega_2 \times (0, T), \tag{3.17}$$

$$h_2 = g_{D,2}, \qquad \text{on } \partial_D \Omega_2 \times (0, T), \tag{3.18}$$

$$h_2 = g_{0,2}, \qquad \text{in } \Omega_2 \ \times \{0\}. \tag{3.19}$$

Here, (3.13) and (3.14) are the mass conservation equation and multiphase Darcy's law, respectively (see also Section 2.2). Equations (3.15) and (3.16) are the internal boundary conditions, which, as in the preceding model, are required to match the mortar fluxes λ_1 and λ_2 (defined below). Equations (3.17) and (3.18) impose the usual Neumann and Dirichlet boundary conditions, and (3.19) the initial conditions. We use $\theta_2 = \theta(h_2 - \zeta_2)$ and $k_{rw,2} = k_{rw}(h_2 - \zeta_2)$ to denote the water content and the relative permeability in the bulk. Recall that θ and k_{rw} are given by (2.19) and (2.21), respectively.

The governing equations in the fracture Ω_1 are given by

$$\frac{\mathrm{d}V_1}{\mathrm{d}t} = \int_{\Omega_1} \left(\lambda_1 + \lambda_2\right) \mathrm{d}\mathbf{x} + \int_{\Omega_1} f_1 \mathrm{d}\mathbf{x}, \qquad \text{in } \Omega_1 \times (0, T), \tag{3.20}$$

$$h_1 = g_{0,1}, \qquad \text{in } \Omega_1 \times \{0\}. \tag{3.21}$$

where V_1 is water volume in the fracture, and f_1 is an external source or sink of water, which can be employed to mimic direct infiltration or evaporation scenarios. The water volume V_1 can generally be written as a function of the hydraulic head h_1 in the fracture

$$V_1(h_1) = \int_{\substack{\Omega_1 \\ z \le h_1}} a_1 \, \mathrm{d}\mathbf{x}, \tag{3.22}$$

Equation (3.20) is an ODE that keeps track of the volumetric changes of water inside the fracture, requiring the rate of change of water volume to match the amount of water exchanged via the mortar fluxes (in the direction normal to the fracture) plus the source terms. Note that, in contrast to the single-phase case, boundary conditions are not imposed in the fracture.

At interfaces Γ_1 and Γ_2, the mortar fluxes λ_1 and λ_2 are given by the multiphase extension of (3.11) and (3.12):

$$\lambda_1 = -\gamma_1 \frac{2k_1}{a_1} k_{rw,1} \left(h_1 - h_2\right), \qquad \text{on}\, \Gamma_1 \times (0,T), \tag{3.23}$$

$$\lambda_2 = -\gamma_2 \frac{2k_2}{a_1} k_{rw,2} \left(h_1 - h_2\right), \qquad \text{on}\, \Gamma_2 \times (0,T). \tag{3.24}$$

Here, γ_1 and γ_2 are threshold functions that control the flow through the interfaces:

$$\gamma_j(h_1, h_2, \psi_L) = \begin{cases} 1, & h_2 - \zeta_2 > \psi_L \\ 1, & h_1 - \zeta_1 > \psi_L \\ 0, & \text{otherwise} \end{cases}, \qquad \text{on}\ \Gamma_j \times (0,T), \qquad j \in \{1,2\}, \tag{3.25}$$

where ψ_L is the capillary pressure head threshold; that is, the minimum pressure head necessary to break the capillary barrier. In practice, ψ_L takes values very close to zero. However, values below zero are also possible [125].

Furthermore, $k_{rw,1}$ and $k_{rw,2}$ are the mortar relative permeabilities given by a classical potential-based upstream weighting [33]:

$$k_{rw,j}(h_1, h_2) = \begin{cases} k_{rw}(h_2 - \zeta_2), & h_2 \geq h_1 \\ k_{rw}(h_1 - \zeta_1), & h_2 < h_1, \end{cases}, \qquad \text{on}\ \Gamma_j \times (0,T), \qquad j \in \{1,2\}. \tag{3.26}$$

Equations (3.23) and (3.24) complemented by (3.25) and (3.26) establish that: (1) interfaces act as impervious barriers unless the capillary barrier is overcome, and (2) if the capillary barrier is overcome, interfaces become conductive with the direction of the flow given by the upstream direction.

An important point to consider is that, in practice, one does not know *a priori* which part of the interfaces will be blocking and which will be conductive, since the hydraulic heads needed for the evaluation of γ_1 and γ_2 are part of the solution. Thus, solution strategies based on regularization or active level sets must be employed. The former is computationally efficient but prone to mass conservation errors (situations where water leaks into the fracture before overcoming the capillary barrier are not uncommon) whereas the latter is more costly but conserves mass exactly. In Paper D, we employ a strategy based on active level sets.

3.3 Extension to fracture networks

The aim of this section will be to generalize the results from the previous section to fracture networks. To do that, we first introduce a mixed-dimensional decomposition of

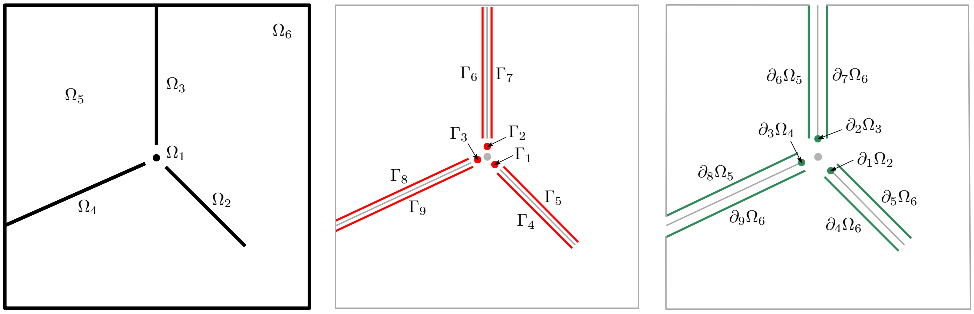


Figure 3.3: Decomposition of a domain $Y \subset \mathbb{R}^2$. Left: Subdomains. Center: Interfaces. Right: Internal boundaries.

the domain of interest, then proceed to write the generic model for the single-phase case, and finally present the generic model for the unsaturated case.

3.3.1 Mixed-dimensional geometric decomposition

Let $Y \subset \mathbb{R}^n$, $n \in \{2, 3\}$, represent a domain decomposed into m flat open-bounded subdomains Ω_i of know dimensionality $d_i = d(i)$, such that $Y = \cup_{i=1}^{m} \Omega_i$. We assume all $d-$dimensional subdomains to be strictly disjointed; for the case $n = 3$, this means that the bulk is composed by simply connected three-dimensional subdomains, fractures are simply connected two-dimensional subdomains, the intersection between fractures are one-dimensional lines, and the intersection between intersection lines are zero-dimensional points. In other words, we consider only the case where there is a dimension-one gap between subdomains; this excludes cases such as one-dimensional lines embedded in a three-dimensional host.

Subdomains separated from one dimension are allowed to be connected via interfaces Γ_j, for $j \in \{1, \ldots, M\}$ (see center Fig. 3.3). To be precise, we use $\hat{\jmath} \in \{1, \ldots, m\}$ and $\check{\jmath} \in \{1, \ldots, m\}$ to denote the indices of the higher and lower dimensional neighboring subdomains of the interface with the index j. Furthermore, the internal boundary of $\Omega_{\hat{\jmath}}$ spatially coinciding with Γ_j is denoted by $\partial_j \Omega_{\hat{\jmath}}$; which in turn coincides with $\Omega_{\check{\jmath}}$ (see the right Figure 3.3).

To keep track of the connection between subdomains and interfaces, we introduce the index sets $\hat{S}_i$ and $\check{S}_i$, containing the indices of the higher-dimensional (respectively lower-dimensional) neighboring interfaces of Ω_i. As an example, consider Ω_3 from the left Figure 3.3, where $\hat{S}_3 = \{6, 7\}$ and $\check{S}_3 = \{2\}$.

We shall then consider the disjointed unions of subdomains and interfaces as

$$\Omega = \bigsqcup_{i=1}^{m} \Omega_i \qquad \text{and} \qquad \Gamma = \bigsqcup_{j=1}^{M} \Gamma_j .$$

Moreover, we decompose the boundary of the domain Ω into its Neumann, Dirichlet, and internal parts. To be precise:

$$\partial\Omega = \partial_N\Omega \cup \partial_D\Omega \cup \partial_I\Omega, \tag{3.27}$$

where $\partial_N\Omega = \cup_{i=1}^{m}\partial_N\Omega$, $\partial_D\Omega = \cup_{i=1}^{m}\partial_D\Omega$, and $\partial_I\Omega = \cup_{i=1}^{m}\cup_{j\in\check{S}_i}\partial_j\Omega_i$.

In Paper C, the reader will encounter a more formal mixed-dimensional geometry decomposition. In particular, the theory requires fully embedded tips to be represented as separate (in this case zero-dimensional) subdomains. Since the aim of this section is to introduce models valid for fracture networks based on physical principles, we prefer to avoid such a formalism.

3.3.2 Single-phase flow equations for fracture networks

We are now in a position to write the set of equations valid for a generic fracture networks. We shall assume that a domain is decomposed into m subdomains of dimensionality 0 to n, and M interfaces of dimensionality of 0 to $n-1$. The set of equations, in strong form, is then given by

$$\begin{aligned}
\nabla_i\cdot\mathbf{q}_i - \sum_{j\in\hat{S}_i}\lambda_j &= f_i, && \text{in}\,\Omega_i, && d_i = 0,\dots,n, && (3.28)\\
\mathbf{q}_i &= -\mathbf{K}_i\nabla_i h_i, && \text{in}\,\Omega_i, && d_i = 1,\dots,n, && (3.29)\\
\lambda_j &= -\frac{2k_j}{a_{\check{j}}}\left(h_{\check{j}} - h_{\hat{j}}\right), && \text{on}\,\Gamma_j, && d_j = 0,\dots,n-1, && (3.30)\\
\mathbf{q}_{\hat{j}}\cdot\mathbf{n}_{\hat{j}} &= \lambda_j, && \text{on}\,\partial_j\Omega_{\hat{j}}, && d_j = 0,\dots,n-1, && (3.31)\\
\mathbf{q}_i\cdot\mathbf{n}_i &= g_{N,i}, && \text{on}\,\partial_N\Omega_i, && d_i = 1,\dots,n, && (3.32)\\
p_i &= g_{D,i}, && \text{on}\,\partial_D\Omega_i, && d_i = 1,\dots,n. && (3.33)
\end{aligned}$$

Equation (3.28) is the mass conservation equation valid for all subdomains. Note, however, that the first term is void for 0-dimensional domains (since 0-dimensional Darcy fluxes are nonphysical), whereas the second term is void for n-dimensional domains (since n-dimensional mortar fluxes are also nonphysical). Moreover, note that the summation term now allows for multiple higher-dimensional adjacent interfaces, which can be seen as a generalization of the second term in (3.7).

The rest of the equations are straightforward extensions of the preceding section, and do not require further explanation.

3.3.3 Unsaturated flow equations for fracture networks

Let us now present the extension of the model introduced in Section 3.2.2 to fracture networks.

A key point to stress is that, contrary to what is required by the generic single phase flow model from Section 3.3.2, due to our hypothesis of instantaneous equilibration, the generic unsaturated model does not employ the full mixed-dimensional decomposition.

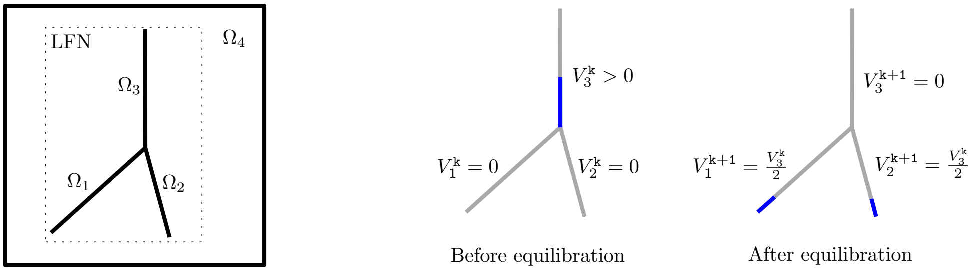


Figure 3.4: Local fracture networks and the distribution phase. Left: A local fracture network with members Ω_1, Ω_2, and Ω_3 fully embedded in a matrix Ω_4. Right: Schematic representation of the distribution phase. Before equilibration Ω_3 is in hydrostatic equilibrium, but the LFN is not. The distribution algorithm enforces equilibrium by distributing the water volume from top to bottom in an LFN.

In particular, fracture intersections (nor intersection of intersections) are included in the model, and therefore governing equations are not required for these subdomains.

This simplification is easily justifiable by the negligible volumetric contribution of $(n-2)$-dimensional objects relative to the total volume of $(n-1)$-dimensional fractures. The same argument holds true for intersections of intersections. Nonetheless, we still require each n-dimensional object to be disjointed. As shown in the left Figure 3.4, we refer to a cluster of $(n-1)$-dimensional objects as a Local Fracture Network (LFN).

With these considerations, the complete set of equations, valid for subdomains of dimensionality $n-1$ to n, and interfaces of dimensionality $n-1$, is given by

$$\frac{\partial \theta_i}{\partial t} + \nabla \cdot \mathbf{q}_i - f_i = 0, \qquad \text{in } \Omega_i \times (0,T), \qquad d_i = n, \tag{3.34}$$

$$\frac{\mathrm{d}V_i}{\mathrm{d}t} - \int_{\Omega_i} \sum_{j \in \hat{S}_i} \lambda_j \,\mathrm{d}\mathbf{x} - \int_{\Omega_i} f_i \,\mathrm{d}\mathbf{x} = 0, \qquad \text{in } \Omega_i \times (0,T), \qquad d_i = n-1, \tag{3.35}$$

$$\mathbf{q}_i + \mathbf{K}_i k_{rw,i} \nabla h_i = 0, \qquad \text{in } \Omega_i \times (0,T), \qquad d_i = n, \tag{3.36}$$

$$\lambda_j + \gamma_j \frac{2k_j}{a_{\check{j}}} k_{rw,j} \left(h_{\check{j}} - h_{\hat{j}}\right) = 0, \qquad \text{on } \Gamma_j \times (0,T), \qquad d_j = n-1, \tag{3.37}$$

$$\mathbf{q}_{\hat{j}} \cdot \mathbf{n}_{\hat{j}} - \lambda_j = 0, \qquad \text{on } \partial_j \Omega_{\hat{j}} \times (0,T), \qquad d_j = n-1, \tag{3.38}$$

$$\mathbf{q}_i \cdot \mathbf{n}_i - g_{N,i} = 0, \qquad \text{on } \partial_N \Omega_i \times (0,T), \qquad d_i = n, \tag{3.39}$$

$$h_i - g_{D,i} = 0, \qquad \text{on } \partial_D \Omega_i \times (0,T), \qquad d_i = n, \tag{3.40}$$

$$h_i - g_{0,i} = 0, \qquad \text{in } \Omega_i \times \{0\}, \qquad d_i = n-1, n, \tag{3.41}$$

and complemented with relevant constitutive relationships.

We remark that the above model guarantees hydrostatic equilibrium in each $(n-1)$-dimensional individual fracture and not in an LFN. Naturally, this might lead to non-equilibrium states. Thus, to ensure that an LFN is in equilibrium before proceeding to the next time step, hydrostatic equilibrium must be enforced *between* time-steps using a water redistribution algorithm. A schematic representation of this process, for an equilibration phase `k`, where equal distribution in a *top-to-bottom* fashion is enforced, is shown in the right Figure 3.4. Naturally, a more intricate LFN will demand for a more robust redistribution algorithm. This is an interesting problem on its own which might benefit from a graph-like approach. The reader is referred to Paper D for further ideas in this direction.

Chapter 4

Functional *a posteriori* error estimates for linear elliptic problems

Thus far, we have introduced mathematical models in non-fractured (Chapter 2) and fractured (Chapter 3) domains based on physical principles, and as such, only the strong forms of such models were required. In this chapter, however, we will need a slightly more involved mathematical language. Indeed, the aim of this chapter is to introduce the basic idea of functional *a posteriori* error estimates, which provides the basis for Paper C. However, unlike Paper C where we derive error estimates for approximations to the single-phase flow equations in fracture networks (see Section 3.3.2), in this chapter, we limit our exposition to a single domain (see Section 2.1) for the sake of simplicity.

To avoid confusion, from now on we will use u to refer to the potential (i.e., the hydraulic head). We consider the linear elliptic problem with homogeneous Dirichlet boundary conditions in the strong mixed dual form: Find $(\mathbf{q}, u)$ such that

$$\nabla \cdot \mathbf{q} = f, \qquad \text{in}\,\Omega, \tag{4.1a}$$

$$\mathbf{q} = -\mathbf{K}\nabla u, \qquad \text{in}\,\Omega, \tag{4.1b}$$

$$u = 0, \qquad \text{on}\,\partial\Omega. \tag{4.1c}$$

The model problem can also be written only in terms of the potential by substituting (4.1b) into (4.1a). The resulting set of equations is referred to as the strong primal form and reads: Find u such that

$$-\nabla \cdot \mathbf{K}\nabla u = f, \qquad \text{in}\,\Omega, \tag{4.2a}$$

$$u = 0, \qquad \text{on}\,\partial\Omega. \tag{4.2b}$$

To be able to perform the mathematical analysis, we further require $f \in L^2(\Omega)$ and $\mathbf{K}$ to be symmetric, bounded, and uniformly positive definite, such that

$$c_{1,\Omega}^2|\xi|^2 \leq \mathbf{K}\xi \cdot \xi \leq c_{2,\Omega}^2|\xi|^2 \quad \forall \xi \in \mathbb{R}^n. \tag{4.3}$$

Solutions to the problems (4.1) and (4.2) are required to hold pointwise, which can often be too restrictive. The usual practice is then to test the governing equations against smooth functions and with this to relax the pointwise regularity needed to guarantee the existence of the derivatives in the classical sense. This can be achieved by replacing the classical derivatives by the distributional (weak) derivatives [25]. From this point forward, whenever we use $\nabla(\cdot)$ and $\nabla\cdot(\cdot)$, these will denote the weak gradient and weak divergence, respectively. The reformulated problems are then referred to weak problems (or forms) and the resulting solutions are called weak solutions.

Before writing the weak forms of the model problems, we need to introduce the spaces in which we will search for the weak solutions. In particular, we will need the energy space:

$$H^1(\Omega) = \left\{ v \in L^2(\Omega) \,:\, \nabla v \in \mathbf{L}^2(\Omega) \right\}, \tag{4.4}$$

and the energy space with vanishing functions on the boundary of Ω:

$$H_0^1(\Omega) = \left\{ v \in H^1(\Omega) \,:\, v = 0 \text{ on } \partial\Omega \right\}. \tag{4.5}$$

Furthermore, we will need the vector space

$$\mathbf{H}(\mathrm{div},\Omega) = \left\{ \mathbf{r} \in \mathbf{L}^2(\Omega) \,:\, \nabla\cdot\mathbf{r} \in L^2(\Omega) \right\}, \tag{4.6}$$

Throughout this chapter, $\langle \cdot,\cdot \rangle$ will denote the L^2 inner-product in Ω and $\|\cdot\|$ its corresponding norm.

With the above spaces formally introduced, we are now in position to write the weak versions of the primal and dual problems. The primal weak form reads: Find $u \in H_0^1(\Omega)$ such that

$$\langle \mathbf{K}\nabla u, v \rangle = \langle f, v \rangle \qquad \forall\, v \in H_0^1(\Omega), \tag{4.7}$$

whereas the dual-mixed primal form reads: Find $(\mathbf{q}, u) \in \mathbf{H}(\mathbf{div},\Omega) \times L^2(\Omega)$ such that

$$\left\langle \mathbf{K}^{-1}\mathbf{q}, \mathbf{r} \right\rangle - \langle u, \nabla\cdot\mathbf{r} \rangle = 0 \qquad \forall\, \mathbf{r} \in \mathbf{H}(\mathrm{div},\Omega), \tag{4.8a}$$

$$\langle \nabla\cdot u, v \rangle = \langle f, v \rangle \qquad \forall\, v \in L^2(\Omega). \tag{4.8b}$$

The reader is referred to [92] for the proof that the primal and mixed dual forms are indeed equivalent.

Throughout this chapter, we aim at obtaining computable majorants for the difference between the exact primal solution $u \in H_0^1(\Omega)$ and an arbitrary approximation $v \in H_0^1(\Omega)$, such that

$$|||u - v||| \le \mathscr{M}(v, \mathbf{r}, \mathscr{D}) \qquad \forall\, v \in H_0^1(\Omega),\ \mathbf{r} \in \mathbf{H}(\mathrm{div},\Omega), \tag{4.9}$$

where $|||\cdot||| \,:\, H_0^1(\Omega) \to \mathbb{R}_+$ is the energy norm (4.18) and $\mathscr{D}$ is the set of given data; which includes the domain Ω, the source term f, and the material coefficient $\mathbf{K}$.

Indeed, we say that a majorant is computable and continuous if

$$\mathscr{M}(v, \mathbf{r}, \mathscr{D}) \to 0 \quad \text{when} \quad (v, \mathbf{r}) \to (u, \mathbf{q}). \tag{4.10}$$

Majorants $\mathscr{M}(v, \mathbf{r}, \mathscr{D})$ can be derived using purely functional methods via *variational methods* [80, 96, 96–98, 113] or *transformation of integral identities* [99, 102, 105]. For linear elliptic problems, both methods result in the same majorants [101].

More generally, functional *a posteriori* error estimates were successfully applied to numerous problems in the last decades. A non-exhaustive list includes: elasto-plastic problems [113], the Stokes problem [100], the heat equation [99], the Poisson equation with mixed boundary conditions [108], the reaction-diffusion problem [107], Maxwell's equation [103], mixed-dual approximations of elliptic problems [109], problems with non-linear boundary conditions [111], elliptic problems in exterior domains [86], discontinuous Galerkin approximations to elliptic problems [66], electro-magnetostatic problems [85], non-conforming approximations of elliptic problems [110], variational problems with obstacles [112], and the Biot's equations [63].

However, we remark that several other methods are available in the literature. The reader is referred to the following references for further details: for residual methods [10, 131], for gradient-recovery methods [10, 144], for goal-oriented estimates [84], and for methods based on local flux/stress equilibration [4, 5, 17–19, 26, 42–44, 59, 75, 134, 135].

The rest of this chapter is organized as follows: In Section 4.1, we introduce important tools from functional analysis that are needed in the *a posteriori* estimation. Guaranteed upper bounds for the primal variable are derived in Section 4.2. In Section 4.3, we improve the bounds obtained in the preceding section by exploiting locally mass-conservative properties of the approximations. To provide concrete approximations, in Section 4.4, we introduce the finite element spaces and two families of locally mass-conservative approximations. In Section 4.5, we discuss flux and potential reconstruction techniques necessary to obtain energy-conforming functions starting from locally mass-conservative approximations. Finally, in Section 4.6, we briefly comment on extending the error bounds from monodimensional to mixed-dimensional geometries.

4.1 Tools from functional analysis

In this section, we summarize the standard results from functional analysis that will be needed to later derive the *a posteriori* bounds.

Green's theorem

For any pair $(v, r) \in H_0^1(\Omega) \times \mathbf{H}(\mathrm{div}, \Omega)$, there holds

$$\langle \nabla v, \mathbf{r} \rangle + \langle v, \nabla \cdot \mathbf{r} \rangle = 0. \tag{4.11}$$

Friedrich inequality

There exists a constant $C_{\mathrm{F},\Omega}$, dependent only on Ω, such that

$$\|v\| \le C_{\mathrm{F},\Omega} \|\nabla v\| \quad \forall\, v \in H_0^1(\Omega). \tag{4.12}$$

Local Poincaré inequality [133, 135]

Let $K \subset \Omega$ be a convex subdomain. Then, there exists a constant $C_{\mathrm{P},K}$ dependent only on K, such that

$$\left\|v - \{v\}_K\right\|_K \leq C_{\mathrm{P},K} \|\nabla v\|_K \quad \forall v \in H^1(K), \tag{4.13}$$

where $\{v\}_K = \frac{1}{|K|}\langle v, 1\rangle_K$ denotes the mean quantity of v over K. Furthermore, due to the convexity of K, there holds [12, 87]

$$C_{\mathrm{P},K} \leq \frac{\mathrm{diam}(K)}{\pi} = \frac{h_K}{\pi}. \tag{4.14}$$

Bilinear forms and energy norms [135]

Let the bilinear $\mathscr{B}$ form acting on scalars be defined so that

$$\mathscr{B}(v, w) = \langle \mathbf{K}\nabla v, \nabla w\rangle, \qquad \forall\, v, w \in H^1_0(\Omega), \tag{4.15}$$

and the bilinear form $\mathscr{A}$ acting on the vectors be defined such that

$$\mathscr{A}(\mathbf{r}, \mathbf{s}) = \left\langle \mathbf{K}^{-1}\mathbf{r}, \mathbf{s}\right\rangle, \qquad \forall\, \mathbf{r}, \mathbf{s} \in \mathbf{L}^2(\Omega), \tag{4.16}$$

which are related via

$$\mathscr{B}(v, w) = \mathscr{A}(\mathbf{K}\nabla\, v, \mathbf{K}\nabla\, w) \qquad \forall\, v, w \in\ H^1_0(\Omega). \tag{4.17}$$

The bilinear form $\mathscr{B}$ induces the energy norm

$$|||v||| = \mathscr{B}(v, v)^{1/2} = \left\|\mathbf{K}^{1/2}\nabla v\right\|, \qquad \forall\, v \in\ H^1_0(\Omega), \tag{4.18}$$

whereas the bilinear form $\mathscr{A}$ induces the energy norm

$$|||\mathbf{r}|||_* = \mathscr{A}(\mathbf{r}, \mathbf{r})^{1/2} = \left\|\mathbf{K}^{-1/2}\mathbf{r}\right\|, \qquad \forall\, \mathbf{r} \in\ \mathbf{L}^2(\Omega). \tag{4.19}$$

4.2 Guaranteed upper bound for the primal variable

Let us now present the *a posteriori* error bounds for the primal error in the energy norm [105]. The main result is given in the following theorem:

Theorem 1: Let $u \in H^1_0(\Omega)$ be the solution to the primal weak form (4.7) and $v \in H^1_0(\Omega)$ arbitrary. Then, for any function $\mathbf{r} \in \mathbf{H}(\mathrm{div}, \Omega)$, there holds

$$|||u - v||| \leq |||\mathbf{r} + \mathbf{K}\nabla v|||_* + \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}} \|f - \nabla \cdot \mathbf{r}\| := \mathscr{M}_1(v, \mathbf{r}, f), \tag{4.20}$$

where $\mathscr{M}_1(q, \mathbf{r}, f)$ is the majorant for the primal error measured in the energy norm.

Proof: Start by measuring the difference between the weak primal solution $u \in H_0^1(\Omega)$ and an arbitrary function $v \in H_0^1(\Omega)$ in the energy norm (4.18):

$$\begin{aligned} |||u-v|||^2 &= \mathscr{B}(u-v,u-v) \\ &= \langle \mathbf{K}\nabla(u-v), \mathbf{K}\nabla(u-v)\rangle \\ &= \langle \mathbf{K}\nabla u, \mathbf{K}\nabla(u-v)\rangle + \langle -\mathbf{K}\nabla v, \mathbf{K}\nabla(u-v)\rangle \\ &= \langle f, u-v\rangle + \langle -\mathbf{K}\nabla v, \mathbf{K}\nabla(u-v)\rangle. \end{aligned} \tag{4.21}$$

Here, we used (4.18), (4.15), and the weak primal form (4.7) for the test function $u-v \in H_0^1(\Omega)$. Now, we fix an arbitrary function $\mathbf{r} \in \mathbf{H}(\mathrm{div},\Omega)$ and use Green's theorem (4.11) on the pair $(u-v,\mathbf{r}) \in H_0^1(\Omega) \times \mathbf{H}(\mathrm{div},\Omega)$ to obtain the following identity

$$-\langle \nabla\cdot\mathbf{r}, u-v\rangle - \langle \mathbf{r}, \nabla(u-v)\rangle = 0, \tag{4.22}$$

which is added to (4.21), to obtain

$$\begin{aligned} |||u-v|||^2 &= \langle f-\nabla\cdot\mathbf{r}, u-v\rangle + \langle -(\mathbf{r}+\mathbf{K}\nabla u), \mathbf{K}\nabla(u-v)\rangle \\ &= \langle f-\nabla\cdot\mathbf{r}, u-v\rangle + \left\langle -\mathbf{K}^{-1/2}(\mathbf{r}+\mathbf{K}\nabla u), \mathbf{K}^{1/2}\nabla(u-v)\right\rangle. \end{aligned} \tag{4.23}$$

The first term of (4.23) can be bounded applying the Cauchy-Schwarz inequality, the Friedrich-inequality (4.12), assumption (4.3), and the definition of the energy norm (4.18):

$$\begin{aligned} \langle f-\nabla\cdot\mathbf{r}, u-v\rangle &\le \|f-\nabla\cdot\mathbf{r}\|\;\|u-v\| \\ &\le C_{\mathrm{F},\Omega}\|f-\nabla\cdot\mathbf{r}\|\;\|\nabla(u-v)\| \\ &\le \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}}\|f-\nabla\cdot\mathbf{r}\|\;\left\|\mathbf{K}^{1/2}\nabla(u-v)\right\| \\ &= \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}}\|f-\nabla\cdot\mathbf{r}\|\;|||u-v|||. \end{aligned} \tag{4.24}$$

The second term of (4.23) can be bounded by applying the Cauchy-Schwarz inequality and the definition of the energy norms (4.18) and (4.19)

$$\begin{aligned} \left\langle -\mathbf{K}^{-1/2}(\mathbf{r}+\mathbf{K}\nabla v), \mathbf{K}^{1/2}\nabla(u-v)\right\rangle &\le \left\|\mathbf{K}^{-1/2}(\mathbf{r}+\mathbf{K}\nabla q)\right\|\;\left\|\mathbf{K}^{1/2}\nabla(u-v)\right\| \\ &= |||\mathbf{r}+\mathbf{K}\nabla v|||_*\;|||u-v|||. \end{aligned} \tag{4.25}$$

Substituting (4.24) and (4.25) into (4.23), we get

$$|||u-v|||^2 \le |||u-q|||\left(|||\mathbf{r}+\mathbf{K}\nabla v|||_* + \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}}\|f-\nabla\cdot\mathbf{r}\|\right), \tag{4.26}$$

from which the assertion follows. ■

The majorant $\mathscr{M}_1(v,\mathbf{r},f)$ from (4.20) has two clear contributions: 1) a diffusive error $\eta_{\mathrm{DF}}(v,\mathbf{r}) = |||\mathbf{r}+\mathbf{K}\nabla v|||_*$ measuring the difference between the approximated flux $\mathbf{r}$ and the flux obtained from the $H_0^1(\Omega)$-potential v, and 2) a residual error $\eta_{\mathrm{R}}(\mathbf{r},f) = \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}}\|f-\nabla\cdot\mathbf{r}\|$ measuring how close $\nabla\cdot\mathbf{r}$ is to the exact source f.

The upper bound (4.20) is said to be *sharp*, in the sense that no constant $c_{\mathrm{DF}} < 1$ can multiply the first term and no constant $c_{\mathrm{R}} < \frac{C_{\mathrm{F},\Omega}}{c_{1,\Omega}}$ can multiply the second term [105].

4.3 Improving the bounds using mass-conservative properties

The big limitation of the upper bound from Theorem 1 is that the Friedrich constant $C_{\mathrm{F},\Omega}$ is only available for a small set of geometries [30, 106, 132]. However, if one employs approximations that satisfy mass conservation in some convex partition K of the domain Ω, it is possible to obtain *sharper*, and *fully computable* upper bounds (4.31). The aim of this section is to derive such bounds following, e.g., [104]. We remark that similar bounds can be obtained with a slightly different approach [133, 135, 136].

Let us start by considering the decomposition of the domain,

$$\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} K, \tag{4.27}$$

where K denotes a strictly non-overlapping subdomain of dimension n.

Now, we require the approximated fluxes $\mathbf{r}$ entering the *a posteriori* error estimation to be in $\mathbf{U} \subset \mathbf{H}(\mathrm{div}, \Omega)$, where

$$\mathbf{U} = \{\mathbf{r} \in \mathbf{H}(\mathrm{div}, \Omega) : \langle f, 1\rangle_K = \langle \nabla \cdot \mathbf{r}, 1\rangle_K \ \ \forall\, K \in \mathcal{T}_h\}. \tag{4.28}$$

We will also need the equivalence between the continuous and broken norms

$$\|v\|^2 = \sum_{K \in \mathcal{T}_h} \|v\|_K^2 \qquad \forall\, v \in \ L^2(\Omega), \tag{4.29}$$

$$|||v|||^2 = \sum_{K \in \mathcal{T}_h} \left\|\mathbf{K}^{1/2} \nabla v\right\|_K^2 \qquad \forall\, v \in \ H_0^1(\Omega). \tag{4.30}$$

Below, we present the improved upper bound:

Theorem 2: Let $u \in H_0^1(\Omega)$ be the solution to the weak primal form (4.7) and $v \in H_0^1(\Omega)$ arbitrary. Then, for any function $\mathbf{r} \in \mathbf{U}$, there holds

$$|||u - v||| \leq |||\mathbf{r} + \mathbf{K}\nabla v|||_* + \left(\sum_{K \in \mathcal{T}_h} \frac{h_K^2}{\pi^2 c_{1,K}^2} \|f - \nabla \cdot \mathbf{r}\|_K^2 \right)^{1/2} := \mathcal{M}_2(v, \mathbf{r}, f). \tag{4.31}$$

Proof: The proof closely follows the previous case, modulo the treatment of the residual term. Indeed, the first term of (4.23) can now be bounded as

$$
\begin{aligned}
\langle f - \nabla \cdot \mathbf{r}, u - v \rangle &= \sum_{K \in \mathscr{T}_h} \langle f - \nabla \cdot \mathbf{r}, u - v \rangle_K \\
&= \sum_{K \in \mathscr{T}_h} \langle f - \nabla \cdot \mathbf{r}, (u - v) - \{u - v\}_K \rangle_K \\
&\leq \sum_{K \in \mathscr{T}_h} \|f - \nabla \cdot \mathbf{r}\|_K \left\|(u - v) - \{u - v\}_K\right\|_K \\
&\leq \sum_{K \in \mathscr{T}_h} C_{\mathrm{P},K} \|f - \nabla \cdot \mathbf{r}\|_K \, \|\nabla(u - v)\|_K \\
&\leq \sum_{K \in \mathscr{T}_h} \frac{C_{\mathrm{P},K}}{c_{1,K}} \|f - \nabla \cdot \mathbf{r}\|_K \left\|\mathbf{K}^{1/2} \nabla(u - v)\right\|_K \\
&\leq \left(\sum_{K \in \mathscr{T}_h} \frac{h_K^2}{\pi^2 c_{1,K}^2} \|f - \nabla \cdot \mathbf{r}\|_K^2 \right)^{1/2} \left(\sum_{K \in \mathscr{T}_h} \left\|\mathbf{K}^{1/2} \nabla(u - v)\right\|_K^2 \right)^{1/2} \\
&\leq \left(\sum_{K \in \mathscr{T}_h} \frac{h_K^2}{\pi^2 c_{1,K}^2} \|f - \nabla \cdot \mathbf{r}\|_K^2 \right)^{1/2} |||u - v|||,
\end{aligned} \tag{4.32}
$$

where we used the equivalence (4.29), the locally mass-conservative property of the fluxes $\mathbf{r} \in \mathbf{U}$, the Cauchy-Schwarz inequality, the local Poincaré inequality (4.13), the ellipticity assumption (4.3) on $K \subset \Omega$, the Cauchy-Swchwarz inequality once again, and finally the equivalence (4.30).

By substituting (4.25) and (4.32) into (4.23), we obtain

$$
|||u - v|||^2 \leq |||u - v||| \left(|||\mathbf{r} + \mathbf{K}\nabla v|||_* + \left(\sum_{K \in \mathscr{T}_h} \frac{h_K^2}{\pi^2 c_{1,K}^2} \|f - \nabla \cdot \mathbf{r}\|_K^2 \right)^{1/2} \right), \tag{4.33}
$$

from which the assertion clearly follows. ■

The bound (4.31) is superior to (4.20) in two important aspects: (1) it is fully computable in the sense that only constants associated with the mesh size h_K and the material parameter $c_{1,K}$ are involved, and (2) the residual estimators are super-convergent for $\mathbf{r}|_K \in \mathbf{RTN}_s(K)$ (Raviart-Thomas-Nédélec elements of degree $s \geq 0$) due to the local Poincaré constants $C_{\mathrm{P},K}$ instead of the Friedrich constant $C_{\mathrm{F},\Omega}$.

Thanks to this super-convergent property of the residual estimator, there holds

$$
\mathscr{M}_2(v, \mathbf{r}, f) \leq \mathscr{M}_1(v, \mathbf{r}, f). \tag{4.34}
$$

4.4 Locally mass-conservative approximations

Naturally, one does not count with arbitrary approximations but rather approximations obtained with numerical methods. Here, we are mainly interested in deriving computable

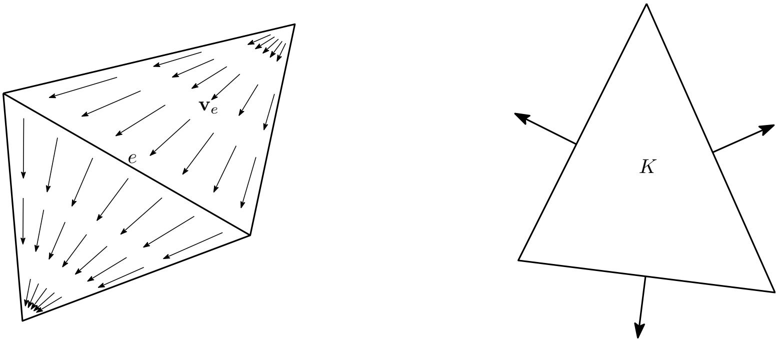


Figure 4.1: Basis functions $\mathbf{v}_e$ (left) and degrees of freedom for a two-dimensional simplex (right) associated with the space $\mathbf{RTN}_0(\mathscr{T}_h)$.

estimates for locally mass-conservative approximations of the lowest order. Classical examples of such methods are the Cell-Centered Finite Volume Methods (CCFVM) and the Mixed-Finite Element Method (MFEM).

To write the concrete approximations, we first need to introduce some further notation, as well as the discrete finite element spaces.

For each $K \in \mathscr{T}_h$, we will use ∂K to denote the boundary of the simplex K. Let $e \in \mathscr{E}_K$ denote an edge in the set of edges of K. Consider now the following discrete spaces defined on the partition $\mathscr{T}_h$,

$$\mathbb{P}_s(\mathscr{T}_h) = \{q_h \in L^2(\Omega) \,:\, v_h|_K \in \mathbb{P}_s(K) \quad \forall\, K \in \mathscr{T}_h\} \tag{4.35}$$

$$\mathbf{RTN}_0(\mathscr{T}_h) = \{\mathbf{r}_h \in \mathbf{H}(\mathrm{div},\Omega) \,:\, \mathbf{r}_h|_K \in \mathbf{RTN}_0(K) \quad \forall\, K \in \mathscr{T}_h\} \tag{4.36}$$

where

$$\mathbf{RTN}_0(K) = [\mathbb{P}_0(K)]^n + \mathbf{x}\mathbb{P}_0(K), \tag{4.37}$$

satisfying

$$\nabla \cdot \mathbf{r}_h \in \mathbb{P}_0(K) \quad \forall\, K \in \mathscr{T}_h \qquad \text{and} \qquad \mathbf{r}_h \cdot \mathbf{n}_e \in \mathbb{P}_0(e) \quad \forall\, e \in \mathscr{E}_K.$$

In Figure 4.1 we show the basis functions $\mathbf{v}_e$ associated with the space $\mathbf{RTN}_0(\mathscr{T}_h)$ in triangles and the degrees of freedom associated with the simplex K.

4.4.1 Cell-Centered Finite Volume Methods

The general principle of cell-centered finite volume approximations is based on integrating the conservation equation (4.1a) in all $K \in \mathscr{T}_h$, such that

$$\int_K \nabla \cdot \mathbf{q}\, \mathrm{d}x = \int_{\partial K} \mathbf{q} \cdot \mathbf{n}\, \mathrm{d}S = \sum_{e \in \mathscr{E}_K} \mathbf{q}_e \cdot \mathbf{n}_e\, A_e = \sum_{e \in \mathscr{E}_K} Q_e = \int_K f\, \mathrm{d}x, \tag{4.38}$$

where Q_e are the *exact* normal fluxes on the edges of K.

The exact flux is approximated using a discrete version of Darcy's law, where, on each edge, the fluxes are linearly related to a given number of neighboring (cell-center) potentials $u_h|_K$, such that

$$Q_e \approx F_e = \sum_{K \in \mathscr{F}_e} t_{K,e} u_{h,K} \quad \forall\, e \in \mathscr{E}_K, \quad \forall\, K \in \mathscr{T}_h. \tag{4.39}$$

Here, $\mathscr{F}_e$ denotes the set of neighboring potentials associated with the edge e and $t_{K,e}$ are the transmissibility coefficients [1]. Note that we slightly abused of notation and used K also as an index. In general, we can write the approximated problem for a CCFVM method as follows [136]: Find $u_h \in \mathbb{P}_0(\mathscr{T}_h)$ such that

$$\sum_{e \in \mathscr{E}_K} \langle F_e, 1 \rangle_K = \langle f, 1 \rangle_K, \qquad \forall\, K \in \mathscr{T}_h. \tag{4.40}$$

4.4.2 Lowest order Mixed-Finite Element Method

Lowest order MFEM is based on solving the approximated version of the dual weak mixed form (4.8). We look simultaneously for approximated fluxes $\mathbf{q}_h$ and approximated potentials u_h.

The approximated problem reads [136]: Find $(\mathbf{q}_h, u_h) \in \mathbf{RTN}_0(\mathscr{T}_h) \times \mathbb{P}_0(\mathscr{T}_h)$ such that

$$\left\langle \mathbf{K}^{-1} \mathbf{q}_h, \mathbf{r}_h \right\rangle - \langle u_h, \nabla \cdot \mathbf{r}_h \rangle = 0 \qquad \forall\, \mathbf{r}_h \in \mathbf{RTN}_0(\mathscr{T}_h) \tag{4.41a}$$

$$\langle \nabla \cdot u_h, v_h \rangle = \langle f, v_h \rangle \qquad \forall\, v_h \in \mathbb{P}_0(\mathscr{T}_h). \tag{4.41b}$$

We will use "RT0-P0" to denote method (4.41).

4.5 Concrete bounds

Recall that Theorem 1 and Theorem 2 require the arbitrary flux in $\mathbf{H}(\mathrm{div}, \Omega)$ and the arbitrary potential in $H_0^1(\Omega)$. Note that RT0-P0 satisfies only the first of those requirements ($\mathbf{q}_h \in \mathbf{RTN}_0(\mathscr{T}_h) \subset \mathbf{H}(\mathrm{div}, \Omega)$) whereas an CCFVM fulfill none of them.

Thus, to obtain computable estimates, we need to enhance the regularity of the approximated potential u_h and obtain a reconstructed version $\tilde{u}_h$ with higher regularity. Additionally, for a CCFVM, we have to extend the normal fluxes F_e with the help of $\mathbf{RTN}_0(K)$ basis functions in each $K \in \mathscr{T}_h$. We will refer, respectively, to these procedures as *potential reconstruction* and *extension of normal fluxes*.

4.5.1 Extension of normal fluxes

Assume that $F_e \in \mathbb{P}_0(e)$ on each $e \in \mathscr{E}_K$ for all $K \in \mathscr{T}_h$ are available. Then, the reconstruction procedure reads: Obtain $\mathbf{u}_h \in \mathbf{RTN}_0(\mathscr{T}_h)$ such that

$$\langle \mathbf{u}_h \cdot \mathbf{n}_e, 1 \rangle_e = F_e \qquad \forall\, e \in \mathscr{E}_k, \qquad \forall\, K \in \mathscr{T}_h. \tag{4.42}$$

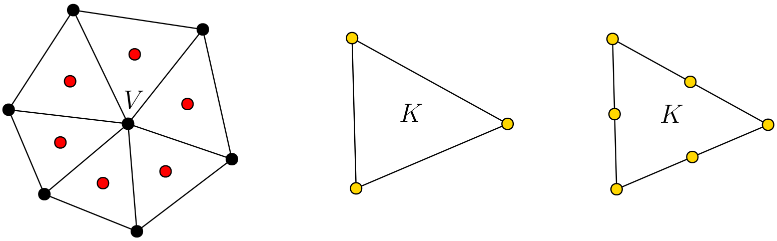


Figure 4.2: Left: A patch associated with the internal node V of a two-dimensional grid. In black, the physical nodes. In red, the cell-centers. Center: $\mathbb{P}_1(K)$-Lagrangian nodes for a two-dimensional simplex K. Right: $\mathbb{P}_2(K)$-Lagrangian nodes for a two-dimensional simple K.

4.5.2 Potential reconstruction

The last ingredient to make our upper bounds computable is to obtain an approximated potential in $H_0^1(\Omega)$. In general, we look for an interpolant of the form

$$\mathscr{G} : \mathbb{P}_0(\mathscr{T}_h) \to \mathbb{P}_s(\mathscr{T}_h) \cap H_0^1(\Omega), \quad s > 0. \tag{4.43}$$

In the following techniques, we use V to denote a Lagrangian node of K. Moreover, $\mathscr{T}_V$ is the set of elements K associated with the Lagrangian node V, see, e.g. the left Figure 4.2 for a two-dimensional example.

Potential reconstruction I (PRI) [34]

Assume $u_h \in \mathbb{P}_0(\mathscr{T}_h)$ is available. Then, find $\tilde{u}_h \in \mathbb{P}_1(\mathscr{T}_h) \cap H_0^1(\Omega)$ such that

$$\tilde{u}_h(V) = \begin{cases} \frac{\sum_{K\in\mathscr{T}_V} |K| u_h|_K}{\sum_{K\in\mathscr{T}_V} |K|}, & V \in \Omega, \\ 0, & V \in \partial\Omega. \end{cases} \tag{4.44}$$

This technique is arguably the simplest way to reconstruct the potential $u_h \in \mathbb{P}_0(\mathscr{T}_h)$. It consists in performing an average of the cell-center potentials to obtain the $\mathbb{P}_1(K)$-Lagrangian nodes in each $K \in \mathscr{T}_h$.

Potential reconstruction II (PRII)

Assume $u_h \in \mathbb{P}_0(\mathscr{T}_h)$ and $\mathbf{q}_h \in \mathbf{RTN}_0(\mathscr{T}_h)$ are available. Let $\mathbf{s}|_K \in \left[\mathbb{P}_0(K)\right]^n + \mathbf{x}\mathbb{P}_0(K)$ be such that,

$$\mathbf{s}|_K = -\mathbf{K}^{-1}|_K\, \mathbf{q}_h|_K \qquad \forall\, K \in \mathscr{T}_h. \tag{4.45}$$

Moreover, let $\mathbf{d}_{K,V}$ be the vector constructed from the barycenter $\mathbf{b}_K(\mathbf{x})$ of K to the node V. Then, find $\tilde{u}_h \in \mathbb{P}_1(\mathscr{T}_h) \cap H_0^1(\Omega)$ such that

$$\tilde{u}_h(V) = \begin{cases} \frac{\sum_{K\in\mathscr{T}_V} |K| \left(u_{h,K} + \mathbf{s}(\mathbf{b}_K)\cdot\mathbf{d}_{K,V}\right)}{\sum_{K\in\mathscr{T}_V} |K|}, & V \in \Omega, \\ 0, & V \in \partial\Omega. \end{cases} \tag{4.46}$$

Unlike PRI, this technique uses local flux information $\mathbf{q}_h|_K \in \mathbf{RTN}_0(K)$ to approximate the gradient of the potential $\mathbf{s}|_K$, locally. Then, $\mathbf{s}|_K$ is evaluated in the barycenter of K and projected onto the Lagrangian node V. The projected quantity is added to the cell center value $u_{h,K}$, resulting in a Taylor-like expansion truncated to the first derivative.

Potential reconstruction III (PRIII) [133, 135]

Assume $u_h \in \mathbb{P}_0(\mathscr{T}_h)$ and $\mathbf{q}_h \in \mathbf{RTN}_0(\mathscr{T}_h)$ are available. First, find $u_h' \in \mathbb{P}_2(\mathscr{T}_h)$ such that

$$\left\langle -\mathbf{K}\nabla u_h', 1 \right\rangle_K = \langle \mathbf{q}_h, 1 \rangle_K \qquad \forall\, K \in \mathscr{T}_h, \tag{4.47}$$

$$\frac{1}{|K|}\left\langle u_h', 1 \right\rangle_k = u_h|_K \qquad \forall\, K \in \mathscr{T}_h. \tag{4.48}$$

Now, find $\tilde{u}_h \in \mathbb{P}_2(\mathscr{T}_h) \cap H_0^1(\Omega)$ such that

$$\tilde{u}_h(V) = \begin{cases} \frac{1}{|\mathscr{T}_V|} \sum_{K \in \mathscr{T}_V} u_h'|_K(V) & V \in \Omega \\ 0, & V \in \partial\Omega \end{cases} \tag{4.49}$$

This technique is based on first obtaining a post-processed potential $u_h' \in \mathbb{P}_2(\mathscr{T}_h)$ whose mean value in each $K \in \mathscr{T}_h$ matches the values of $u_h \in \mathbb{P}_0(\mathscr{T}_h)$. This potential is non-conforming, in the sense that $u_h' \notin H_0^1(\Omega)$. Thus, in a subsequent step, the conformity is enforced using an averaging operator (Oswald interpolator) that takes the average of $u_h'|_K \in \mathbb{P}_2(K)$ at the Lagrangian nodes V [3, 29, 55].

4.6 Extension to mixed-dimensional geometries

In Paper C, we extend the results presented in this chapter to fracture networks such as that in Section 3.3.2. To be precise, we propose guaranteed upper bounds for approximations to the primal and dual variables, and two-sided bounds for the primal-dual pair.

The reader will appreciate that such extensions are non-trivial. In particular, since we are required to deal with mixed-dimensional function spaces, in addition to the inherent complexity associated with working with several subdomains (some of them potentially floating, e.g., Ω_2 in Figure 3.3) and interfaces. In Paper C, we address these challenges by recasting the model problem into a compact mixed-dimensional elliptic model with the help of tools from exterior calculus [22]. Indeed, this allows us to perform the *a posteriori* analysis, following very closely the monodimensional case.

However, we remark that the main idea of measuring the deviation between the exact weak solutions with arbitrary energy-conforming functions prevails even in this most general setting.

Chapter 5

Summary and outlook

This chapter provides a summary of the papers and the outlook of the doctoral thesis.

5.1 Summary of papers

In the following, we present a brief summary of the papers included in Part II. The papers are presented in chronological order. For completeness, in Figure 5.1, we also show the interdependence between the papers.

5.1.1 Paper A [129]: A finite-volume-based module for unsaturated poroelasticity

This book chapter provides a description of `fv-unsat`, a multipoint finite-volume-based MRST module for simulating the equations of unsaturated poroelasticity. The set of equations corresponds to a coupled flow/mechanical problem that is obtained as a natural extension of Biot's equations of poroelasticity, cf. Section 2.4. We discretize the set of equations using backward Euler in time. Moreover, in space, the flow sub-problem is discretized with the Multi-Point Flux Approximation (MPFA) method whereas the mechanical sub-problem is discretized with the Multi-Point Stress Approximation (MPSA) method.

We solve the coupled flow/mechanical problem fully implicitly and use a modified Picard iteration to linearize the system of equations. We provide two converge tests, one neglecting deformation effects (in this case the set of equations reduces to the well-known Richards' equation, cf. Section 2.2) and another accounting for deformation. In both cases, we obtain similar convergence rates as the ones reported for single-phase flow (Section 2.1) and saturated poroelasticity (Section 2.3), respectively. The rest of the chapter is devoted to providing examples on how to use the module, including a water infiltration process in an initially dry soil and the evaporation process from a saturated clayey sample subjected to atmospheric boundary conditions (see e.g. Section 2.6).

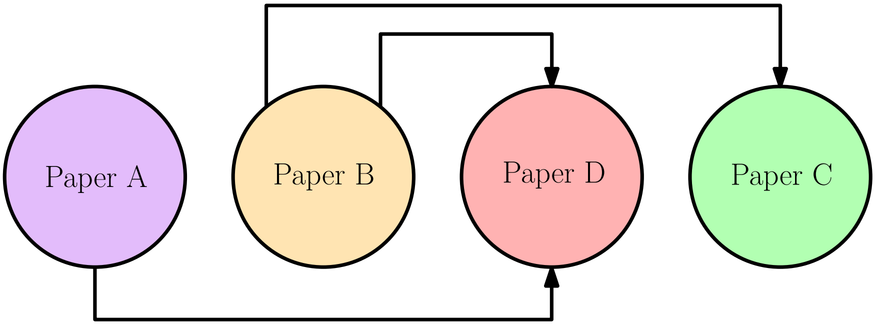


Figure 5.1: Relations between the papers. Paper C and Paper D are implemented in PorePy (Paper B). Additionally, Paper D is partially based on the results obtained in Paper A. The suggested reading order is: Paper A, Paper B, Paper D, and Paper C.

5.1.2 Paper B [57]: Porepy: an open-source software for simulation of multiphysics processes in fractured porous media

In this article, we present a new tool for simulating multiphysics processes in fractured porous media named PorePy. We start by discussing the design principles when mixed-dimensional geometries are required to be included in the processes, in particular, we discuss the permissible types of geometries and allowable coupling structures. We then move on to introduce the models available in a strong form, including (1) flow in fractured media, (2) coupled flow and transport in fractured media, and (3) poroelastic fracture deformation by contact mechanics.

Implementation aspects are discussed next. In particular, we discuss the overall architecture of PorePy, mixed-dimensional geometrical aspects, gridding strategies, parameter specification, declaration of primary variables, discretization classes, and the global assembly of equations. Then, we present numerical validations including a two-dimensional flow benchmark problem, Mandel's problem, and Sneddon's problem. Finally, we present applications of multiphysics processes, including a fully coupled flow and transport example using matching and nonmatching grids, and a three-dimensional poroelasticity and fracture deformation simulation.

5.1.3 Paper C [128]: *A posteriori* error estimates for hierarchical mixed-dimensional elliptic equations

In this article, we derive *a posteriori* error estimates for approximations to hierarchical mixed-dimensional elliptic equations, which, in particular, model the single-phase flow in fractured media as introduced in Section 3.3.2. To be more precise, we obtain guaranteed upper bounds for the primal and dual variables, and two-sided bounds for the primal-dual pair.

The bounds are based on error estimates of the functional type (see Chapter 4) and as such are valid for *any* conforming approximation in the energy space. However, unlike the standard functional approach that does not exploit the conservation properties of the approximations (and thus assumes no-conservation), we exploit these conservation properties to obtain sharper bounds when mass-conservation can be satisfied either in the subdomain level, the grid level, or point-wise (exactly).

Moreover, by recasting the model using a compact mixed-dimensional notation and exploiting the fact that Poincaré-type inequalities imply weighted norms, we are able to encompass the four aforementioned cases into a unified abstract framework. Therefrom, different majorants depending upon the level at which residual balances can be satisfied are presented in terms of local quantities. This results not only in guaranteed but also in fully computable estimates when mass is conserved either locally or exactly.

Concrete bounds are then obtained for MFEM approximations of the lowest order, and the applicability to other locally mass-conservative schemes is discussed. Finally, the bounds are validated for four locally mass-conservative approximations (RT0-P0, MVEM-P0, MPFA, and TPFA) both with two- and three-dimensional manufactured solutions and two- and three-dimensional applications based on benchmark problems in fractured media.

5.1.4 Paper D [130]: A model for unsaturated flow in the presence of fractures acting as capillary barriers

This paper deals with modeling of unsaturated flow (i.e., simultaneous flow of water and air in a porous media) accounting for fractures which naturally act as capillary barriers for the water flow in the vadose zone. We adopt the DFM paradigm where Richards' equation is used to model the unsaturated flow in the matrix, whereas instantaneous hydrostatic equilibration is assumed in the fractures.

Due to the presence of capillary barriers, the water phase in the matrix is allowed to break into the fractures only if a pressure threshold is exceeded at the matrix-fracture interface. Thus, each interface essentially represents a potential seepage face. Indeed, the fact that the pressure head on a given interface is unknown (i.e., is part of the solution) adds a non-trivial layer of complexity to the problem. The resulting model can therefore be classified as a nonlinear coupled PDE-ODE system with variational inequalities.

The resulting set of equations is discretized with MPFA in space and backward Euler in time. Variational inequalities are solved using active level set methods. Moreover, due to the instantaneous equilibration hypothesis, when local fracture networks are present in the domain, equilibrium is enforced between time steps using a water redistribution algorithm.

Finally, we provide numerical examples for the case of a single fracture and the case of a local fracture network.

5.2 Outlook

This thesis has made contributions to three important fields of mathematics, namely: mathematical modeling, computational mathematics, and numerical analysis. Topics ranging from water evaporation in clayey materials under atmospheric conditions to superconvergence properties of residual estimators were discussed.

We strongly believe that our contributions, not only build upon previous work, but also provided a robust ground for further extensions. In the following, we summarize what we believe to be natural extensions of this current work.

Extensions to Paper A: The natural extension for this case is to include fractures in the domain. This will allow us to study processes such as desiccation cracking, which are of high relevance for arid zones. This extension is nontrivial, since fractures are usually not static. Indeed, the common scenario takes place when cracks initiate at some point in the domain (usually a small region of inhomogeneity) and then propagate due to the increase in internal tensile forces in response to desaturation. The fracture propagation problem will then need to be addressed using phase fields or some type of discrete-fracture propagation approach.

Extensions to Paper C: The extensions that one can propose for monodimensional elliptic problems, are also valid for error bounds derived for the case of mixed-dimensional elliptic problems. The mixed-dimensional heat equation being the most natural one. Other extensions, such as the derivation of error bounds for approximations to the mixed-dimensional linear elastic and poroelastic equations, are also possible but considerably more challenging.

Extensions to Paper D: Extensions for this case can be made in several aspects. On the modeling side, the instantaneous hydrostatic equilibrium in the fractures can be replaced with a more realistic dynamical process such as water film flow. On the algorithmic side, a robust and generic water redistribution algorithm could be developed. From the solver side, one can consider solving the set of equations fully implicitly, for which robust mass-conservative regularization techniques must be developed. Finally, a less trivial task would be to extend the model equations to account for mechanical effects.

Bibliography

[1] Aavatsmark, I. (2002). An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences 6*(3-4), 405–432. doi: 10.1023/A:1021291114475.

[2] Abdelghani, F. B., Aubertin, M., Simon, R., and Therrien, R. (2015). Numerical simulations of water flow and contaminants transport near mining wastes disposed in a fractured rock mass. *International Journal of Mining Science and Technology 25*(1), 37–45. doi: 10.1016/j.ijmst.2014.11.003.

[3] Achdou, Y., Bernardi, C., and Coquel, F. (2003). A priori and a posteriori analysis of finite volume discretizations of Darcy's equations. *Numerische Mathematik 96*(1), 17–42. doi: 10.1007/s00211-002-0436-7.

[4] Ahmed, E., Ali Hassan, S., Japhet, C., Kern, M., and Vohralík, M. (2019). A posteriori error estimates and stopping criteria for space-time domain decomposition for two-phase flow between different rock types. *SMAI J. Comput. Math. 5*, 195–227. doi: 10.5802/smai-jcm.47.

[5] Ainsworth, M. (2005). Robust a posteriori error estimation for nonconforming finite element approximation. *SIAM Journal on Numerical Analysis 42*(6), 2320–2341. doi: 10.1137/S0036142903425112.

[6] Allen, M. B. and Murphy, C. (1985). A finite element collocation method for variably saturated flows in porous media. *Numerical Methods for partial differential equations 1*(3), 229–239. doi: 10.1002/num.1690010306.

[7] Angot, P., Boyer, F., and Hubert, F. (2009). Asymptotic and numerical modelling of flows in fractured porous media. *M2AN Math. Model. Numer. Anal. 43*(2), 239–275. doi: 10.1051/m2an/2008052.

[8] Antonietti, P. F., Formaggia, L., Scotti, A., Verani, M., and Verzott, N. (2016). Mimetic finite difference approximation of flows in fractured porous media. *ESAIM: Mathematical Modelling and Numerical Analysis 50*(3), 809–832.

[9] Arbogast, T., Douglas, J., Jr, and Hornung, U. (1990). Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM Journal on Mathematical Analysis 21*(4), 823–836. doi: https://doi.org/10.1137/0521046.

[10] Babuška, I. and Rheinboldt, W. C. (1978). Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal. 15*(4), 736–754. doi: 10.1137/0715049.

[11] Bear, J. (1989). Dynamics of fluids in porous media. Dover Publications Inc. ISBN 9780486656755.

[12] Bebendorf, M. (2003). A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen 22*(4), 751–756. doi: 10.4171/ZAA/1170.

[13] Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. (2014). The hitchhiker's guide to the virtual element method. *Mathematical models and methods in applied sciences 24*(08), 1541–1573. doi: 10.1142/S021820251440003X.

[14] Belhachmi, Z. (2003). A posteriori error estimates for the 3D stabilized mortar finite element method applied to the Laplace equation. *M2AN Math. Model. Numer. Anal. 37*(6), 991–1011. doi: 10.1051/m2an:2003064.

[15] Berre, I., Boon, W. M., Flemisch, B., Fumagalli, A., Gläser, D., Keilegavlen, E., Scotti, A., Stefansson, I., Tatomir, A., Brenner, K., et al. (2021). Verification benchmarks for single-phase flow in three-dimensional fractured porous media. *Advances in Water Resources 147*, 103759. doi: 10.1016/j.advwatres.2020.103759.

[16] Berre, I., Doster, F., and Keilegavlen, E. (2019). Flow in fractured porxous media: A review of conceptual models and discretization approaches. *Transport in Porous Media 130*(1), 215–236. doi: 10.1007/s11242-018-1171-6.

[17] Bertrand, F., Kober, B., Moldenhauer, M., and Starke, G. (2021). Weakly symmetric stress equilibration and a posteriori error estimation for linear elasticity. *Numerical Methods for Partial Differential Equations 37*(4), 2783–2802. doi: 10.1002/num.22741.

[18] Bertrand, F., Moldenhauer, M., and Starke, G. (2019). A posteriori error estimation for planar linear elasticity by stress reconstruction. *Computational Methods in Applied Mathematics 19*(3), 663–679. doi: 10.1515/cmam-2018-0004.

[19] Bertrand, F. and Starke, G. (2021). A posteriori error estimates by weakly symmetric stress reconstruction for the Biot problem. *Computers & Mathematics with Applications 91*, 3–16. doi: 10.1016/j.camwa.2020.10.011.

[20] Biot, M. A. (1941). General theory of three-dimensional consolidation. *Journal of Applied Physics 12*(2), 155–164. doi: 10.1063/1.1712886.

[21] Boffi, D., Brezzi, F., and Fortin, M. (2013). Mixed finite element methods and applications, volume 44 of *Springer Series in Computational Mathematics*. Springer, Heidelberg. ISBN 978-3-642-36518-8; 978-3-642-36519-5. doi: 10.1007/978-3-642-36519-5.

[22] Boon, W. M., Nordbotten, J. M., and Vatne, J. E. (2021). Functional analysis and exterior calculus on mixed-dimensional geometries. *Annali di Matematica Pura ed Applicata (1923-)* page 757–789. doi: 10.1007/s10231-020-01013-1.

[23] Boon, W. M., Nordbotten, J. M., and Yotov, I. (2018). Robust discretization of flow in fractured porous media. *SIAM Journal on Numerical Analysis 56*(4), 2203–2233. doi: 10.1137/17M1139102.

[24] Both, J. W., Pop, I. S., and Yotov, I. (2021). Global existence of weak solutions to unsaturated poroelasticity. *ESAIM: Mathematical Modelling and Numerical Analysis 55*(6), 2849–2897. doi: 10.1051/m2an/2021063.

[25] Braess, D. (2007). Finite elements: Theory, fast solvers, and applications in solid mechanics. Cambridge University Press. ISBN 9780521705189.

[26] Braess, D. and Schöberl, J. (2008). Equilibrated residual error estimator for edge elements. *Mathematics of Computation 77*(262), 651–672. `http://www.jstor.org/stable/40234528`.

[27] Brenner, S. C., Scott, L. R., and Scott, L. R. (2008). The mathematical theory of finite element methods, volume 3. Springer. ISBN 978-1-4757-4338-8.

[28] Brouyère, S. (2006). Modelling the migration of contaminants through variably saturated dual-porosity, dual-permeability chalk. *Journal of Contaminant Hydrology 82*(3-4), 195–219. doi: 10.1016/j.jconhyd.2005.10.004.

[29] Burman, E. and Ern, A. (2006). Continuous interior penalty hp-finite element methods for transport operators. In *Numerical mathematics and advanced applications*, pages 504–511. Springer. `https://www.jstor.org/stable/40234423`.

[30] Carstensen, C. and Funken, S. A. (2000). Constants in Clément-interpolation error and residual based a posteriori estimates in finite element methods. *East-West J. Numer. Math 8*(3), 153–175.

[31] Celia, M. A., Bouloutas, E. T., and Zarba, R. L. (1990). A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research 26*(7), 1483–1496. doi: 10.1029/WR026i007p01483.

[32] Chen, H. and Sun, S. (2017). A residual-based a posteriori error estimator for single-phase Darcy flow in fractured porous media. *Numer. Math. 136*(3), 805–839. doi: 10.1007/s00211-016-0851-9.

[33] Chen, Z., Huan, G., and Ma, Y. (2006). Computational methods for multiphase flows in porous media, volume 2. SIAM. ISBN 9780898716061.

[34] Cochez-Dhondt, S., Nicaise, S., and Repin, S. I. (2009). A posteriori error estimates for finite volume approximations. *Math. Model. Nat. Phenom. 4*(1), 106–122. doi: 10.1051/mmnp/20094105.

[35] Coussy, O. (2004). Poromechanics. John Wiley & Sons. ISBN 9780470849200.

[36] da Veiga, L. B., Lipnikov, K., and Manzini, G. (2014). The mimetic finite difference method for elliptic problems, volume 11. Springer. ISBN 978-3-319-02662-6.

[37] Das, B., Steinberg, S., Weber, S., and Schaffer, S. (1994). Finite difference methods for modeling porous media flows. *Transport in Porous Media 17*, 171–200. doi: 10.1007/BF00624731.

[38] Di Pietro, D. A. and Ern, A. (2011). Mathematical aspects of discontinuous Galerkin methods, volume 69. Springer Science & Business Media. ISBN 978-3-642-22979-4.

[39] Dragila, M. I. and Wheatcraft, S. W. (2001). Free-surface films, pages 217–241. National Academy Press Washington, DC. doi: 10.17226/10102.

[40] Durlofsky, L. J. (1991). Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water resources research 27*(5), 699–708. doi: 10.1029/91WR00107.

[41] Ern, A. and Guermond, J.-L. (2004). Theory and practice of finite elements, volume 159. Springer. ISBN 978-1-4757-4355-5.

[42] Ern, A., Nicaise, S., and Vohralík, M. (2007). An accurate $H(\text{div})$ flux reconstruction for discontinuous Galerkin approximations of elliptic problems. *Comptes Rendus Mathematique 345*(12), 709–712. doi: 10.1016/j.crma.2007.10.036.

[43] Ern, A. and Vohralík, M. (2009). Flux reconstruction and a posteriori error estimation for discontinuous Galerkin methods on general nonmatching grids. *Comptes Rendus Mathematique 347*(7-8), 441–444. doi: 10.1016/j.crma.2009.01.017.

[44] Ern, A. and Vohralík, M. (2015). Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations. *SIAM J. Numer. Anal. 53*(2), 1058–1081. doi: 10.1137/130950100.

[45] Farthing, M. W. and Ogden, F. L. (2017). Numerical solution of Richards' equation: A review of advances and challenges. *Soil Science Society of America Journal 81*(6), 1257–1269. doi: 10.2136/sssaj2017.02.0058.

[46] Flint, A. L., Flint, L. E., Kwicklis, E. M., Bodvarsson, G. S., and Fabryka-Martin, J. M. (2001). Hydrology of Yucca Mountain, Nevada. *Reviews of Geophysics 39*(4), 447–470. doi: 10.1029/1999RG000075.

[47] Formaggia, L., Fumagalli, A., Scotti, A., and Ruffo, P. (2014). A reduced model for Darcy's problem in networks of fractures. *ESAIM: Mathematical Modelling and Numerical Analysis 48*(4), 1089–1116. doi: 10.1051/m2an/2013132.

[48] Formaggia, L., Scotti, A., and Fumagalli, A. (2020). Numerical Methods for Flow in Fractured Porous Media. doi: 10.48550/ARXIV.2003.14279.

[49] Fredlund, D. G. (2006). Unsaturated soil mechanics in engineering practice. *Journal of Geotechnical and Geoenvironmental Engineering 132*(3), 286–321. doi: 10.1061/(ASCE)1090-0241(2006)132:3(286).

[50] Fumagalli, A. and Keilegavlen, E. (2019). Dual virtual element methods for discrete fracture matrix models. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles 74*, 41. doi: 10.2516/ogst/2019008.

[51] Fumagalli, A., Scotti, A., and Formaggia, L. (2021). Performances of the mixed virtual element method on complex grids for underground flow. In *Polyhedral Methods in Geosciences*, pages 299–329. Springer. doi: 10.1007/978-3-030-69363-3_8.

[52] Gatica, G. N. (2014). A simple introduction to the mixed finite element method. *Theory and Applications. Springer Briefs in Mathematics. Springer, London* .

[53] Hecht, F., Mghazli, Z., Naji, I., and Roberts, J. E. (2019). A residual *a posteriori* error estimators for a model for flow in porous media with fractures. *J. Sci. Comput. 79*(2), 935–968. doi: 10.1007/s10915-018-0875-7.

[54] Heinz, C., Moses, B., and Hoffmann, J. (1996). The listings package.

[55] Karakashian, O. A. and Pascal, F. (2003). A posteriori error estimates for a discontinuous Galerkin approximation of second-order elliptic problems. *SIAM Journal on Numerical Analysis 41*(6), 2374–2399. doi: 10.1137/S0036142902405217.

[56] Kavetski, D., Binning, P., and Sloan, S. (2001). Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation. *Advances in water resources 24*(6), 595–605. doi: 10.1016/S0309-1708(00)00076-2.

[57] Keilegavlen, E., Berge, R., Fumagalli, A., Starnoni, M., Stefansson, I., Varela, J., and Berre, I. (2021). Porepy: An open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences 25*(1), 243–265. doi: 10.1007/s10596-020-10002-5.

[58] Keilegavlen, E. and Nordbotten, J. (2015). Finite volume methods for elasticity with weak symmetry. *International Journal for Numerical Methods in Engineering 112*. doi: 10.1002/nme.5538.

[59] Kim, K. Y. (2007). A posteriori error analysis for locally conservative mixed methods. *Math. Comp. 76*(257), 43–66. doi: 10.1090/S0025-5718-06-01903-X.

[60] Kirkland, M. R., Hills, R., and Wierenga, P. (1992). Algorithms for solving Richards' equation for variably saturated soils. *Water Resources Research 28*(8), 2049–2058. doi: 10.1029/92WR00802.

[61] Koohbor, B., Fahs, M., Hoteit, H., Doummar, J., Younes, A., and Belfort, B. (2020). An advanced discrete fracture model for variably saturated flow in fractured porous media. *Advances in Water Resources 140*, 103602. doi: 10.1016/j.advwatres.2020.103602.

[62] Kordilla, J., Sauter, M., Reimann, T., and Geyer, T. (2012). Simulation of saturated and unsaturated flow in karst systems at catchment scale using a double continuum approach. *Hydrology and Earth System Sciences 16*(10), 3909–3923. doi: 10.5194/hess-16-3909-2012.

[63] Kumar, K., Kyas, S., Nordbotten, J. M., and Repin, S. I. (2021). Guaranteed and computable error bounds for approximations constructed by an iterative decoupling of the Biot problem. *Computers & Mathematics with Applications 91*, 122–149. doi: 10.1016/j.camwa.2020.05.005.

[64] Kumar, K., List, F., Pop, I. S., and Radu, F. A. (2020). Formal upscaling and numerical validation of unsaturated flow models in fractured porous media. *Journal of Computational Physics 407*, 109138. doi: 10.1016/j.jcp.2019.109138.

[65] Kuráž, M., Mayer, P., Lepš, M., and Trpkošová, D. (2010). An adaptive time discretization of the classical and the dual porosity model of Richards' equation. *Journal of computational and applied mathematics 233*(12), 3167–3177. doi: 10.1016/j.cam.2009.11.056.

[66] Lazarov, R., Repin, S. I., and Tomar, S. K. (2009). Functional a posteriori error estimates for discontinuous Galerkin approximations of elliptic problems. *Numerical Methods for Partial Differential Equations 25*(4), 952–971. doi: 10.1002/num.20386.

[67] LeVeque, R. J. (2007). Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems. SIAM. ISBN 978-0-898716-29-0.

[68] Lewis, R. B. and Schrefler, B. A. (1998). The finite element method in the static and dynamic deformation and consolidation of porous media. Wiley. ISBN 9780471928096.

[69] Li, X. and Li, D. (2019). A numerical procedure for unsaturated seepage analysis in rock mass containing fracture networks and drainage holes. *Journal of Hydrology 574*, 23–34. doi: 10.1016/j.jhydrol.2019.04.014.

[70] Lie, K.-A. (2019). An introduction to reservoir simulation using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST). Cambridge University Press. doi: 10.1017/9781108591416.

[71] List, F., Kumar, K., Pop, I. S., and Radu, F. A. (2020). Rigorous upscaling of unsaturated flow in fractured porous media. *SIAM Journal on Mathematical Analysis 52*(1), 239–276. doi: 10.1137/18M1203754.

[72] List, F. and Radu, F. A. (2016). A study on iterative methods for solving Richards' equation. *Computational Geosciences 20*(2), 341–353. doi: 10.1007/s10596-016-9566-3.

[73] Liu, R., Li, B., Jiang, Y., and Huang, N. (2016). Mathematical expressions for estimating equivalent permeability of rock fracture networks. *Hydrogeology Journal 24*(7), 1623–1649. doi: 10.1007/s10040-016-1441-8.

[74] Lubliner, J. and Papadopoulos, P. (2016). Introduction to solid mechanics. Springer. ISBN 9783319792620.

[75] Luce, R. and Wohlmuth, B. I. (2004). A local a posteriori error estimator based on equilibrated fluxes. *SIAM Journal on Numerical Analysis 42*(4), 1394–1414. doi: 10.1137/S0036142903433790.

[76] Martin, V., Jaffré, J., and Roberts, J. E. (2005). Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput. 26*(5), 1667–1691. doi: 10.1137/S1064827503429363.

[77] Masciopinto, C. and Caputo, M. C. (2011). Modeling unsaturated–saturated flow and nickel transport in fractured rocks. *Vadose Zone Journal 10*(3), 1045–1057. doi: 10.2136/vzj2010.0087.

[78] Mghazli, Z. and Naji, I. (2019). Guaranteed *a posteriori* error estimates for a fractured porous medium. *Math. Comput. Simulation 164*, 163–179. doi: 10.1016/j.matcom.2019.02.002.

[79] Mualem, Y. (1976). A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water resources research 12*(3), 513–522. doi: 10.1029/WR012i003p00513.

[80] Neittaanmäki, P. and Repin, S. I. (2004). Reliable methods for computer simulation, volume 33 of *Studies in Mathematics and its Applications*. Elsevier Science B.V, Amsterdam. ISBN 0-444-51376-0. Error control and a posteriori estimates.

[81] Nordbotten, J. M. (2014). Cell-centered finite volume discretizations for deformable porous media. *International Journal for Numerical Methods in Engineering 100*(6), 399–418. doi: 10.1002/nme.4734.

[82] Nordbotten, J. M. (2016). Stable cell-centered finite volume discretization for Biot equations. *SIAM Journal on Numerical Analysis 54*(2), 942–968. doi: 10.1137/15m1014280.

[83] Nordbotten, J. M., Boon, W. M., Fumagalli, A., and Keilegavlen, E. (2019). Unified approach to discretization of flow in fractured porous media. *Computational Geosciences 23*(2), 225–237. doi: 10.1007/s10596-018-9778-9.

[84] Oden, J. T. and Prudhomme, S. (2001). Goal-oriented error estimation and adaptivity for the finite element method. *Comput. Math. Appl. 41*(5-6), 735–756. doi: 10.1016/S0898-1221(00)00317-5.

[85] Pauly, D. and Repin, S. (2010). Two-sided a posteriori error bounds for electromagnetostatic problems. *Journal of Mathematical Sciences 166*(1), 53–62. doi: 10.1007/s10958-010-9844-x.

[86] Pauly, D. and Repin, S. I. (2009). Functional a posteriori error estimates for elliptic problems in exterior domains. *Journal of Mathematical Sciences 162*(3), 393–406. doi: 10.1007/s10958-009-9643-4.

[87] Payne, L. E. and Weinberger, H. F. (1960). An optimal Poincaré inequality for convex domains. *Arch. Rational Mech. Anal. 5*, 286–292 (1960). doi: 10.1007/BF00252910.

[88] Pechstein, C. and Scheichl, R. (2013). Weighted Poincaré inequalities. *IMA J. Numer. Anal. 33*(2), 652–686. doi: 10.1093/imanum/drs017.

[89] Pencheva, G. V., Vohralík, M., Wheeler, M. F., and Wildey, T. (2013). Robust a posteriori error control and adaptivity for multiscale, multinumerics, and mortar coupling. *SIAM J. Numer. Anal. 51*(1), 526–554. doi: 10.1137/110839047.

[90] Pinder, G. F. and Celia, M. A. (2006). Subsurface hydrology. John Wiley & Sons. ISBN 9780471742432.

[91] Pinder, G. F. and Gray, W. G. (2008). Essentials of multiphase flow and transport in porous media. John Wiley & Sons. ISBN 9780470317624.

[92] Quarteroni, A. and Valli, A. (1994). Numerical approximation of partial differential equations, volume 23 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin. ISBN 3-540-57111-6.

[93] Radcliffe, D. E. and Simunek, J. (2018). Soil physics with HYDRUS: Modeling and applications. CRC press.

[94] Rathfelder, K. and Abriola, L. M. (1994). Mass conservative numerical solutions of the head-based Richards equation. *Water Resources Research 30*(9), 2579–2586. doi: 10.1029/94WR01302.

[95] Rathmair, M. (2019). On how Poincaré inequalities imply weighted ones. *Monatsh. Math. 188*(4), 753–763. doi: 10.1007/s00605-019-01266-w.

[96] Repin, S. I. (1997). A posteriori error estimation for variational problems with power growth functionals based on duality theory. *Zapiski Nauchnych Seminarov POMI 249*, 244–255.

[97] Repin, S. I. (1999). A posteriori error estimation for approximate solutions of variational problems by duality theory. In *Proceedings of ENUMATH*, volume 97, pages 524–531.

[98] Repin, S. I. (1999). A unified approach to a posteriori error estimation based on duality error majorants. *Mathematics and Computers in Simulation 50*(1-4), 305–321. doi: 10.1016/S0378-4754(99)00081-6.

[99] Repin, S. I. (2002). Estimates of deviations from exact solutions of initial-boundary value problem for the heat equation. *Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei (9) Mat. Appl. 13*(2), 121–133. `http://dml.mathdoc.fr/item/RLIN_2002_9_13_2_121_0`.

[100] Repin, S. I. (2002). A posteriori estimates for the Stokes problem. *Journal of Mathematical Sciences 109*(5), 1950–1964. doi: 10.1023/A:1014400626472.

[101] Repin, S. I. (2003). Two-sided estimates of deviation from exact solutions of uniformly elliptic equations. *Translations of the American Mathematical Society-Series 2 209*, 143–172. doi: 10.1090/trans2/209/06.

[102] Repin, S. I. (2005). Estimates of deviations from exact solutions for boundary-value problems with incompressibility condition. *St. Petersburg Mathematical Journal 16*(5), 837–862. doi: 10.1090/S1061-0022-05-00882-4.

[103] Repin, S. I. (2007). Functional a posteriori estimates for Maxwell's equation. *Journal of Mathematical Sciences 142*(1). doi: 10.1007/s10958-007-0091-8.

[104] Repin, S. I. (2008). Advanced forms of functional a posteriori error estimates for elliptic problems. *Russian J. Numer. Anal. Math. Modelling 23*(5), 505–521. doi: 10.1515/RJNAMM.2008.029.

[105] Repin, S. I. (2008). A posteriori estimates for partial differential equations, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin. ISBN 978-3-11-019153-0. doi: 10.1515/9783110203042.

[106] Repin, S. I. (2012). Computable majorants of constants in the Poincaré and Friedrichs inequalities. volume 186, pages 307–321. doi: 10.1007/s10958-012-0987-9. Problems in mathematical analysis. No. 66.

[107] Repin, S. I. and Sauter, S. (2006). Functional a posteriori estimates for the reaction–diffusion problem. *Comptes Rendus Mathematique 343*(5), 349–354. doi: 10.1016/j.crma.2006.06.024.

[108] Repin, S. I., Sauter, S., and Smolianski, A. (2004). A posteriori error estimation for the Poisson equation with mixed Dirichlet/Neumann boundary conditions. *Journal of Computational and Applied Mathematics 164*, 601–612. doi: 10.1016/S0377-0427(03)00491-6.

[109] Repin, S. I., Sauter, S., and Smolianski, A. (2007). Two-sided a posteriori error estimates for mixed formulations of elliptic problems. *SIAM Journal on Numerical Analysis 45*(3), 928–945. doi: 10.1137/050641533.

[110] Repin, S. I. and Tomar, S. K. (2011). Guaranteed and robust error bounds for nonconforming approximations of elliptic problems. *IMA Journal of Numerical Analysis 31*(2), 597–615. doi: 10.1093/imanum/drp037.

[111] Repin, S. I. and Valdman, J. (2008). Functional a posteriori error estimates for problems with nonlinear boundary conditions. *Journal of Numerical Mathematics* doi: 10.1515/JNUM.2008.003.

[112] Repin, S. I. and Valdman, J. (2018). Error identities for variational problems with obstacles. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 98*(4), 635–658. doi: 10.1002/zamm.201700105.

[113] Repin, S. I. and Xanthis, L. S. (1996). A posteriori error estimation for elastoplastic problems based on duality theory. *Computer Methods in Applied Mechanics and Engineering 138*(1-4), 317–339. doi: 10.1016/S0045-7825(96)01136-X.

[114] Richards, L. A. (1931). Capillary conduction of liquids through porous mediums. *Physics 1*(5), 318–333. doi: 10.1063/1.1745010.

[115] Rivière, B. (2008). Discontinuous Galerkin methods for solving elliptic and parabolic equations: theory and implementation. SIAM. ISBN 978-0-89871-656-6.

[116] Robineau, T., Tognelli, A., Goblet, P., Renard, F., and Schaper, L. (2018). A double medium approach to simulate groundwater level variations in a fissured karst aquifer. *Journal of Hydrology 565*, 861–875. doi: 10.1016/j.jhydrol.2018.09.002.

[117] Schrefler, B., Shiomi, T., Chan, A., Zienkiewicz, O., and Pastor, M. (1999). Computational geomechanics with special reference to earthquake engineering. Wiley, Chichester. ISBN 0471982857.

[118] Schwarzkopf, O. (1995). The extensible drawing editor Ipe. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 410–411. `https://ipe.otfried.org`.

[119] Scott, R. B. and Bonk, J. (1984). Preliminary geologic map of Yucca Mountain, Nye County, Nevada with geologic sections. Technical report, Geological Survey. doi: 10.3133/ofr84494.

[120] Spiridonov, D., Vasilyeva, M., and Chung, E. T. (2020). Generalized Multiscale Finite Element method for multicontinua unsaturated flow problems in fractured porous media. *Journal of Computational and Applied Mathematics 370*, 112594. doi: 10.1016/j.cam.2019.112594.

[121] Su, G. W., Nimmo, J. R., and Dragila, M. I. (2003). Effect of isolated fractures on accelerated flow in unsaturated porous rock. *Water Resources Research 39*(12). doi: 10.1029/2002WR001691.

[122] Therrien, R. and Sudicky, E. (1996). Three-dimensional analysis of variably-saturated flow and solute transport in discretely-fractured porous media. *Journal of Contaminant Hydrology 23*(1-2), 1–44. doi: 10.1016/0169-7722(95)00088-7.

[123] Tocci, M. D., Kelley, C., and Miller, C. T. (1997). Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines. *Advances in Water Resources 20*(1), 1–14. doi: 10.1016/S0309-1708(96)00008-5.

[124] Tokunaga, T. K. and Wan, J. (1997). Water film flow along fracture surfaces of porous rock. *Water Resources Research 33*(6), 1287–1295. doi: 10.1029/97WR00473.

[125] Tokunaga, T. K. and Wan, J. (2001). Approximate boundaries between different flow regimes in fractured rocks. *Water Resources Research 37*(8), 2103–2111. doi: 10.1029/2001WR000245.

[126] Tokunaga, T. K., Wan, J., and Sutton, S. R. (2000). Transient film flow on rough fracture surfaces. *Water Resources Research 36*(7), 1737–1746. doi: 10.1029/2000WR900079.

[127] van Genuchten, M. T. (1980). A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal 44*(5), 892–898. doi: 10.2136/sssaj1980.03615995004400050002x.

[128] Varela, J., Ahmed, E., Keilegavlen, E., Nordbotten, J. M., and Radu, F. A. (2021). A posteriori error estimates for hierarchical mixed-dimensional elliptic equations. Preprint at `https://arxiv.org/abs/2101.08331`.

[129] Varela, J., Gasda, S. E., Keilegavlen, E., and Nordbotten, J. M. (2021). A Finite-Volume-Based Module for Unsaturated Poroelasticity, page 515–548. Cambridge University Press. doi: 10.1017/9781009019781.019.

[130] Varela, J., Keilegavlen, E., Nordbotten, J. M., and Radu, A. F. (2022). A model for unsaturated flow in the presence of fractures acting as capillary barriers. In preparation.

[131] Verfürth, R. (1999). A review of a posteriori error estimation techniques for elasticity problems. *Comput. Methods Appl. Mech. Engrg. 176*(1-4), 419–440. doi: 10.1016/S0045-7825(98)00347-8. New advances in computational methods (Cachan, 1997).

[132] Vohralík, M. (2005). On the discrete Poincaré-Friedrichs inequalities for nonconforming approximations of the Sobolev space H^1. *Numer. Funct. Anal. Optim. 26*(7-8), 925–952. doi: 10.1080/01630560500444533.

[133] Vohralík, M. (2007). A posteriori error estimates for finite volume and mixed finite element discretizations of convection-diffusion-reaction equations. In *Paris-Sud Working Group on Modelling and Scientific Computing 2006–2007*, volume 18 of *ESAIM Proc.*, pages 57–69. EDP Sci., Les Ulis. doi: 10.1051/proc:071806.

[134] Vohralík, M. (2007). A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations. *SIAM J. Numer. Anal. 45*(4), 1570–1599. doi: 10.1137/060653184.

[135] Vohralík, M. (2010). Unified primal formulation-based a priori and a posteriori error analysis of mixed finite element methods. *Math. Comp. 79*(272), 2001–2032. doi: 10.1090/S0025-5718-2010-02375-0.

[136] Vohralík, M. (2013). A posteriori error estimates for efficiency and error control in numerical simulations. *Lecture Notes. Paris: Université Pierre et Marie Curie* `https://who.rocq.inria.fr/Martin.Vohralik/Enseig/APost/a_posteriori.pdf`.

[137] Šimůnek, J. and Bradford, S. A. (2008). Vadose zone modeling: Introduction and importance. *Vadose Zone Journal 7*(2), 581–586. doi: 10.2136/vzj2008.0012.

[138] Wang, J. and Narasimhan, T. N. (1985). Hydrologic mechanisms governing fluid flow in a partially saturated, fractured, porous medium. *Water Resources Research 21*(12), 1861–1874. doi: 10.1029/WR021i012p0186.

[139] Warren, J. and Root, P. J. (1963). The behavior of naturally fractured reservoirs. *Society of Petroleum Engineers Journal 3*(03), 245–255. doi: 10.2118/426-PA.

[140] Wheeler, M. F. and Yotov, I. (2005). A posteriori error estimates for the mortar mixed finite element method. *SIAM J. Numer. Anal. 43*(3), 1021–1042. doi: 10.1137/S0036142903431687.

[141] White, F. M. and Corfield, I. (2006). Viscous fluid flow. McGraw-Hill New York, 3rd edition. ISBN 9780072402315.

[142] Wohlmuth, B. I. (1999). Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers. *SIAM J. Numer. Anal. 36*(5), 1636–1658. doi: 10.1137/S0036142997330512.

[143] Zha, Y., Yang, J., Yin, L., Zhang, Y., Zeng, W., and Shi, L. (2017). A modified Picard iteration scheme for overcoming numerical difficulties of simulating infiltration into dry soil. *Journal of Hydrology 551*, 56–69. doi: 10.1016/j.jhydrol.2017.05.053.

[144] Zienkiewicz, O. C. and Zhu, J. Z. (1987). A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Numer. Methods Engrg. 24*(2), 337–357. doi: 10.1002/nme.1620240206.

Part II

Scientific Results

Paper A

A Finite-Volume-Based Module for Unsaturated Poroelasticity

Varela, J., Gasda, S. E., Keilegavlen, E., and Nordbotten, J. M.

In K. Lie and O. Møyner (editors)

Advanced Modeling with the MATLAB Reservoir Simulation Toolbox, 515–548. Cambridge: Cambridge University Press. 2021.

doi: 10.1017/9781009019781.019

13

A Finite-Volume-Based Module for Unsaturated Poroelasticity

JHABRIEL VARELA, SARAH E. GASDA, EIRIK KEILEGAVLEN, AND JAN MARTIN NORDBOTTEN

Abstract

In this chapter, we present `fv-unsat`, a multipoint finite-volume-based solver for unsaturated flow in deformable and nondeformable porous media. The latter is described using the mixed form of Richards' equation, whereas the former by the equations of unsaturated linear poroelasticity. The module aims at flexibility, relying heavily on discrete operators and equations, exploiting the automatic differentiation framework provided by the MATLAB Reservoir Simulation Toolbox (MRST). Our examples cover two numerical convergence tests and two three-dimensional practical applications, including the water infiltration process in a nondeformable soil column and a realistic desiccation process of a deformable clay sample using atmospheric boundary conditions. The resulting convergence rates are in agreement with previously reported rates for single-phase models, and the practical applications capture the physical processes accurately.

13.1 Introduction

The unsaturated zone has been a constant focus of attention by the industrial and research communities due to its high relevance in areas such as environmental sciences, hydrogeology, soil mechanics, and agriculture. Relevant natural and anthropogenic processes take place in this zone; transmission of water from the atmosphere to the saturated zone via infiltration or precipitation, support of plants via root uptake, active return of water from the subsurface to the atmosphere via evapotranspiration, drying of soils during drought seasons, extraction of groundwater via wells, construction and operations of dams, etc. [46].

Although many of these processes can be studied by only taking into account the simultaneous flow of water and air, some of them, such as the desiccation of muddy soils, require the incorporation of the deformation effects due to the strong coupling between flow and mechanics. This gives rise to a nonlinear coupled flow/mechanical set of partial differential equations. Under the assumption of small deformations and linear constitutive relations for the mechanical behavior of the soils, this set of equations can be expressed as a natural extension of Biot's equations of poroelasticity [26], for which recently global existence of the weak solution has been proven [9].

Given the complexity of the resulting model, it is imperative to use robust discretization techniques in a flexible computational setting. The fully coupled system is not commonly treated by numerical software, and the few available codes are limited to the use of finite-element methods [26] or mixed finite-element methods [7]. In this module, we propose the use of finite-volume methods (FVM), which are inherently conservative while keeping the advantages of robust discretization schemes; i.e., flexibility in representing complex domains.

In the FVM framework, two-point flux approximation (TPFA) is the most widely used method for discretizing scalar equations. However, TPFA is only consistent for K-orthogonal grids [1] and cannot be directly applied to vector equations. The first of these issues can be addressed with the multipoint flux approximation (MPFA) method [1], and the second with the multipoint stress approximation (MPSA) method [33]. Both methods are currently well established in academia and slowly taking hold in industry.

As we mentioned before, computational flexibility is an important aspect of a module when it comes to solving a broad range of applications. With this goal in mind, we have written `fv-unsat` taking advantage of the high-level coding capacities of the MATLAB Reservoir Simulation Toolbox (MRST), such as automatic differentiation [24]. This module is based on the work of [44] and requires the module `fvbiot`, which provides the discrete MPFA and MPSA operators, along with the coupling operators for the flow/mechanical problem.[1]

The existing implementation of `fv-unsat` does not cover the full width of modeling options of the governing equations; e.g., mixed and time-dependent boundary conditions for the mechanical problem. If the interested user needs to include these setups, we recommend the Python-based framework PorePy [20], which provides a more general implementation of MPSA for poroelastic problems.

Our notation follows MRST's conventions [28]. In physical space, x represents a scalar, $\vec{x}$ a vector, and $\mathbf{x}$ a tensor. In a discrete sense, $\boldsymbol{x}$ is a vector and $\texttt{ope}(\boldsymbol{x})$ is a discrete operator acting on $\boldsymbol{x}$; i.e., the matrix–vector product between `ope` and $\boldsymbol{x}$.

The chapter is structured as follows: in Section 13.2 we provide the continuous formulations for the unsaturated flow in nondeformable (Richards' equation) and

[1] After this chapter was written, `mpsaw`, a new and improved implementation of the MPSA-W method has been released with the core MRST distribution. The module can also be downloaded separately at https://bitbucket.org/mrst/mpsaw/src/master/.

deformable (unsaturated poroelasticity) porous media; in Section 13.3 we introduce the MPFA and MPSA methods, together with the discrete operators and the discrete equations; in Section 13.4 we present two numerical convergence tests and two practical applications with in-depth explanation regarding the module; and in Section 13.5 we draw the conclusions.

13.2 Governing Equations

In this section, we provide the set of equations that governs the physical processes in the continuous domain. We do not attempt to provide detailed derivations of these equations; for that matter we refer to [13, 26, 37].

13.2.1 Richards' Equation

Richards' equation models the flow of water in partially saturated porous media, and it is based on the assumption of inviscid air. This assumption is supported by the contrast in physical properties between water and air; e.g., at atmospheric conditions air is three orders of magnitude less dense and two orders less viscous than water [37]. Because the unsaturated zone is connected to the atmosphere, it is reasonable to assume that the air remains at atmospheric pressure. This is usually referred to as the Richards assumption and it was first proposed in [40].

We start the derivation by stating the mass-balance equation for the water phase

$$\frac{\partial\left(\rho_w S_w n\right)}{\partial t} + \nabla \cdot \left(\rho_w S_w n \vec{v}_w\right) = \dot{m}_w. \tag{13.1}$$

Here, ρ_w and S_w are the density and saturation, n is the porosity of the porous medium, $\vec{v}_w$ is the water velocity, and $\dot{m}_w$ is the rate of external addition/subtraction of fluid mass per volume of representative elementary volume [5]. If water and solid phases are assumed to be incompressible, we can rewrite (13.1) as

$$n\rho_w \frac{\partial S_w}{\partial t} + \rho_w \nabla \cdot \left(S_w n \vec{v}_{ws}\right) = \dot{m}_w, \tag{13.2}$$

where $\vec{v}_{ws} := \vec{v}_w - \vec{v}_s$ is the velocity of the water with respect to the solids [26]. We recognize the term $S_w n \vec{v}_{ws}$ as the Darcy velocity of the water phase, given by

$$\vec{q}_w = S_w n \vec{v}_{ws} = -\frac{\mathbf{k}}{\mu_w} k_{rw} \left(\nabla p_w - \rho_w \vec{g}\right), \tag{13.3}$$

where $\mathbf{k}$ is the intrinsic permeability tensor, μ_w is the water dynamic viscosity, p_w is the water pressure, and $\vec{g}$ is the gravity acceleration considered positive downwards. The relative permeability $k_{rw} \in [0, 1]$ is included to account for the simultaneous flow of water and air.

In hydrology, it is common to express Darcy's law (13.3) in terms of heads,

$$\vec{q}_w = -\mathbf{K}_w^{sat} k_{rw} \nabla \left(\psi_w + \zeta\right). \tag{13.4}$$

Here, $\psi_w = (p_w - p_a)/(\rho_w g)$ is the water pressure head (relative to the atmospheric pressure p_a), $\zeta = z - z_0$ is the elevation head (e.g., the height from a reference to the measurement point), and $\mathbf{K}_w^{sat} := \rho_w g \mathbf{k}/\mu_w$ is the hydraulic conductivity at saturated conditions [17, 37].

If Richards' assumption holds true, the air pressure is constant and equal to p_a, which is assumed to be zero. This allows us to write the equations purely in terms of the water phase. Note that the capillary pressure is still present; i.e., $p_c = p_a - p_w = - p_w$. To get to the final expression, we substitute (13.4) into (13.2) and divide by ρ_w:

$$\frac{\partial \theta_w}{\partial t} - \nabla \cdot \left(\mathbf{K}_w^{sat} k_{rw} \nabla \left(\psi_w + \zeta\right)\right) = \frac{\dot{m}_w}{\rho_w}, \tag{13.5}$$

where we introduce the water content $\theta_w := n S_w$ and use the fact that the porosity is constant. Equation (13.5) is referred to as the mixed-based form of Richards' equation. The term "mixed" suggests that both the water content and the pressure head appear explicitly in the equation. Alternative formulations include the pressure head–based and the water content–based forms [37]. On a continuous level, all forms of Richards' equation are equivalent under strictly unsaturated conditions. However, on a discrete level, the ψ-based lacks conservative properties [12] and the θ-based fails to converge when $S_w \to 1$ [38]. Therefore, in this module, we employ the mixed-based formulation.

In unsaturated systems, the usual practice is to express k_{rw} and θ_w in terms of ψ_w. These relationships are called soil/water retention curves (SWRCs). One such family of curves is the van Genuchten–Mualem (vG-M) model, originally proposed in [43]. For the vG-M model, the water content is given by

$$\theta_w = \begin{cases} \dfrac{\theta_w^s - \theta_w^r}{\left[1 + \left(\alpha_v \left|\psi_w\right|\right)^{n_v}\right]^{m_v}} + \theta_w^r, & \psi_w < 0, \\ \theta_s^w, & \psi_w \geq 0, \end{cases} \tag{13.6}$$

where θ_w^s and θ_w^r are the water content at saturated and residual conditions and α_v, n_v, and m_v are fitting parameters. Note that $\psi_w < 0$ denotes unsaturated conditions. The relative permeability is given by

$$k_{rw} = \begin{cases} \dfrac{\left\{1 - \left(\alpha_v \left|\psi_w\right|\right)^{n_v - 1} \left[1 + \left(\alpha_v \left|\psi_w\right|\right)^{n_v}\right]^{-m_v}\right\}^2}{\left[1 + \left(\alpha_v \left|\psi_w\right|\right)^{n_v}\right]^{m_v/2}}, & \psi_w < 0, \\ 1, & \psi_w \geq 0. \end{cases} \tag{13.7}$$

We also introduce the specific moisture capacity $C_\psi := \mathrm{d}\theta_w/\mathrm{d}\psi_w$:

$$C_\psi = \begin{cases} -\dfrac{m_v n_v \psi_w \left(\theta_w^s - \theta_w^r\right)(\alpha_v\,|\psi_w|)^{n_v}}{|\psi_w|^2\left[(\alpha_v\,|\psi_w|)^{n_v}+1\right]^{m_v+1}}, & \psi_w < 0, \\ 0, & \psi_w \geq 0. \end{cases} \tag{13.8}$$

13.2.2 Unsaturated Poroelasticity

Herein, we present the equations that govern an unsaturated poroelastic medium as a natural extension of Biot's equations [26]. The momentum conservation for an unsaturated poroelastic medium reads

$$\nabla \cdot \boldsymbol{\sigma}_t + \big((1-n)\rho_s + nS_w\rho_w\big)\vec{g} = 0, \tag{13.9}$$

where $\boldsymbol{\sigma}_t$ is the total stress tensor and ρ_s the density of the solids, with the second term representing the body forces. For a poroelastic medium, the total stress has two contributions: the part that acts on the solid skeleton and the part that acts on the fluid. The relation is given by the extended principle of effective stress [16],

$$\boldsymbol{\sigma}_t = \boldsymbol{\sigma}_e - \alpha p_w S_w \mathbf{I}. \tag{13.10}$$

The term $\boldsymbol{\sigma}_e$ is the effective stress tensor, and it is responsible for causing the actual deformation of the material; thus the name "effective" [45]. The second term affects the pore pressure of the fluid, where α is the Biot coupling coefficient and $\mathbf{I}$ is the identity tensor. The negative sign follows the convention that tensile forces are positive whereas compressive forces are negative [31]. Substitution of (13.10) into (13.9) gives the final version of the unsaturated momentum balance equation,

$$\nabla \cdot \boldsymbol{\sigma}_e - \alpha\nabla\left(S_w p_w\right) + \big((1-n)\rho_s + nS_w\rho_w\big)\vec{g} = 0. \tag{13.11}$$

Assuming small deformations and a linear stress–strain relation, the effective stress $\boldsymbol{\sigma}_e$ can be related to the displacement field $\vec{u}$ employing the generalized Hooke's law

$$\boldsymbol{\sigma}_e = \mathbf{C} : \tfrac{1}{2}\big(\nabla\vec{u} + (\nabla\vec{u})^T\big), \tag{13.12}$$

where $\mathbf{C}$ is the stiffness matrix, a fourth-order tensor in its most general form. For the particular case of an isotropic medium, (13.12) can be written as

$$\boldsymbol{\sigma}_e = \mu_s\big(\nabla\vec{u} + (\nabla\vec{u})^T\big) + \lambda_s\left(\nabla \cdot \vec{u}\right)\mathbf{I}, \tag{13.13}$$

where λ_s and μ_s are the first and second Lamé parameters [30].

A statement of the mass conservation principle for both phases (water and solid skeleton) can be used to derive the unsaturated storage equation (see [44] for a detailed derivation):

$$\xi(S_w)\frac{\partial p_w}{\partial t} + \chi(S_w, p_w)\frac{\partial S_w}{\partial t} + \alpha S_w \frac{\partial}{\partial t}\left(\nabla \cdot \vec{u}\right) + \nabla \cdot \vec{q}_w = \frac{\dot{m}_w}{\rho_w}, \tag{13.14}$$

where $\xi := (\alpha - n)\, C_s S_w^2 + n C_w S_w$ and $\chi := (\alpha - n)\, C_s S_w p_w + n$ are compressibility-like terms. In (13.14), the first two terms represent accumulation terms, the third term is the change of strain at constant saturation, the fourth term is the divergence of the Darcy velocity, and the last terms are sources or sinks of water [41].

Note that the above set of equations is written in terms of (p_w, S_w) instead of (ψ_w, θ_w). Because the SWRC is expressed in terms of the latter variables, we have to adapt the original vG-M model to be consistent with the (p_w, S_w) representation. This can be easily achieved using the following relations:

$$\psi_w = \frac{p_w}{\rho_w g}, \qquad \theta_w = n S_w, \qquad C_\psi = n \rho_w g C_p,$$

where all of the terms have been previously introduced, except the specific saturation capacity $C_p := \partial S_w / \partial p_w$.

13.2.3 Boundary and Initial Conditions

To close the systems of partial differential equations, we must provide boundary and initial conditions for the flow and mechanical problems. For the flow problem, two types of conditions can be specified: pressure (or pressure head) and fluxes. For the mechanical problem, we can impose displacement and traction force vectors. Denoting Ω the domain of interest and $\partial\Omega$ its boundary, the boundary conditions are given by

$$p_w = g_{p,D} \quad \text{on} \quad \Gamma_{p,D}, \tag{13.15}$$

$$\vec{q}_w \cdot \vec{n} = g_{p,N} \quad \text{on} \quad \Gamma_{p,N}, \tag{13.16}$$

$$\vec{u} = g_{\vec{u},D} \quad \text{on} \quad \Gamma_{\vec{u},D}, \tag{13.17}$$

$$\boldsymbol{\sigma}_t \cdot \vec{n} = g_{\vec{u},N} \quad \text{on} \quad \Gamma_{\vec{u},N}, \tag{13.18}$$

where $\vec{n}$ is the normal vector pointing outwards, and the subindices D and N denote Dirichlet and Neumann boundary conditions. The boundary of the domain is given by $\partial\Omega = \Gamma_D \cup \Gamma_N$ with $\Gamma_D \cap \Gamma_N = \emptyset$.

The initial conditions are specified as

$$p_w = p_{w,0} \quad \text{for} \quad t = 0, \tag{13.19}$$

$$\vec{u} = \vec{u}_0 \quad \text{for} \quad t = 0. \tag{13.20}$$

13.3 Discretization and Implementation

This section is devoted to the discretization techniques and computational implementation. First, we briefly introduce the numerical methods; e.g., MPFA/MPSA finite-volume (FV) schemes. Then, we employ the discrete operators to derive the discrete version of the governing equations. Finally, we describe the general strategy for solving the resulting nonlinear set of equations. In particular, we discuss the workflow of the iterative solver and the timestepping algorithm.

13.3.1 MPFA and MPSA

Before writing the discrete version of the governing equations, we briefly introduce the MPFA and MPSA methods. From an implementation standpoint, the discretization routines for both methods are provided by the third-party module `fvbiot`. Nevertheless, we should remark that TPFA is the standard scheme employed in MRST for the discretization of flow equations. In addition, MRST provides an alternative MPFA implementation based on the mimetic method available through the `mpfa` module (see section 6.4 of the MRST textboox [28] for further details). Because both techniques (MPFA and MPSA) are well established in the literature we do not go in-depth. We refer to [1, 4, 23] for an introduction to MPFA and to [21, 33] for an introduction to MPSA.

MPFA

In an FVM framework applied to the flow problem, we aim to discretize the integrated version of (13.4) over a face. For a cell-centered FVM, we use the cell-centered pressures to estimate the fluxes across the faces; i.e., $Q = \int_S \vec{q} \cdot \vec{n}\, \mathrm{d}\Sigma$. Hence, for a given face, we have to define the number of points to be considered for approximating Q.

The simplest choice is to consider two points, say, 1 and 2 from top Figure 13.1. This technique is referred to as TPFA, with the flux across the shared face j given by

$$Q_j \approx \lambda_j T_j (p_1 - p_2), \tag{13.21}$$

where Q_j is the water flux, $\lambda_j = k_{rw,j}/\mu_w$ is the water mobility, and T_j is the transmissibility. For readability, we drop the subindices denoting the water phase.

The MPFA method is a generalization of the TPFA method, where instead of using two points of information, we use a larger set of potentials (see bottom of Figure 13.1). For the MPFA method, the flux can be approximated as

$$Q_j \approx \lambda_j \sum_{i \in \mathcal{I}} t_{ij} p_i, \tag{13.22}$$

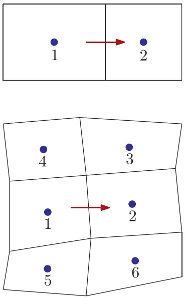


Figure 13.1 Flux approximations: TPFA (top) relies on first neighbors only, whereas MPFA (bottom) also includes second neighbors. Adapted from [1]

where t_{ij} are the transmissibility coefficients satisfying $\sum_{i \in \mathcal{I}} t_{ij} = 0$, and $\mathcal{I}$ is the set of of cells used to approximate the flux through the face j. The size of the set $\mathcal{I}$ depends on the dimensionality of the problem and the type of element employed. For quadrilaterals, the set $\mathcal{I}$ consists of six neighbors. With this increase in accuracy, MPFA results in a consistent discretization method compared to TPFA (which gives nonphysical results when applied to non-K-orthogonal grids) [1]. An interesting discussion regarding consistency of the numerical methods can be found in chapter 6 of the MRST textbook [28].

Mobilities λ_j are evaluated at the faces using either an arithmetic mean or an upstream weighting of the cell-centered values. The arithmetic mean implies $\lambda_j = (\lambda_1 + \lambda_2)/2$, whereas the upstream weighting is based on the flux direction; i.e., $\lambda_j = \lambda_1$ if $\sum_{i \in \mathcal{I}} t_{ij} p_i > 0$ ($T_j(p_1 - p_2) > 0$ for TPFA) and $\lambda_j = \lambda_2$ otherwise [1].

Provided that the pressures are known, both problems are reduced to determining T_j and t_{ij}. For TPFA, these are given by the harmonic average; however, finding t_{ij} is more complicated. Several families of MPFA methods obtain t_{ij} in different ways. The key difference among the methods lies in the way interaction regions are constructed and continuity points selected. Interaction regions are composed of the relevant neighboring cells and identified using the dual of the mesh (see Figure 13.2). We refer to [14] for an excellent discussion on the topic. In this module, we use the MPFA-O method, implemented in the `fvbiot` module.

MPSA

In recent years, the MPSA method was developed as a generalization of the MPFA method applied to vector equations, such as the Navier–Lamé equations [21, 33] or

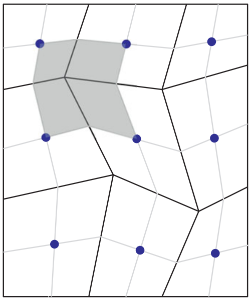


Figure 13.2 Dual mesh (light gray), conservation cells (black), and interaction region for the O-method (shaded). Adapted from [33]

the Biot equations [35]. MPSA uses the displacements $\vec{u}$ located at the cell centers as the only primary unknowns with the traction forces $\vec{T} = \int_S \boldsymbol{\sigma}_e \cdot \vec{n}\, \mathrm{d}\Sigma$ defined on the faces. On each face, the traction is linearly approximated by

$$\vec{T}_j \approx \sum_{i \in \mathcal{I}} s_{ij} \vec{u}_i, \tag{13.23}$$

where $s_{ij} = -s_{ij}$ are the stress weight tensors, and $\mathcal{I}$ is the set of neighboring cells to the face j. In essence, (13.23) can be seen as a local version of Hooke's law (13.12). Now the problem is reduced to the calculation of the stress weight tensors s_{ij} for each face of the domain. Similar to MPFA, there are several ways to estimate s_{ij} depending on the continuity points. The procedure for calculating the stress weights is beyond the scope of this chapter; we refer to [21, 33] for further details. The `fvbiot` module provides the MPSA-W version from [21], which is used herein.

13.3.2 Discretization

Herein we introduce the discrete MPFA/MPSA operators and discretize the governing equations. The way discrete operators are defined in our module is heavily inspired by MRST's rapid prototyping philosophy. In particular, they are in agreement with the basic structure of the simulators based on automatic differentiation utilized in MRST; see, for example, chapter 7 of the MRST textbook [28] for an excellent introduction. As the reader will note, this enables us to write the discrete equations in a fairly compact way, while simultaneously providing a concise way to structure the code.

Table 13.1 *Definition of the MPFA/MPSA operators.*

Description	Mapping	Operator dimension
Flux	`F` : $\mathbb{P} \to \mathbb{F}$	$N_f \times N_c$
Flux boundaries	`boundF` : $\mathbb{F} \to \mathbb{F}$	$N_f \times N_f$
Flux divergence	`divF` : $\mathbb{F} \to \mathbb{P}$	$N_c \times N_f$
Stress	`S` : $\mathbb{U} \to \mathbb{S}$	$dN_f \times dN_c$
Stress boundaries	`boundS` : $\mathbb{S} \to \mathbb{S}$	$dN_f \times dN_f$
Stress divergence	`divS` : $\mathbb{S} \to \mathbb{U}$	$dN_c \times dN_f$
Pressure gradient	`gradP` : $\mathbb{P} \to \mathbb{U}$	$dN_c \times N_c$
Displacement divergence	`divU` : $\mathbb{U} \to \mathbb{P}$	$N_c \times dN_c$
Compatibility	`compat` : $\mathbb{P} \to \mathbb{P}$	$N_c \times N_c$

Discrete MPFA/MPSA Operators

Let d denote the dimensionality of the problem – i.e., $d = 2, 3$ – and let N_c and N_f represent the number of cells and faces of a nonoverlapping domain Ω. Each cell of the domain is identified as Ω_i and its enclosed surface as $\partial\Omega_i$.

We first introduce the discrete version of the variables of interest; i.e., pressure, displacement, flux, and traction:

$$\boldsymbol{p} := \{p_1, \cdots, p_{N_c}\}^T \in \mathbb{P}, \qquad \mathbb{P} = \mathbb{R}^{N_c}, \tag{13.24}$$

$$\boldsymbol{u} := \{\vec{u}_1, \cdots, \vec{u}_{N_c}\}^T \in \mathbb{U}, \qquad \mathbb{U} = \mathbb{R}^{dN_c}, \tag{13.25}$$

$$\boldsymbol{Q} := \{Q_1, \cdots, Q_{N_f}\}^T \in \mathbb{F}, \qquad \mathbb{F} = \mathbb{R}^{N_f}, \tag{13.26}$$

$$\boldsymbol{T} := \{\vec{T}_1, \cdots, \vec{T}_{N_f}\}^T \in \mathbb{S}, \qquad \mathbb{S} = \mathbb{R}^{dN_f}. \tag{13.27}$$

For vector-valued quantities, such as displacement and traction, the length of the vector depends on the dimensionality of the problem. For example, for a 2D problem using two cells, $\boldsymbol{u} = \{u_1, u_2, u_3, u_4\}^T = \{u_{1_x}, u_{1_y}, u_{2_x}, u_{2_y}\}^T$.

Following MRST's operator-based approach, in Table 13.1 we introduce the discrete MPFA and MPSA operators along with the coupling operators. The first three operators are related to the discretization of flow problems: `F`$(\cdot)$ acts on the potential and computes the fluxes (by first determining t_{ij} and then computing the gradient of the potential); `boundF`$(\cdot)$ deals with the boundary conditions; i.e., either constant pressure or constant flux. This operator will take care of the mapping from boundary values to the right discretization, keeping track of how Neumann and Dirichlet conditions should be treated differently. Finally, `divF`$(\cdot)$ computes the divergence of the flux, mapping back from faces to cell centers.

The next three operators are analogous to the first three, `S`$(\cdot)$ acting on the displacement, `boundS`$(\cdot)$ acting on the mechanic boundary conditions, and `divS`$(\cdot)$ computing the divergence of the (integrated) stress.

The last three operators are necessary for the coupled mechanics flow setting; `gradP`$(\cdot)$ computes the gradient of the pressure, `divU`$(\cdot)$ takes the divergence of the displacement, and `compat`$(\cdot)$ is a compatibility operator. This last operator (which acts on the pressure) arises naturally from the discretization process. This term has the physical interpretation of representing the volumetric expansion (or contraction) of a grid cell in response to the deviation in pressure of the cell relative to its neighbors. It is especially relevant when small timesteps are employed, providing stability to the discretized coupled system [35].

Discrete Richards' Equation

Having defined the discrete operators, we can write the discrete version of the governing equations. In an FVM framework, we typically integrate the mass conservation equation (13.5) over a cell volume,

$$\int_{\Omega_i} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V + \int_{\Omega_i} \nabla \cdot \vec{q}_w \, \mathrm{d}V = \int_{\Omega_i} \frac{\dot{m}_w}{\rho_w} \, \mathrm{d}V, \quad \forall i \in [1, N_c]. \tag{13.28}$$

Assuming that the equation is solved using an iterative strategy (see Subsection 13.3.3), after applying backward Euler, the accumulation term from (13.28) becomes

$$\frac{\partial \theta_w}{\partial t} = \frac{\theta_w^{n+1,m+1} - \theta_w^n}{\Delta t^n}, \tag{13.29}$$

where n denotes the time level and m the iteration level, and Δt is the timestep. As suggested in [12], to ensure local mass conservation, we use the modified Picard iteration to Taylor-expand $\theta_w^{n+1,m+1}$ from (13.29) as a function of ψ_w,

$$\theta_w^{n+1,m+1} = \theta_w^{n+1,m} + C_\psi^{n+1,m} \left(\psi_w^{n+1,m+1} - \psi_w^{n+1,m} \right) + H.O.T. \tag{13.30}$$

Using (13.29) and (13.30) with the higher-order terms neglected and computing the integral, the accumulation term from (13.28) is given by

$$\int_{\Omega_i} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V = \frac{V_i}{\Delta t^n} \left[\theta_{w,i}^{n+1,m} + C_{\psi,i}^{n+1,m} \left(\psi_{w,i}^{n+1,m+1} - \psi_{w,i}^{n+1,m} \right) - \theta_{w,i}^n \right], \quad \forall i \in [1, N_c],$$

where V_i is the volume of the cell i. Alternatively, we can write the previous equation in vector form as

$$\int_{\boldsymbol{\Omega}} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V = \frac{\boldsymbol{V}}{\Delta t^n} \big(\boldsymbol{\theta}_w^{n+1,m} + \boldsymbol{C}_\psi^{n+1,m} \left(\boldsymbol{\psi}_w^{n+1,m+1} - \boldsymbol{\psi}_w^{n+1,m} \right) - \boldsymbol{\theta}_w^n \big), \tag{13.31}$$

where with a slight abuse of notation, we denote

$$\int_{\boldsymbol{\Omega}} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V = \left\{ \int_{\Omega_1} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V, \ \dots \ , \int_{\Omega_{N_c}} \frac{\partial \theta_w}{\partial t} \, \mathrm{d}V \right\}^T .$$

In (13.31), $\boldsymbol{V} := \{V_1, \ \cdots, V_{N_c}\}^T$ is a vector representing the volumes of each cell of the domain. Note that the product between vectors should be interpreted as element-wise multiplications. An analogous procedure gives the expression for the source term,

$$\int_{\boldsymbol{\Omega}} \frac{\dot{m}_w}{\rho_w} \, \mathrm{d}V = \boldsymbol{V} \frac{\dot{\boldsymbol{m}}_w^n}{\rho_w}. \tag{13.32}$$

Applying the divergence theorem, the second term of (13.28) can be written as

$$\int_{\Omega_i} \nabla \cdot \vec{q}_w \, \mathrm{d}V = \int_{\partial\Omega_i} \vec{q}_w \cdot \vec{n} \, \mathrm{d}A = \sum_{j \in \mathcal{F}_i} \vec{q}_{w,j} \cdot \vec{n}_j A_j = \sum_{j \in \mathcal{F}_i} Q_j, \quad \forall i \in [1, N_c],$$

where $\mathcal{F}_i$ is the set of faces associated with the cell i. Alternatively, in vector form,

$$\int_{\boldsymbol{\Omega}} \nabla \cdot \vec{q}_w \, \mathrm{d}V = \texttt{divF}\left(\boldsymbol{Q}_w \right), \tag{13.33}$$

where we use the discrete divergence operator `divF` acting on $\boldsymbol{Q}_w$.

Combining (13.31), (13.32), and (13.33), we can write the discrete version of mass conservation as

$$\frac{\boldsymbol{V}}{\Delta t^n} \big(\boldsymbol{\theta}_w^{n+1,m} + \boldsymbol{C}_\psi^{n+1,m} \left(\boldsymbol{\psi}_w^{n+1,m+1} - \boldsymbol{\psi}_w^{n+1,m} \right) - \boldsymbol{\theta}_w^n \big) + \texttt{divF}\left(\boldsymbol{Q}_w \right) = \boldsymbol{V} \frac{\dot{\boldsymbol{m}}_w^n}{\rho_w}. \tag{13.34}$$

The discrete version of the Darcy flux through a face j is given by

$$Q_{w,j} = \frac{\rho_w g}{\mu_w} \breve{k}_{rw,j}^{n+1,m} \sum_{i \in \mathcal{I}} t_{ij} \left(\psi_{w,i}^{n+1,m+1} + \zeta_i \right), \quad \forall j \in \left[1, N_f \right],$$

where $\breve{k}_{rw,j}$ denotes the relative permeabilities evaluated at the faces; i.e., obtained by arithmetic average or upstream weighting. The previous equation written in vector form reads

$$\boldsymbol{Q}_w = \frac{\rho_w g}{\mu_w} \breve{\boldsymbol{k}}_{rw}^{n+1,m} \left(\texttt{F}(\boldsymbol{\psi}_w^{n+1,m+1} + \boldsymbol{\zeta}) + \texttt{boundF}(\boldsymbol{b}_f) \right), \tag{13.35}$$

where $\boldsymbol{b}_f \in \mathbb{F}$ is the vector of flow boundary conditions. Equations (13.34) and (13.35) represent a closed system of nonlinear algebraic equations, perfectly suited for an iterative solver. Finally, note that even though the physical model is referred to as the mixed-based version, the discretized version of the model is solved only for the pressure head $\boldsymbol{\psi}_w^{n+1,m+1}$, because we can to express $\theta_w = \theta_w(\psi_w)$ from (13.6).

Discrete Equations of Unsaturated Poroelasticity

Following the same procedure as in the Richards equation, the unsaturated storage equation (13.14) in vector form is given by

$$\begin{aligned} &\boldsymbol{V}\boldsymbol{\xi}^n \left(\boldsymbol{p}_w^{n+1,m+1} - \boldsymbol{p}_w^n\right) + \boldsymbol{V}\boldsymbol{\chi}^n \left(\boldsymbol{S}_w^{n+1,m} + \boldsymbol{C}_p^{n+1,m}\left(\boldsymbol{p}_w^{n+1,m+1} - \boldsymbol{p}_w^{n+1,m}\right) - \boldsymbol{S}_w^n\right) \\ &\quad + \alpha \boldsymbol{S}_w^n \texttt{divU}\left(\boldsymbol{u}^{n+1,m+1} - \boldsymbol{u}^n\right) + \alpha^2 \texttt{compat}\left(\boldsymbol{S}_w^n \boldsymbol{p}_w^{n+1,m+1}\right) \\ &\quad + \Delta t^n \texttt{divF}\left(\boldsymbol{Q}_w\right) = \boldsymbol{V}\Delta t^n \frac{\dot{\boldsymbol{m}}_w^n}{\rho_w}, \end{aligned} \tag{13.36}$$

where the time derivatives are approximated using backward Euler and we applied the modified Picard iteration to Taylor-expand $S_w^{n+1,m+1}$ in terms of p_w. The `compat` operator appears naturally in the MPFA/MPSA discretization of the coupled system and provides compatibility when $\Delta t^n \ll 1$. We choose to evaluate the accumulation-like terms ξ and χ at the time level n to reduce the nonlinearities; nevertheless, we acknowledge that other choices are possible.

The Darcy flux (integrated version of (13.3)) in terms of pressure reads

$$\boldsymbol{Q}_w = \frac{1}{\mu_w} \breve{\boldsymbol{k}}_{rw}^{n+1,m} \left(\texttt{F}\left(\boldsymbol{p}_w^{n+1,m+1} + \rho_w g \boldsymbol{\zeta}\right) + \texttt{boundF}\left(\boldsymbol{b}_f\right)\right). \tag{13.37}$$

The (semidiscrete) unsaturated linear momentum equation (13.11) can be integrated over each cell of the domain, giving

$$\begin{aligned} &\int_{\Omega_i} \nabla \cdot \boldsymbol{\sigma}_e \, \mathrm{d}V - \int_{\Omega_i} \alpha \nabla \left(S_w^n p_w^{n+1,m+1}\right) \mathrm{d}V \\ &\qquad + \int_{\Omega_i} \left[(1-n)\rho_s + n S_w^n \rho_w\right] \vec{g} \, \mathrm{d}V = 0, \quad \forall i \in [1, N_c]. \end{aligned} \tag{13.38}$$

Applying the divergence theorem, the first term from (13.38) can be written as

$$\int_{\Omega_i} \nabla \cdot \boldsymbol{\sigma}_e \, \mathrm{d}V = \int_{\partial\Omega_i} \boldsymbol{\sigma}_e \cdot \vec{n} \, \mathrm{d}A = \sum_{j \in \mathcal{F}_i} \boldsymbol{\sigma}_{e,j} \cdot \vec{n}_j \, A_j = \sum_{j \in \mathcal{F}_i} \vec{T}_j, \quad \forall i \in [1, N_c],$$

or in vector form as

$$\int_{\boldsymbol{\Omega}} \nabla \cdot \boldsymbol{\sigma}_e \, \mathrm{d}V = \mathtt{divS}(\boldsymbol{T}). \tag{13.39}$$

The second term of (13.38) is given by

$$\int_{\boldsymbol{\Omega}} \alpha \nabla \left(S_w^n p_w^{n+1,m+1} \right) \mathrm{d}V = \alpha \mathtt{gradP} \left(\boldsymbol{S}_w^n \boldsymbol{p}_w^{n+1,m+1} \right), \tag{13.40}$$

whereas the discretization of the body forces reads

$$\int_{\boldsymbol{\Omega}} \left[(1-n)\rho_s + n S_w^n \rho_w \right] \vec{g} \, \mathrm{d}V = \mathtt{dnc}(\boldsymbol{V}) \left((1-n)\rho_s + n \mathtt{dnc}(\boldsymbol{S}_w^n) \rho_w \right) \boldsymbol{g}. \tag{13.41}$$

Here, we have used the $\mathtt{dnc}(\cdot)$ operator, which converts a vector of length N_c to a vector of length dN_c by repeating each element of the N_c vector d times. For example, for a 2D problem with two cells, $\mathtt{dnc}(\boldsymbol{V}) = \mathtt{dnc}(\{V_1, V_2\}^T) = \{V_1, V_2, V_1, V_2\}^T$.

Combining (13.39), (13.40), and (13.41) gives the discrete version of the momentum equation in vector form,

$$\begin{aligned} &\mathtt{divS}\,(\boldsymbol{T}) - \alpha \mathtt{gradP} \left(\boldsymbol{S}_w^n \boldsymbol{p}_w^{n+1,m+1} \right) \\ &\quad + \mathtt{dnc}\,(\boldsymbol{V}) \left((1-n)\rho_s + n \mathtt{dnc}(\boldsymbol{S}_w^n) \rho_w \right) \boldsymbol{g} = \boldsymbol{0}. \end{aligned} \tag{13.42}$$

Finally, for a generic face j, the traction forces acting on that face are given by

$$\vec{T}_j = \sum_{i \in \mathcal{I}} s_{ij} \vec{u}_i^{n+1,m+1}, \quad \forall j \in \left[1, N_f\right],$$

or in vector form,

$$\boldsymbol{T} = \mathtt{S}(\boldsymbol{u}^{n+1,m+1}) + \mathtt{boundS}\,(\boldsymbol{b_m}), \tag{13.43}$$

where $\boldsymbol{b_m} \in \mathbb{S}$ is the vector of boundary conditions for the mechanical problem. Equations (13.36), (13.37), (13.42), and (13.43) represent the complete set of discrete equations. This set of equations can be solved using a sequential approach [6, 22] or a monolithic approach [34]. The latter is the preferred method for this module, with the vector $\{\boldsymbol{u}_w^{n+1,m+1}, \boldsymbol{p}_w^{n+1,m+1}\}^T$ as the only compound primary variable.

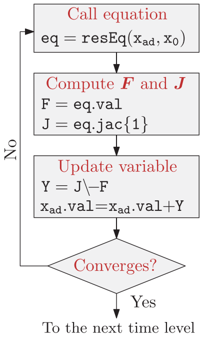


Figure 13.3 Workflow of the iterative solver applied to a generic equation.

13.3.3 Solving the Equations

To solve the system of equations we implement the modified Picard iteration method as a part of an iterative solver as presented in the MRST textbook [28]. Other types of linearization schemes have been successfully applied to Richards' equation and to a lesser extent to unsaturated poroelasticity. Usual schemes include the classical Newton method, the Picard method, the Picard–Newton method, and the L-scheme with and without Anderson acceleration (see [8, 19, 29]).

The resulting iterative scheme can be written as

$$\frac{\mathrm{d}\boldsymbol{F}}{\mathrm{d}\boldsymbol{x}}(\boldsymbol{x}^m)\,\delta\boldsymbol{x}^{m+1} = -\boldsymbol{F}(\boldsymbol{x})^m, \qquad \boldsymbol{x}^{m+1} \leftarrow \boldsymbol{x}^m + \delta\boldsymbol{x}^{m+1}, \tag{13.44}$$

where $\boldsymbol{F}$ is the residual vector, $\boldsymbol{J} := \mathrm{d}\boldsymbol{F}/\mathrm{d}\boldsymbol{x}$ is the Jacobian matrix depending on the current solution $\boldsymbol{x}^m$, and $\delta\boldsymbol{x}^{m+1}$ is the updated solution. Generally, the manual computation of $\boldsymbol{J}$ is a tedious and error-prone process. To avoid such a process, we exploit the automatic differentiation (AD) interface available in MRST, which in essence consists of breaking down the computation into nested elementary differentiation operations (see [24, 27, 28]). Figure 13.3 shows a schematic representation of the workflow of the iterative solver.

The selection of the timestep Δt plays a key role in a solver's performance. As a general rule, the smaller the timestep the greater the chances of convergence. However, decreasing the timestep too much could be unfeasible for some simulations due to the increase in computational time. A better strategy is to use an adaptive timestepping algorithm, such as the one implemented in Hydrus-1D [47].

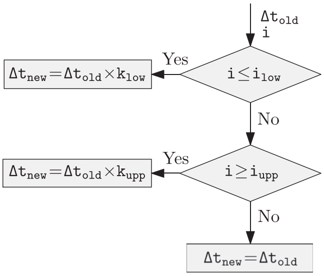


Figure 13.4 Workflow of the adaptive time stepping algorithm.

The algorithm determines the next timestep size based on the number of iterations needed to achieve convergence in the last time level (see Figure 13.4). The idea is to increase Δt in case the number of iterations `i` is less (or equal) than a lower optimal iteration range $\mathtt{i}_{\mathtt{low}}$ (i.e., 3), decrease Δt if `i` is greater (or equal) than an upper optimal iteration range $\mathtt{i}_{\mathtt{upp}}$ (i.e., 7), or keep the same value otherwise. To increase Δt, we multiply $\Delta t_{\mathtt{old}}$ by a lower multiplication factor $\mathtt{k}_{\mathtt{low}}$ (i.e., 1.3), and to decrease it, we multiply $\Delta t_{\mathtt{old}}$ by an upper multiplication factor $\mathtt{k}_{\mathtt{upp}}$ (i.e., 0.7).

13.4 Numerical Examples

In this section we present four numerical examples; the first two are numerical convergence tests and the last two are practical applications. The convergence tests include Richards' equation (`convAnalysisRE.m`) and the equations of unsaturated poroelasticity (`convAnalysisUnsatBiot.m`). The third example is a well-known problem for unsaturated flow, where we simulate the water infiltration in a nondeformable initially dry soil (see `waterInfiltrationRE.m`). The last example, `desiccationUnsatBiot.m`, consists of a desiccation process of a clayey soil under atmospheric evaporation in a Petri dish.

Even though the codes for the convergence tests are included in the module, in principle they are not meant as tutorials. To start using `fv-unsat`, we recommend `waterInfiltrationRE.m`, which offers a step-by-step explanation of the module.

13.4.1 Numerical Convergence Tests

The first two examples involve numerical convergence tests, one for Richards' equation and one for the unsaturated poroelastic equations. Before that, we define the errors used to determine the converge rates.

We are interested in measuring the errors for the pressure (or pressure head), displacement, flux, and traction forces. We use the subscript h to denote the numerical approximation and no subscript for the exact solution. We define the following relative discrete L_2-type errors as in [33]:

$$\varepsilon_p^{h,\Delta t} = \frac{\left(\sum_i^{N_c} V_i \,|p_i - p_{h,i}|^2\right)^{1/2}}{\left(\sum_i^{N_c} V_i \,|p_i|^2\right)^{1/2}}, \qquad \varepsilon_Q^{h,\Delta t} = \frac{\left(\sum_j^{N_f} A_j \,|Q_j - Q_{h,j}|^2\right)^{1/2}}{\left(\sum_j^{N_f} A_j \,|Q_j|^2\right)^{1/2}},$$

$$\varepsilon_{\vec{u}}^{h,\Delta t} = \frac{\left(\sum_i^{N_c} V_i \,|\vec{u}_i - \vec{u}_{h,i}|^2\right)^{1/2}}{\left(\sum_i^{N_c} V_i \,|\vec{u}_i|^2\right)^{1/2}}, \qquad \varepsilon_{\vec{T}}^{h,\Delta t} = \frac{\left(\sum_j^{N_f} A_j \,|\vec{T}_j - \vec{T}_{h,j}|^2\right)^{1/2}}{\left(\sum_j^{N_f} A_j \,|\vec{T}_j|^2\right)^{1/2}},$$

where V_i and A_j are the cell volumes and face areas, respectively. For a given variable, we define the reduction between two successive levels of refinement as the ratio between the errors obtained by halving the spatial resolution for a fixed time step. For example, for the pressure, we have the reduction and the convergence rate given by

$$\text{Red}_p = \varepsilon_p^{h,\Delta t}/\varepsilon_p^{h/2,\Delta t}, \qquad \text{Rate}_p = \log_2(\text{Red}_p).$$

Richards' Equation

In this example, we present a numerical convergence analysis of the two-dimensional incompressible mixed-based formulation of Richards' equation. This analysis is performed in a unit square with a final simulation time of 1 and a timestep $\Delta t = 0.1$. The computational mesh is a structured Cartesian grid. The relative permeabilities on the faces are approximated using an arithmetic mean of the cell centers, and for simplicity, gravity effects are neglected. Moreover, all of the physical parameters are assumed to be equal to one, except $\alpha_v = 0.4$, $\theta_w^s = 0.4$, $\theta_w^r = 0.1$, $n_v = 2$, and $m_v = 0.5$. We assume the existence of a time-dependent solution

$$\psi_w(x,y,t) = -t(1-x)x\sin(\pi x)(1-y)y\cos(\pi y) - 1,$$

satisfying $\psi_w(0,y,t) = \psi_w(1,y,t) = \psi_w(x,0,t) = \psi_w(x,1,t) = \psi_w(x,y,0) = -1$. With this assumption, it is possible to obtain an exact expression for the source term $\dot{m}_w/\rho_w = f_w$ and compute the errors. We refer to [39] for more details.

Table 13.2 shows the results for five different levels of spatial refinement. Pressure head and fluxes show quadratic convergence rates. These results are consistent with reported rates for MPFA schemes on structured-uniform grids (see, e.g., [2, 3]).

Unsaturated Poroelasticity

In this analysis, we investigate the numerical convergence rates for the unsaturated poroelastic equations. The domain, final simulation time, timestep, average of k_{rw},

Table 13.2 *Convergence test for Richards' equation.*

h	$\varepsilon_{\psi}^{h,\Delta t}$	Red_{ψ}	Rate_{ψ}	$\varepsilon_{Q}^{h,\Delta t}$	Red_{Q}	Rate_{Q}
0.1	5.338×10^{-4}	3.9949	1.9982	2.965×10^{-2}	3.9823	1.9936
0.05	1.336×10^{-4}	3.9970	1.9989	7.445×10^{-3}	4.0207	2.0075
0.025	3.343×10^{-5}	3.9991	1.9997	1.852×10^{-3}	4.0182	2.0065
0.0125	8.386×10^{-6}	3.9998	1.9999	4.608×10^{-3}	4.0110	2.0040
0.00625	2.090×10^{-6}			1.149×10^{-4}		

Table 13.3 *Convergence test for unsaturated poroelasticity: pressure and displacement.*

h	$\varepsilon_{p}^{h,\Delta t}$	Red_{p}	Rate_{p}	$\varepsilon_{\vec{u}}^{h,\Delta t}$	$\mathrm{Red}_{\vec{u}}$	$\mathrm{Rate}_{\vec{u}}$
0.2	8.817×10^{-4}	3.9106	1.9674	9.156×10^{-2}	4.0877	2.0313
0.1	2.255×10^{-4}	3.9381	1.9775	2.240×10^{-2}	4.0017	2.0006
0.05	5.725×10^{-5}	3.9836	1.9941	5.597×10^{-3}	3.9859	1.9949
0.025	1.437×10^{-5}	3.9970	1.9989	1.404×10^{-3}	3.9852	1.9946
0.0125	3.596×10^{-6}	3.9992	1.9997	3.524×10^{-4}	3.9753	1.9911
0.00625	8.991×10^{-7}			8.864×10^{-5}		

and water retention parameters are the same as in the last example. However, we now include gravity contributions. The physical parameters different from unity are $C_s = 0.1$, $n = 0.4$, and $\alpha = 0.9$.

We are interested in convergence rates of pressures and displacements, as well as fluxes and traction forces. We assume the following time-dependent solutions for the primary variables:

$$p_w(x,y,t) = -tx(1-x)y(1-y)\sin(\pi x)\cos(\pi y) - 1,$$
$$\vec{u}(x,y,t) = tx(1-x)y(1-y)\big[\sin(\pi x),\, \cos(\pi y)\big]^T.$$

We employ Dirichlet boundary conditions for the pressure and displacement satisfying the above equations. The initial conditions are obtained by setting $t = 0$, the mesh is a structured triangular grid, and the analysis is performed for six different levels of spatial refinement. The results are shown in Tables 13.3 and 13.4.

Pressures, displacements, and fluxes show quadratic convergence rate. The convergence rate for traction is less uniform. Nevertheless, it is greater than 1.5 and lower than 2, which is in agreement with previously reported rates on structured grids for elasticity and (saturated) poroelasticity [21, 35].

Table 13.4 *Convergence test for unsaturated poroelasticity: flux and traction.*

h	$\varepsilon_Q^{h,\Delta t}$	Red$_Q$	Rate$_Q$	$\varepsilon_{\vec{T}}^{h,\Delta t}$	Red$_{\vec{T}}$	Rate$_{\vec{T}}$
0.2	1.025×10^{-2}	3.6349	1.8619	7.048×10^{-2}	3.4914	1.8038
0.1	2.821×10^{-3}	3.8863	1.9584	2.019×10^{-2}	3.2813	1.7143
0.05	7.258×10^{-4}	3.9755	1.9911	6.152×10^{-3}	3.3032	1.7239
0.025	1.826×10^{-4}	3.9968	1.9989	1.862×10^{-3}	2.9773	1.5740
0.0125	4.568×10^{-5}	4.0007	2.0020	6.256×10^{-4}	3.0802	1.6230
0.00625	1.142×10^{-5}			2.031×10^{-4}		

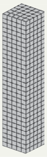
13.4.2 Water Infiltration in a Column of Dry Soil

In this example, we solve a water infiltration problem in an initially dry soil column. The water flows from top to bottom and is modeled using Richards' equation. The simplicity of the problem represents an excellent opportunity to introduce the module (see `waterInfiltrationRE.m` from the `examples` folder).

We start by constructing a Cartesian grid consisting of five cells in the x- and y-directions and 30 cells in the z-direction. The domain is $100 \times 100 \times 100$ cm^3. We refer to chapter 3 of the MRST textbook [28] for more details regarding mesh generation in MRST.

```
nx = 5;     ny = 5;     nz = 30;           % cells
Lx = 1;     Ly = 1;     Lz = 1;            % domain lenght [m]
G = cartGrid([nx, ny, nz], [Lx, Ly, Lz]); % create Cartesian grid
G = computeGeometry(G);                    % compute geometry

% Plotting grid
newplot; plotGrid(G); axis off;
pbaspect([1, 1, 5]); view([-51, 26]);
```



Next, we declare the hydraulic parameters of the soil. We use the physical parameters of a field sample from New Mexico [37]. Most of the properties can be accessed from our mini-catalog of soils (see `getHydraulicProperties.m`). The properties are stored in SI units inside the `phys` structure, which, in turn, contains the `flow` substructure. For coupled problems, the `phys` structure will also contain the `mech` substructure (see next example):

```
soil = getHydraulicProperties('newMexSample');  % get soil properties
phys = struct(); % create structure to store physical properties

% Flow parameters
phys.flow.rho       = 1 * gram / (centi * meter)^3;           % density
phys.flow.mu        = 0.01 * gram / (centi * meter * second); % viscosity
phys.flow.g         = 980.66 * centi * meter / (second^2);     % gravity
phys.flow.gamma     = phys.flow.rho * phys.flow.g;  % specific gravity
phys.flow.K         = soil.K_s;  % saturated hydraulic conductivity
phys.flow.perm      = (phys.flow.K * phys.flow.mu / phys.flow.gamma) .* ...
                        ones(G.cells.num, 1);  % intrinsic permeability
phys.flow.alpha     = soil.alpha / meter;  % vGM parameter
phys.flow.n         = soil.n;              % vGM parameter
phys.flow.m         = 1-(1/phys.flow.n);   % vGM parameter
phys.flow.theta_s   = soil.theta_s;        % Water content at saturation conditions
phys.flow.theta_r   = soil.theta_r;        % Residual water content
```

Boundary and initial conditions are declared next. For this problem, $\psi_w = -75$ cm is set at the top and $\psi_w, = -1\,000$ cm at the bottom, and the rest are set as no flux by default. Initially, we set $\psi_w = -1\,000$ cm for all cells. Boundary conditions are declared following the MRST convention (see chapter 5 of the MRST textbook [28]). In addition, we need to create `bcVal` (a vector containing the values of the boundary conditions) for backward compatibility with the `fvbiot` module. It is important to mention that if gravity effects are considered, we must include their contributions to the Dirichlet faces in the `bcVal` vector:

```
% Extracting grid information
:

% Creating the boundary structure
psiT         = -75 * centi * meter;    % Top boundary pressure head
psiB         = -1000 * centi * meter;  % Bottom boundary pressure head
bc           = addBC([], z_min, 'pressure', psiT);
bc           = addBC(bc, z_max, 'pressure', psiB);
bcVal        = zeros(G.faces.num, 1);
bcVal(z_min) = psiT + zetaf(z_min);    % assigning Top boundary
bcVal(z_max) = psiB + zetaf(z_max);    % assigning Bottom boundary
```

The problem is discretized using the `mpfa` routine from the `fvbiot` module. The `mpfa` function takes as input arguments the `G` structure, the `flow` substructure, and the boundary conditions structure `bc`. The output contains the discrete operators that later will be used to construct the model:

```
%% Discretize the flow problem using MPFA
mpfa_discr = mpfa(G, phys.flow, [], 'bc', bc, 'invertBlocks', 'matlab');
```

After declaring parameters structures for time/printing (`time_param`, `print_param`) we are in position to construct the model. This is done by calling the function `modelRE` (from the `models` folder) as follows:

```
%% Call Richards' equation model
modelEqs = modelRE(G, phys, mpfa_discr, bc, bcVal, 'arithmetic', 'on');
```

Listing 13.1 *The principal parts of the workflow of* `modelRE`.

```
function model = modelRE(G, phys, mpfa_discr, bc, bcVal, relPermMethod, gEffects)
:

% Soil Water Retention Curves (SWRC) for the theta-psi model
[theta, krw, C_theta] = vGM_theta(phys);

% Discrete mpfa operators
F      = @(x) mpfa_discr.F * x;          % Flux
boundF = @(x) mpfa_discr.boundFlux * x;  % Boundary fluxes
divF   = @(x) mpfa_discr.div * x;        % Divergence

% Relative permeability at the faces
if strcmp(relPermMethod, 'arithmetic')
    krw_faces = @(psi_m) arithmeticAverageMPFA(G, krw, bc, psi_m);
elseif strcmp(relPermMethod, 'upstream')
    krw_faces = @(psi_m) upstreamWeightingMPFA(G, krw, bc, bcVal, ...
        mpfa_discr, phys, psi_m, 'psi', gEffects);
else
    error('Method not implemented. Use either ''arithmetic'' or ''upstream''')
end

% Darcy Flux
Q = @(psi, psi_m) (phys.flow.gamma ./ phys.flow.mu) .* krw_faces(psi_m) .* ...
    (F(psi + gravOn * zetac) + boundF(bcVal));

% Mass Conservation Equation
psiEq = @(psi, psi_n, psi_m, dt, source)  (V ./ dt) .* (theta(psi_m) ...
    + C_theta(psi_m) .* (psi - psi_m) - theta(psi_n)) ...
    + divF(Q(psi, psi_m)) - V .* source;
:
```

The function `modelRE` takes as input arguments the grid structure `G`, the physical properties structure `phys`, the discretized structure `mpfa_discr`, the boundary conditions structure and vector values `bc` and `bcVal`, and two string arguments. The first string argument specifies the way relative permeabilities at the faces should be calculated (e.g., `'arithmetic'` or `'upstream'`), and the last argument is either `'on'` or `'off'` depending whether gravity effects are included or neglected.

For completeness, we show the principal parts of the workflow of `modelRE` in Listing 13.1. First, we retrieve the SWRC quantities (see (13.6)–(13.8)) using the utility function `vGM_theta`. Because the problem is already discretized, we can create the discrete MPFA operators as introduced in Table 13.1. Next, we compute the relative permeabilities at the faces using the preferred method. Finally, we declare the discrete equations as anonymous functions; i.e., `Q` for the Darcy flux and `psiEq` for the mass conservation equation. The function `modelRE` returns the `model` structure containing the discrete equations together with the

SWRC-related quantities. We remark the straightforward equivalence between the mathematical and computational equations.

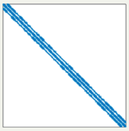
Now, we can solve the nonlinear set of equations using a nested while loop. The first corresponds to the time loop and the second to the solver `solverRE` (shown in a separate code excerpt). Once we exit the solver loop (provided successful convergence), the timestep `dt` for the next time level is calculated using the adaptive timestepping routine `timeStepping`:

```
while time_param.time < time_param.simTime
    psi_n = psi; % current time step (n-index)
    time_param.time = time_param.time + time_param.dt; % current time
    source = zeros(G.cells.num,1); % source term equal to zero
    % Newton loop
    [psi,psi_m,iter] = solverRE(psi_n,modelEqs,time_param,solver_param,source);
    % Determine next time step
    [time_param.dt,print_param.print]=timeStepping(time_param,print_param,iter);
    :
end
```

The solver `solverRE` is written in such a way that it exploits the capabilities of the AD framework:

```
function [psi, psi_m, iter] = solverRE(psi_n, modelEqs, time_param, ...
    solver_param, source)
:
psi_ad = initVariablesADI(psi_n);  % Initialiazing AD-variable

% Newton loop
while (res > solver_param.tol) && (iter <= solver_param.maxIter)
    psi_m = psi_ad.val; % current iteration level (m-index)
    eq = modelEqs.psiEq(psi_ad, psi_n, psi_m, time_param.dt, ...
        source);     % call equation from model
    R = eq.val;      % residual
    J = eq.jac{1};   % Jacobian
    Y = J\-R;        % solve linear system
    psi_ad.val  = psi_ad.val + Y; % update
    res = norm(R);  % compute tolerance
    :
end
psi = psi_ad.val; % return updated pressure head
```



In case the solver does not converge in the prescribed maximum number of iterations, an error is printed in the console. The options to enforce convergence are either to increase `maxIter` or decrease `tol`. The results can be easily accessed via the `sol` object for all printing times. In Figure 13.5, we show the pressure head and water content distributions corresponding to 21.6 hours. Alternatively, the

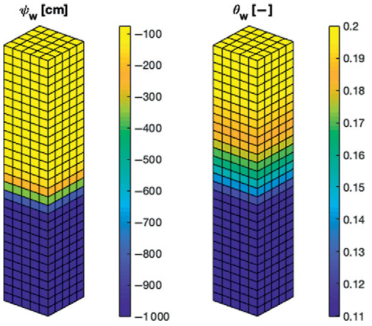


Figure 13.5 Solutions to the water infiltration problem in an initially dry soil. We show the pressure head (left) and water content (right) after 21.6 hours of water infiltration from top to bottom. Note that approximately half of the domain remains in dry conditions.

interested user can explore further plotting functionalities via the `plotToolBar` interface from the `mrst-gui` module.

13.4.3 Desiccation of a Clayey Soil in a Petri Dish

In this numerical experiment, we study the desiccation process of a clayey sample in a Petri dish using real parameters (see `desiccationUnsatBiot.m`). The desiccation is driven by an evaporation process that is modeled using atmospheric boundary conditions, allowing us to resemble with more precision a realistic evaporation scenario. Our main motivation to study soil desiccation is the formation of cracks. Even if fractures are not included in this model, it is useful to predict whether the conditions before cracking exist. The desiccation process involves a gradual reduction of saturation with a simultaneous reduction in the pressure and soil shrinkage [18].

The domain consists of a standard Petri dish (10 cm in diameter and 1.5 cm thick) containing a sample of clay. In such a setup, the soil is constrained everywhere but the top, where the evaporation takes place at stress-free conditions (see Figure 13.6). The evaporation at the top of the Petri dish can be either flux controlled or pressure controlled. In an atmospheric evaporation scenario, the soil initially dries at a maximum evaporation rate (thus a flux-controlled top boundary condition is imposed) and then smoothly decreases, approaching zero in the limit

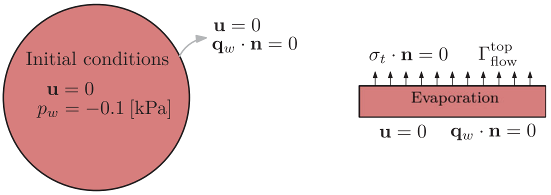

Initial conditions
$\mathbf{u} = 0$
$p_w = -0.1$ [kPa]
$\mathbf{u} = 0$
$\mathbf{q}_w \cdot \mathbf{n} = 0$
$\sigma_t \cdot \mathbf{n} = 0$
$\Gamma^{\text{top}}_{\text{flow}}$
Evaporation
$\mathbf{u} = 0$
$\mathbf{q}_w \cdot \mathbf{n} = 0$

Figure 13.6 The Petri dish domain showing the boundary and initial conditions.

when $S_w \to S_w^r$ (in this second stage a pressure-controlled boundary condition is used). The criteria can be written as

$$\Gamma^{\text{top}}_{\text{flow}} = \begin{cases} E_{\text{max}}, & p_w^{\text{top}} \geq p_w^{\text{crit}}, \\ p_w^{\text{crit}}, & \text{otherwise}, \end{cases}$$

where $\Gamma^{\text{top}}_{\text{flow}}$ is the flow boundary condition at the top of the domain (note that the word “flow” does not refer to a “flux” boundary condition but rather the subproblem as in the flow/mechanics coupled problem), E_{max} is the maximum evaporation rate, and p_w^{crit} is the water critical pressure [15]. There are several correlations to estimate E_{max} for field-scale applications [25]. In this case, we adopt an experimental value obtained by Stirling [42] and more recently employed in numerical experiments in [10]. The critical pressure p_w^{crit} is the minimum allowed pressure at the soil surface. This value is a function of the ambient psychrometric conditions and can be estimated as

$$p_w^{\text{crit}} = \frac{\log(\phi) R T \rho_w}{M},$$

where ϕ is the relative humidity, R is the universal gas constant, T is the absolute temperature, ρ_w is the water density, and M is the molecular weight of water [47].

The soil is initially at virtually saturated conditions – i.e., $S_w = 0.9996$ – and the final simulation time is 2 hours. Now, we describe each step of the simulation process. We highly encourage the interested reader to use `desiccationUnsatBiot.m` along with this explanation.

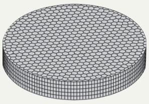
As usual, we start by generating the computational grid. First, we create a Delaunay triangulation on a circle using the (freely available) mesh generator `distemsh` [36]. To add `distmesh` to MRST, we follow the procedure described in [28]:

```
pth = fullfile(ROOTDIR,'utils','3rdparty','distmesh'); mkdir(pth)
unzip('http://persson.berkeley.edu/distmesh/distmesh.zip', pth);
mrstPath('reregister','distmesh', pth);
```

Listing 13.2 *Grid construction for the Petri dish.*

```
% Two-dimensional grid
r      = 50 * milli * meter;          % radii of the Petri-dish
fd     = @(p) sqrt(sum(p.^2, 2)) - r; % circular domain function
min_x  = -r;  max_x = r;              % min and max values in x-axis
min_y  = -r;  max_y = r;              % min and max values in y-axis
h      = (2*r)/25;                    % step size
[p, t] = distmesh2d(fd, @huniform, h, [min_x, min_y; max_x, max_y], []);
p      = p + r;                       % shifting triangulation points
G      = triangleGrid(p, t);          % creating triangular grid
G      = pebi(G);                     % creaing Voronoi diagram

% Extrude in the z-direction
Lz = 15 * milli * meter;  % thickness of the Petri-dish
nz = 5;                   % number of layers in z-axis
dz = Lz/nz;               % thickness of each layer
thick = dz .* ones(nz, 1);     % thickness vector
G = makeLayeredGrid(G, thick); % extrude grid
G = computeGeometry(G);        % compute geometry
```



We employ the function `distmesh2d` to triangulate a circle of radius `r`, with step size `h`. With the triangulation points `p` and the connectivity map `t` available, we can generate the triangular grid using `triangleGrid` and then apply a Voronoi diagram using the `pebi` routine to obtain the hexagonal grid. Finally, to generate the three-dimensional grid, we extrude the hexagonal grid in the z-direction using the function `makeLayeredGrid`[2] (see Listing 13.2).

After extracting useful topological data, we declare the physical parameters for the mechanics and the flow problem using the `phys` structure:

```
% Mechanics parameters [Kaolinite]
phys.mech.lambda = 1.229E11 .* ones(Nc, 1) * Pascal;  % first Lame parameter
phys.mech.mu     = 4.7794E10 .* ones(Nc, 1) * Pascal; % second Lame parameter
phys.mech.C_s    = 5.618E-11 / Pascal;                % solid compressibility
phys.mech.rho    = 1769 * kilo * gram / meter^3;      % solid density
phys.mech.stiff  = shear_normal_stress(Nc, Nd, ...    % stiffnes matrix
                    phys.mech.mu, phys.mech.lambda, 0 .* phys.mech.mu);
```

Here, we assume homogeneity in the physical properties. However, the code is flexible to include heterogeneous permeability and elasticity coefficients. The elastic parameters were taken from [32] for a sample of kaolinite and the hydraulic properties from [11] for clay. The mechanic discretization requires the construction of the stiffness matrix. This is done using the function `shear_normal_stress` from the `fvbiot` module.

[2] Technically speaking, these grids are referred to as 2.5–dimensional grids.

We use the soil catalog to get the hydraulic properties of the clay. The critical pressure is determined using `computeCriticalPressure`. For this example, we assume standard laboratory psychometric conditions; i.e., $T = 298.15\,\mathrm{K}$ and $\phi = 0.5$:

```
% Flow parameters [Water]
soil = getHydraulicProperties('clay');
:
phys.flow.temperature = 298.15 * Kelvin;  % Ambient temperature
phys.flow.relativeHumidity = 0.5;         % Ambient relative humidity
p_crit = computeCriticalPressure(phys);
```

Now, we proceed to declare the boundary conditions. For the mechanics, we set $\vec{u} = 0$ at the sides and bottom of the domain, whereas the top is assumed to be stress-free by default (note that the keyword `'pressure'` indicates a displacement condition and `'flux'` indicates a traction condition):

```
% Creating the boundary structure for the mechanics problem
bcMech = addBC([], sides, 'pressure', 0);       % u=0 at the sides
bcMech = addBC(bcMech, z_max, 'pressure', 0);   % u=0 at the bottom
bcMechVals = zeros(Nd * Nf, 1);
```

For the flow boundary conditions, we have two scenarios: flux and pressure controlled. For the flux-controlled scenario we have only flux conditions:

```
% Creating the boundary structure for flux-controlled BC
bcFlow_f     = addBC([], z_min, 'flux', Qtop_f);
bcFlowVals_f = zeros(Nf, 1);
bcFlowVals_f(z_min) = Qtop_f;
```

whereas for the pressure-controlled, we have zero flux except at the top:

```
% Creating the boundary structure for pressure-controlled BC
bcFlow_p     = addBC([], z_min, 'pressure', p_crit);
bcFlowVals_p = zeros(Nf, 1);
bcFlowVals_p(z_min) = p_crit + phys.flow.gamma .* zetaf(z_min);
```

For the initial conditions, we assume an initially undeformed sample – that is, $\vec{u}(x,y,z,0)$=0 m – and a homogeneous pressure field of $p_w(x,y,z,0) = -0.1$ kPa:

```
u_init = zeros(Nd * Nc, 1) * meter;
p_init = -0.1 * kilo * Pascal * ones(Nc, 1);
```

Once the boundary and initial conditions have been declared, we can discretize the different problems. On one hand, we have the mechanical problem, which is discretized using the `mpsa` routine from `fvbiot`, and, on the other hand, we have the flow problem which is discretized using `mpfa`. Note that the flow problem is divided into the flux- and pressure-controlled subproblems, because different boundary conditions result in different discrete operators:

```
% Discretize mechanics problem
mpsa_discr = mpsa(G,phys.mech.stiff,[],'invertBlocks','matlab','bc',bcMech);

% Discretize flow problem for flux-controlled boundary conditions
mpfa_discr_flux = mpfa(G,phys.flow, [],'invertBlocks','matlab','bc',bcFlow_f);

% Discretize flow problem for pressure-controlled boundary conditions
mpfa_discr_pres = mpfa(G,phys.flow, [],'invertBlocks','matlab','bc',bcFlow_p);
```

After declaring the time and printing parameters, we set up the two different scenarios (flux and pressure controlled) using the function `modelUnsatBiot`:

```
%% Calling the model for the unsaturated poroelastic equations

% Setting up model for flux-controlled problem
modelEqsFlux = modelUnsatBiot(G, phys, mpfa_discr_flux, mpsa_discr, ...
    bcFlow_f, bcFlowVals_f, bcMech, bcMechVals, 'upstream', 'on');

% Setting up model for pressure-controlled problem
modelEqsPres = modelUnsatBiot(G, phys, mpfa_discr_pres, mpsa_discr, ...
    bcFlow_p, bcFlowVals_p, bcMech, bcMechVals, 'upstream', 'on');
```

Note that `modelUnsatBiot` now uses both the mechanics and flow boundary conditions as well as discretization structures. The last two string arguments are the same as in `modelRE`. To avoid being repetitive, and because `modelUnsatBiot` is essentially the same as `modelRE` (structure-wise, not complexity-wise), we prefer not to show this function and proceed with solving the coupled systems.

To solve the coupled problem we create two time loops, one for each flow scenario. The flux-controlled time loop is shown in Listing 13.3. The process is essentially the same as in `waterInfiltrationRE.m`, except for some technicalities. Note that after calling `solverUnsatBiot` we calculate the value of the top pressure of the domain using the function `computeTopPressure`. This function uses a TPFA discretization to approximate the mean value of the surface pressure. Next, we check whether the critical pressure is reached or not. If the pressure is higher, we proceed to determine the next timestep using `timeStepping`.

Listing 13.3 *Flux-controlled time loop.*

```
while (time_param.time < time_param.simTime) && (p_top > p_crit) ...
        && (pControlled == false)

    p_n = p; % current time level (n-index)
    u_n = u; % current time level (n-index)
    time_param.time = time_param.time + time_param.dt; % cumulative time

    % Source terms
    sourceFlow = zeros(Nc, 1); % no sources for the flow
    sourceMech = modelEqsFlux.body(p_n); % sourceMech = body force

    % Calling Newton solver
    [p, p_m, u, iter] = solverUnsatBiot(G, p_n, u_n, modelEqsFlux, ...
        time_param, solver_param, sourceFlow, sourceMech);

    % Approximating top pressure
    fluxTemp = modelEqsFlux.Q(p, p_m);
    p_top = computeTopPressure(G, phys, p, fluxTemp, modelEqsFlux);

    % If it is flux controlled, update time step and store solution
    if (p_top > p_crit)
        % Calling time stepping routine
        [time_param.dt, print_param.print] = timeStepping(time_param, ...
            print_param, iter);

        : % store solution if necessary
    else
        : % change to pressure controlled loop
    end
end
```

If the pressure is less than (or equal to) the critical pressure, we switch to the pressure-controlled time loop.

Because the pressure-controlled loop is essentially the same, we show the solver and the sparsity of the system in Listing 13.4. The Jacobian matrix consists of four blocks, which are characteristic of the monolithic approach:

- Upper-left: displacement contribution to the momentum equation, `eq1`.
- Upper-right: pressure contribution to the momentum equation, `eq2`.
- Lower-left: displacement contribution to the storage equation, `eq3`.
- Lower-right: pressure contribution to the storage equation, `eq4`.

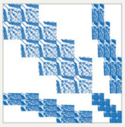
The simulation results are shown in Figures 13.7–13.10. In Figure 13.7, we show the saturation profile for the final simulation time. As expected, the lower saturation zones are located at the top layer due to the evaporation process, whereas the bottom layer remains at nearly saturated conditions. In Figure 13.8, we show the variation

Listing 13.4 *Solver for the unsaturated Biot equations.*

```
function [p, p_m, u, iter] = solverUnsatBiot(G, p_n, u_n, modelEqs, ...
    time_param, solver_param, sourceFlow, sourceMech)
:
% Initializing AD-variables
p_ad = initVariablesADI(p_n);
u_ad = initVariablesADI(u_n);

% Newton loop
while (res > solver_param.tol) && (iter <= solver_param.maxIter)
    % Calling equations
    p_m = p_ad.val; % current iteration level (m-index)
    eq1 = modelEqs.uEq1(u_ad);
    eq2 = modelEqs.uEq2(p_ad, p_n, sourceMech);
    eq3 = modelEqs.pEq1(p_n, u_ad, u_n);
    eq4 = modelEqs.pEq2(p_ad, p_n, p_m, time_param.dt, sourceFlow);

    J = [eq1.jac{1} eq2.jac{1}; eq3.jac{1}, eq4.jac{1}];
    R = [eq1.val + eq2.val; eq3.val + eq4.val];
    Y = J\-R;  % solve linear system
    u_ad.val = u_ad.val + Y(1:Nd*Nc);     % update u
    p_ad.val = p_ad.val + Y(Nd*Nc+1:end); % update p
    res = norm(R);  % compute tolerance
    :
end
p = p_ad.val; % updating pressure value
u = u_ad.val; % updating displacement value
```



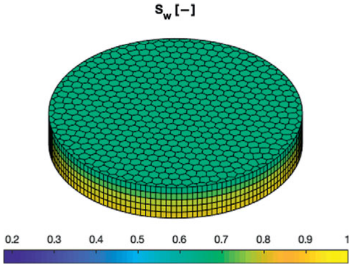


Figure 13.7 Saturation field for the final simulation time.

of the top pressure head and flux with respect to time. The change in boundary condition modes that takes place at 0.44 hours highly influences the evaporation process. After this point, the pressure declines abruptly toward the critical value, whereas the flux smoothly approaches zero as the driven force for the evaporation vanishes.

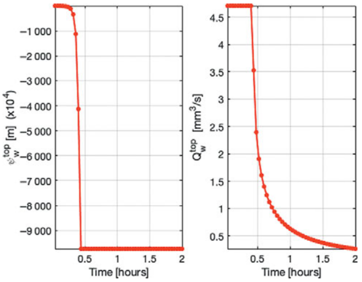


Figure 13.8 Top pressure head (left) and surface flux (right) evolution.

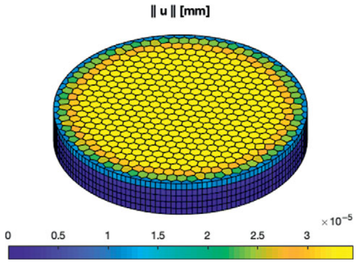


Figure 13.9 Magnitude of the displacement for the final simulation time. The deformation is maximum where the evaporation takes place.

In Figure 13.9, we show the magnitude of the displacement field for the final simulation time. Note that the displacement is maximum at the top layer, which again is in agreement with the expected results. Finally, in Figure 13.10 we show a closeup of the positive quarter domain of the top layer, where the arrows depict the direction of the displacement field, demonstrating the tensile nature of the stresses that eventually cause the rupture of the material.

13.5 Concluding Remarks

In this chapter, we presented a flexible solver based on robust multipoint finite-volume schemes (MPFA/MPSA) for simulating flow in unsaturated soils. We studied the case where deformations effects are neglected (Richards' equation) and the case where small deformations and linear elastic behavior of the soil are assumed

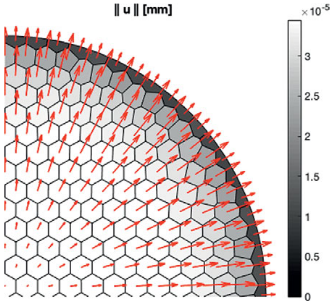


Figure 13.10 Positive quarter domain (top layer). The arrows show the characteristic tensile nature of stresses of clayey soils when subjected to desiccation.

(equations of unsaturated poroelasticity). Numerical tests showed that convergence rates previously found for saturated media are preserved when the models are extended to the (nonlinear) unsaturated case. In addition, we provided two numerical applications, a classical water infiltration case using Richards' equation and a fairly realistic desiccation process of a clayey soil driven by atmospheric evaporation. In both cases, physically coherent results are obtained. Thanks to the AD-based approach, the models presented herein can be extended to include other processes such as scalar transport, chemical reactions, or heat transfer.

Acknowledgement. The funds that made this investigation possible were provided by the Paraguayan Postgraduate Scholarship Program "Don Carlos Antonio López" (Ref 209/2016) and the Norwegian Academy of Science and Letters (VISTA project 6371).

References

[1] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Computational Geosciences*, 6(3-4):405–432, 2002. doi: 10.1023/A:1021291114475.

[2] I. Aavatsmark, G. T. Eigestad, and R. A. Klausen. Numerical convergence of the MPFA O-method for general quadrilateral grids in two and three dimensions. In *Compatible Spatial Discretizations*, pp. 1–21. Springer, 2006. doi: 10.1007/0-387-38034-5_1.

[3] I. Aavatsmark, G. T. Eigestad, R. A. Klausen, M. F. Wheeler, and I. Yotov. Convergence of a symmetric MPFA method on quadrilateral grids. *Computational Geosciences*, 11(4):333–345, 2007. doi: 10.1007/s10596-007-9056-8.

[4] I. Aavatsmark, G. Eigestad, B. Mallison, and J. Nordbotten. A compact multipoint flux approximation method with improved robustness. *Numerical Methods for Partial Differential Equations: An International Journal*, 24(5):1329–1360, 2008. doi: 10.1002/num.20320.

[5] J. Bear. *Dynamics of Fluids in Porous Media*. Dover Publications, 1989.

[6] J. W. Both, M. Borregales, J. M. Nordbotten, K. Kumar, and F. A. Radu. Robust fixed stress splitting for Biot's equations in heterogeneous media. *Applied Mathematics Letters*, 68:101–108, 2017. doi: 10.1016/j.aml.2016.12.019.

[7] J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu. Iterative methods for coupled flow and geomechanics in unsaturated porous media. In M. Vandamme, P. Dangla, J.-M. Pereira, and S. Ghabezloo, eds., *Poromechanics VI*, pp. 411–418. American Society of Civil Engineers, Reston, VA, 2017. doi: 10.1061/9780784480779.050.

[8] J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu. Anderson accelerated fixed-stress splitting schemes for consolidation of unsaturated porous media. *Computers & Mathematics with Applications*, 77(6):1479–1502, 2019. doi: 10.1016/j.camwa.2018.07.033.

[9] J. W. Both, I. S. Pop, and I. Yotov. Global existence of a weak solution to unsaturated poroelasticity. *arXiv preprint arXiv:1909.06679*, 2019.

[10] T. Cajuhi, L. Sanavia, and L. De Lorenzis. Phase-field modeling of fracture in variably saturated porous media. *Computational Mechanics*, 61(3):299–318, 2018. doi: 10.1007/s00466-017-1459-3.

[11] R. F. Carsel and R. S. Parrish. Developing joint probability distributions of soil water retention characteristics. *Water Resources Research*, 24(5):755–769, 1988. doi: 10.1029/wr024i005p00755.

[12] M. A. Celia, E. T. Bouloutas, and R. L. Zarba. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, 26(7): 1483–1496, 1990. doi: 10.1029/WR026i007p01483.

[13] O. Coussy. *Poromechanics*. John Wiley & Sons, 2004.

[14] J. Droniou. Finite volume schemes for diffusion equations: introduction to and review of modern methods. *Mathematical Models and Methods in Applied Sciences*, 24(8):1575–1619, 2014. doi: 10.1142/s0218202514400041.

[15] R. A. Feddes, E. Bresler, and S. P. Neuman. Field test of a modified numerical model for water uptake by root systems. *Water Resources Research*, 10(6):1199–1206, 1974. doi: 10.1029/wr010i006p01199.

[16] D. G. Fredlund. Unsaturated soil mechanics in engineering practice. *Journal of Geotechnical and Geoenvironmental Engineering*, 132(3):286–321, 2006. doi: 10.1061/(ASCE)1090-0241(2006)132:3(286).

[17] R. A. Freeze and J. A. Cherry. *Groundwater*. Prentice-Hall, 1979.

[18] L. Goehring, A. Nakahara, T. Dutta, S. Kitsunezaki, and S. Tarafdar. *Desiccation Cracks and Their Patterns: Formation and Modelling in Science and Nature*. John Wiley & Sons, 2015.

[19] D. Illiano, I. S. Pop, and F. A. Radu. Iterative schemes for surfactant transport in porous media. *Computational Geosciences*, 25:805–822, 2021. doi: 10.1007/s10596-020-09949-2.

[20] E. Keilegavlen, R. Berge, A. Fumagalli, M. Starnoni, I. Stefansson, J. Varela, and I. Berre. PorePy: an open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences*, 25:243–265, 2021. doi: 10.1007/s10596-020-10002-5.

[21] E. Keilegavlen and J. M. Nordbotten. Finite volume methods for elasticity with weak symmetry. *International Journal for Numerical Methods in Engineering*, 112(8):939–962, 2017. doi: 10.1002/nme.5538.

[22] J. Kim, H. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: drained and undrained splits. *Computer Methods*

in Applied Mechanics and Engineering, 200(23–24):2094–2116, 2011. doi: 10.1016/j.cma.2011.02.011.

[23] R. Klausen, F. Radu, and G. Eigestad. Convergence of MPFA on triangulations and for Richards' equation. *International Journal for Numerical Methods in Fluids*, 58(12):1327–1351, 2008. doi: 10.1002/fld.1787.

[24] S. Krogstad, K.-A. Lie, O. Møyner, H. M. Nilsen, X. Raynaud, and B. Skaflestad. MRST-AD – an open-source framework for rapid prototyping and evaluation of reservoir simulation problems. In *SPE Reservoir Simulation Symposium, 23–25 February, Houston, Texas*. Society of Petroleum Engineers, 2015. doi: 10.2118/173317-MS.

[25] K. K. Kumar, K. R. Kumar, and P. Rakhecha. Comparison of Penman and Thornthwaite methods of estimating potential evapotranspiration for Indian conditions. *Theoretical and Applied Climatology*, 38(3):140–146, 1987. doi: 10.1007/bf00868097.

[26] R. B. Lewis and B. A. Schrefler. *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media*. Wiley, 1998.

[27] X. Li and D. Zhang. A backward automatic differentiation framework for reservoir simulation. *Computational Geosciences*, 18(6):1009–1022, 2014. doi: 10.1007/s10596-014-9441-z.

[28] K.-A. Lie. *An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST).* Cambridge University Press, Cambridge, UK, 2019. doi: 10.1017/9781108591416.

[29] F. List and F. A. Radu. A study on iterative methods for solving Richards' equation. *Computational Geosciences*, 20(2):341–353, 2016. doi: 10.1007/s10596-016-9566-3.

[30] J. Lubliner and P. Papadopoulos. *Introduction to Solid Mechanics*. Springer, 2016.

[31] A. Merxhani. An introduction to linear poroelasticity. *arXiv preprint arXiv:1607.04274*, 2016.

[32] N. H. Mondol, J. Jahren, K. Bjørlykke, and I. Brevik. Elastic properties of clay minerals. *The Leading Edge*, 27(6):758–770, 2008. doi: 10.1190/1.2944161.

[33] J. M. Nordbotten. Cell-centered finite volume discretizations for deformable porous media. *International Journal for Numerical Methods in Engineering*, 100(6):399–418, 2014. doi: 10.1002/nme.4734.

[34] J. M. Nordbotten. Finite volume hydromechanical simulation in porous media. *Water Resources Research*, 50(5):4379–4394, 2014. doi: 10.1002/2013wr015179.

[35] J. M. Nordbotten. Stable cell-centered finite volume discretization for Biot equations. *SIAM Journal on Numerical Analysis*, 54(2):942–968, 2016. doi: 10.1137/15m1014280.

[36] P.-O. Persson and G. Strang. A simple mesh generator in MATLAB. *SIAM Review*, 46(2):329–345, 2004. doi: 10.1137/s0036144503429121.

[37] G. F. Pinder and M. A. Celia. *Subsurface Hydrology*. John Wiley & Sons, 2006.

[38] G. F. Pinder and W. G. Gray. *Essentials of Multiphase Flow and Transport in Porous Media*. John Wiley & Sons, 2008.

[39] F. A. Radu and W. Wang. Convergence analysis for a mixed finite element scheme for flow in strictly unsaturated porous media. *Nonlinear Analysis: Real World Applications*, 15:266–275, 2014. doi: 10.1016/j.nonrwa.2011.05.003.

[40] L. A. Richards. Capillary conduction of liquids through porous mediums. *Physics*, 1(5):318–333, 1931. doi: 10.1063/1.1745010.

[41] B. Schrefler, T. Shiomi, A. Chan, O. Zienkiewicz, and M. Pastor. *Computational Geomechanics with Special Reference to Earthquake Engineering*. Wiley, Chichester, England, 1999.

[42] R. A. Stirling. Multiphase modelling of desiccation cracking in compacted soil. PhD thesis, Newcastle University, 2014. URL hdl.handle.net/10443/2492.

[43] M. T. van Genuchten. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal*, 44(5):892–898, 1980. doi: 10.2136/sssaj1980.03615995004400050002x.

[44] J. Varela. Implementation of an MPFA/MPSA-FV solver for the unsaturated flow in deformable porous media. Master's thesis, The University of Bergen, 2018. URL hdl.handle.net/1956/17905.

[45] A. Verruijt. *An Introduction to Soil Mechanics*, Volume 30 of *Theory and Applications of Transport in Porous Media.* Springer, 2017. doi:10.1007/978-3-319-61185-3.

[46] J. Šimůnek and S. A. Bradford. Vadose zone modeling: introduction and importance. *Vadose Zone Journal*, 7(2):581–586, 2008. doi: 10.2136/vzj2008.0012.

[47] J. Šimůnek, M. Šejna, M. T. Van Genuchten, D. Mallants, H. Saito, and M. Sakai. The HYDRUS-1D software package for simulating the one-dimensional movement of water, heat, and multiple solutes in variable-saturated media, 2013. URL www.pc-progress.com/Downloads/Pgm_hydrus1D/HYDRUS1D-4.17.pdf.

Paper B

Porepy: An open-source software for simulation of multiphysics processes in fractured porous media

Keilegavlen E., Berge R., Fumagalli A., Starnoni M., Stefansson I., Varela J., Berre I.

Computational Geosciences **25(1)**, p. 243–265.

doi: 10.1007/s10596-020-10002-5

Computational Geosciences (2021) 25:243–265
https://doi.org/10.1007/s10596-020-10002-5

ORIGINAL PAPER

PorePy: an open-source software for simulation of multiphysics processes in fractured porous media

Eirik Keilegavlen[1] · Runar Berge[1] · Alessio Fumagalli[1,2] · Michele Starnoni[1,3] · Ivar Stefansson[1] · Jhabriel Varela[1] · Inga Berre[1]

Received: 26 August 2019 / Accepted: 10 September 2020 / Published online: 14 October 2020

Abstract

Development of models and dedicated numerical methods for dynamics in fractured rocks is an active research field, with research moving towards increasingly advanced process couplings and complex fracture networks. The inclusion of coupled processes in simulation models is challenged by the high aspect ratio of the fractures, the complex geometry of fracture networks, and the crucial impact of processes that completely change characteristics on the fracture-rock interface. This paper provides a general discussion of design principles for introducing fractures in simulators, and defines a framework for integrated modeling, discretization, and computer implementation. The framework is implemented in the open-source simulation software PorePy, which can serve as a flexible prototyping tool for multiphysics problems in fractured rocks. Based on a representation of the fractures and their intersections as lower-dimensional objects, we discuss data structures for mixed-dimensional grids, formulation of multiphysics problems, and discretizations that utilize existing software. We further present a *Python* implementation of these concepts in the PorePy open-source software tool, which is aimed at coupled simulation of flow and transport in three-dimensional fractured reservoirs as well as deformation of fractures and the reservoir in general. We present validation by benchmarks for flow, poroelasticity, and fracture deformation in porous media. The flexibility of the framework is then illustrated by simulations of non-linearly coupled flow and transport and of injection-driven deformation of fractures. All results can be reproduced by openly available simulation scripts.

Keywords Fractured reservoirs · Mixed-dimensional geometry · Numerical simulations · Multiphysics · Discrete fracture matrix models · Open-source software · Reproducible science

1 Introduction

Simulation of flow, transport, and deformation of fractured rocks is of critical importance to several applications such as subsurface energy extraction and storage and waste disposal. While the topic has received considerable attention in the last decade, the development of reliable simulation tools remains a formidable challenge. Many reasons can be given for this; we here pinpoint four possible causes: First, while natural fractures are thin compared to the characteristic length of the domains of interest, their extent can span the entire domain [1]. The high aspect ratio makes the geometric representation of fractures in the simulation model challenging. Second, the strongly heterogeneous properties of fractures compared to the matrix with respect to flow and mechanics call for methods that can handle strong parameter discontinuities as well as different governing physics for the fractures and the matrix, see for instance [2–4]. Third, phenomena of practical interest tend to involve multiphysics couplings, such as interaction between flow, temperature evolution, geo-chemical effects, and fracture deformation [5]. Correspondingly, there is an ongoing effort to develop and introduce multiphysics

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s10596-020-10002-5) contains supplementary material, which is available to authorized users.

✉ Eirik Keilegavlen
Eirik.Keilegavlen@uib.no

[1] Department of Mathematics, University of Bergen, Pb 7800, 5020 Bergen, Norway

[2] Present address: MOX Laboratory, Department of Mathematics, Politecnico di Milano, via Bonardi 9, 20133 Milan, Italy

[3] Present address: Department of Environment, Land and Infrastructure Engineering, Politecnico di Torino, Turin, Italy

couplings within simulation models [6]. Fourth, fracture networks have highly complex intersection geometries, which must be accounted for in the simulation models. Although the geometry of the walls of individual fractures can be complex by themselves, we will not consider this in any detail, but rather assume that averaged apertures are available at the scale of discretizations.

Traditionally, simulation of flow-driven dynamics in fractured media has been based on two conceptual models. The first is the upscaled representation, where the fracture network geometry and dynamical processes taking place in the network are replaced by equivalent continuum models, which resemble those used in non-fractured porous media. As these models do not resolve the fracture geometry, they are computationally efficient, and have been extended to cover a wide range of multiphysics couplings, as exemplified by the TOUGH2 family of codes [7] as well as PFLOTRAN [8]. The accuracy of the simulations is however highly dependent on the quality of the upscaled model, which in turn depends on the fractured domain's resemblance of a continuous medium with respect to the nature of the physical processes. In practice, the upscaling process ranges from treatable by analytical means for simple fracture geometries and dynamics [9, 10], to extremely challenging in the case of multiphysics couplings and complex fracture geometries [11, 12].

The second traditional class of models, known as the discrete fracture network (DFN) models, is constructed using an explicit representation of the fracture network in the simulation model, while ignoring the surrounding rock mass. The models combine highly accurate representation of dynamics in the fractures with computational efficiency from not having to deal with the rock matrix. DFN simulation models with a high level of sophistication have been developed, notably for coupled flow and transport, see for instance [13–15]. By themselves, DFN models cannot represent processes outside the fracture network; however, the models can be combined with continuum models to achieve fracture-matrix couplings.

The respective limitations of continuum and DFN models have, over the last decade, led to an increased interest in the class of discrete fracture matrix (DFM) models. In DFM models, the fractures are sorted in two classes according to their importance for the dynamics in question [16]. The most important fractures are represented explicitly, while upscaled models are applied for the remaining fractures and the host rock. As such, DFM models represent a flexible compromise between upscaling and explicit representations. The models can represent governing equations in the rock matrix, fractures, and generally also in the intersections between fractures. For computational efficiency, it is common to represent fractures and their intersections as lower-dimensional objects embedded in the three-dimensional rock matrix [17, 18]. We refer to such representation as a mixed-dimensional model [19], and conversely refer to a model of a domain where only a single dimension is considered fixed dimensional.

DFM models can further be divided into two subgroups, according to whether they explicitly represent the fracture surfaces in the computational grid [16]. Models that apply non-conforming gridding include the embedded discrete fracture matrix model (EDFM) [20], and extended finite element methods (XFEM) [21, 22]. These methods avoid the complexities of conforming grid generation discussed below, but must instead incorporate the fracture-matrix interaction in what becomes complex modifications of the numerical method for XFEM [23], or by constructing an upscaled representation, e.g., [24], where the latter approach faces challenges reminiscent of those in continuum-type models. For this reason, our interest herein is DFM methods with conforming grids. Construction of these grids can be challenging for complex fracture networks, particularly in 3d, and the high cell count that may result can put limits in the amount of fractures that can be explicitly represented. Nevertheless, this type of DFM models has been developed for flow and transport, as well as mechanics and poroelasticity, and the explicit representation is particularly useful when the fractures deform. Simulation models that incorporate DFM principles include DuMuX [25], CSMP [26], MOOSE-FALCON [27, 28], OpenGeoSys [29], and Flow123d [30].

The utility of a rapid prototyping framework is illustrated by the wide usage of the Matlab Reservoir Simulation Toolbox (MRST) [31, 32], mainly for non-fractured porous media. Similarly, research into strongly coupled processes in mixed-dimensional geometries will benefit from software of similar flexibility and with a structure tailored to the specific challenges related to fractured porous media.

The goal of this paper is twofold: First, we review challenges related to design of simulation frameworks for multiphysics couplings in mixed-dimensional geometries. Our aim is to discuss design choices that must be made in the implementation of any DFM simulator, including data structures for mixed-dimensional geometries, and representation and discretization of multiphysics problems. Second, we describe a framework for integrated modeling, discretization, and implementation, and an open-source software termed PorePy adhering to this framework. Key to our approach is a decomposition of the geometry into separate objects for rock matrix, individual fractures, and fracture intersections. Governing equations can then be defined separately on each geometric object, as well as on the connection between the objects. This allows for significant code reuse from the discretization of fixed-dimensional problems; thus, our design principles are also applicable to more general PDE software frameworks, such as FEniCS [33], Dune [34], and FireDrake [35]. Furthermore, for scalar and vector elliptic problems (flow and deformation), the models rest on a solid mathematical formulation [36–38].

Built on the mixed-dimensional geometry, PorePy offers several discretization schemes for mathematical models of common processes, such as flow, transport, and mechanical deformation. Multiphysics couplings are easily formulated, and their discretization depends on the availability of appropriate discretization schemes. Moreover, the framework allows for different geometric objects to have different primary variables and governing equations. The software can be used for linear and non-linear problems, with the latter treated by automatic differentiation. PorePy offers automatic gridding of fractured domains in 2d and 3d, relying on the third-party software Gmsh [39] to construct the grid. PorePy is fully open-source (see www.github.com/pmgbergen/porepy) and is released under the GNU General Public License (GPL) version 3.

The paper is structured as follows: In Section 2, we present the principles whereupon we have built the mixed-dimensional framework in PorePy. Section 3 presents models for physical processes central to fractured porous media: single-phase flow, heat transport, and poroelastic rock deformation coupled with fracture deformation modeled by contact mechanics. The implementation of PorePy is presented in Section 4. In Section 5, we benchmark our approach and the PorePy library against well-established test cases. In Section 6, we present two complex applications to illustrate the potential of the framework with respect to advanced physical processes, followed by conclusions in Section 7.

2 Design principles for mixed-dimensional simulation tools

Developing a simulation model for a specific process in mixed-dimensional media requires three main ingredients: A representation of the mixed-dimensional geometry, governing equations for dynamics within and between the geometric objects (rock matrix, fractures, and fracture intersections), and a strategy for discretization and assembly of the equations on the geometry. This in turn leads to decisions on how much of the mixed-dimensional geometry to represent, which type of couplings between different geometric objects to permit, and how to establish communication between the geometric objects.

In this section, we discuss principles for modeling of coupled processes between dimensions in a general context of fractured rocks, together with representation of the geometry in a continuous and discrete setting. The general discussion herein is supplemented by concrete examples of modeling of the important processes presented in Section 3, while discretizations and implementation are discussed in Section 4.

2.1 Representation of a mixed-dimensional geometry

We consider the representation of a fracture network embedded in a 3d domain. The dimension of the fractures is reduced to 2. Similarly, fracture intersections are reduced to 1d objects and intersections of intersection lines to 0d, producing a hierarchy of objects of dimensions 0 to 3. For a fracture network in a 2d domain, the natural simplification applies, i.e., fractures will be objects of dimension 1 and intersections objects of dimension 0. An important modeling choice is which parts of the geometry to represent in the model. We emphasize that, as our focus herein is DFM models with explicit fracture representation, it is assumed that at least the dominating fractures and the matrix will be explicitly represented in the simulation model, and furthermore that the simulation grid will conform to the fractures.

We distinguish between two approaches for the representation of the fracture geometry: The first explicitly represents the full hierarchy of geometric objects (3d–0d). However, for many processes, one can to a good approximation assume that the main dynamics take place in the matrix or in the fractures, while objects of co-dimension more than 1 (intersection lines and points) mainly act as transition zones between fractures. This observation motivates the second approach: The matrix and fractures are represented explicitly, together with some model for direct fracture-fracture interaction.

Representation only of matrix and fractures and not the intersections in some sense constitutes the minimal modification to an existing fixed-dimensional model and has been a popular choice, e.g., for flow and transport problems [40]. The strategy has also been taken a long way towards practical applications, see for instance [41]. There are however drawbacks, notably in the treatment of fracture intersections: Without explicit access to the intersection objects, modeling of interaction between two fractures can be challenging. As an example, for flow, the model does not allow for specifying the permeability of the intersection between two fractures. Significantly, the difficulties tend to increase with increasing complexity of the dynamics, such as countercurrent flow due to gravity and capillary forces, and when transitioning from 2d domains to 3d, i.e., the dimension of the intersections increases from zero to one. This has important consequences for model and method development, as issues related to ad hoc treatment of intersection dynamics may not manifest until relatively late in the development process. For these reasons, we prefer the first approach, where all geometric objects are treated (or "represented") equally, independent of their dimension.

To illustrate our geometry representation, consider Fig. 1a showing three fractures that intersect pairwise along three

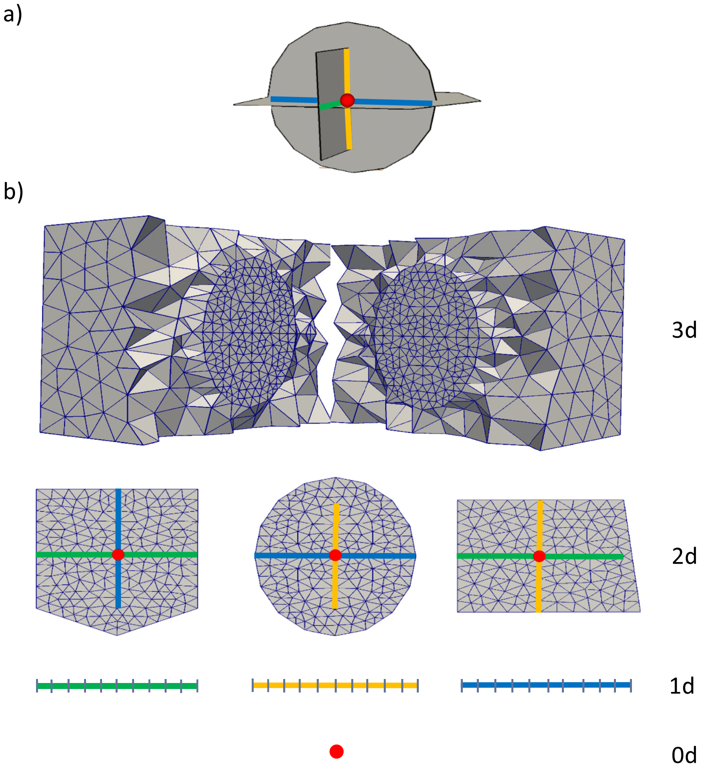


Fig. 1 Conceptual illustration of a fracture network, including grids and lower-dimensional representation. (**a**) Fracture network, the rock matrix is not visualized. (**b**) Grids of all subdomains. Fracture intersections (1d) are represented by colored lines, the 0d grid by a red circle. The 3d grid is cut to expose the circular fracture

lines, which in turn intersect in a point. The fracture network thus defines a set of objects of dimensions $\{0, 1, 2\}$, while the surrounding host medium (not shown) is 3d. We shall refer to each object as a subdomain and denote a generic subdomain by Ω_i. Note that all subdomains of dimension less than 3 are embedded in at least one subdomain of one dimension more, for instance, all lines in the geometry lie on at least two fracture surfaces.

Figure 1b shows the computational grid constructed for each subdomain. The grid on each subdomain conforms to any lower-dimensional subdomains embedded within it, illustrated by the faces in the 3d grid that match the circular fracture. We will discuss grid construction in more detail in Section 4.1.

To finalize the description of the geometry, we introduce the notation for an interface between two subdomains. With reference to Fig. 2, we denote by Ω_h and Ω_l two subdomains one dimension apart so that Ω_l is embedded in Ω_h, and let $\partial_j\Omega_h$ be the part of the boundary of Ω_h that geometrically coincides with Ω_l. Furthermore, we introduce the interface Γ_j on the boundary between $\partial\Omega_h$ and Ω_l. From the dimension reduction, it follows that Γ_j, Ω_l, and $\partial_j\Omega_h$ all coincide geometrically. For completeness, we note that the mathematical framework [36] on which our models are based considers the two sides of Ω_l as

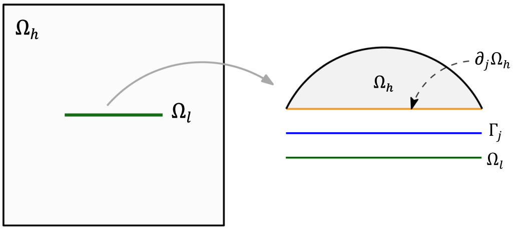


Fig. 2 Mixed-dimensional geometric objects. A higher-dimensional subdomain Ω_h is connected to a lower-dimensional subdomain Ω_l through the interface Γ_j. The part of the boundary of Ω_h geometrically coinciding with Ω_l is denoted by $\partial_j\Omega_h$. The interface Γ_k on the lower side of Ω_l is not shown

different interfaces, Γ_j and Γ_k. Throughout, we will let Γ_j denote a generic interface and use the triplet (Γ_j, Ω_h, Ω_l) to represent an interface and its higher- and lower-dimensional neighbor.

2.2 Permissible coupling structures for mixed-dimensional processes

For modeling purposes, it is important to establish which types of couplings between variables on subdomains and interfaces are permitted. In our framework, we impose the following constraints on the modeling of dynamic processes:

1. There is only coupling between subdomains that are exactly one dimension apart.
2. Interaction between subdomains is formulated as a model on the interface between the subdomains.
3. A model on an interface can depend on variables on the interface and the immediate subdomain neighbors, but not on variables associated with other subdomains or interfaces.

These choices have two important consequences: First, our framework explicitly rules out direct 3d-1d couplings. Second, our model does not permit direct coupling between objects of the same dimension, say, two fractures; the communication must go via a lower- or higher-dimensional object. On the other hand, the imposed constraints make the structure of the equations on a subdomain relatively simple, as the dynamics depend only on variables internal to the subdomain and on neighboring interfaces.

In some cases, it can be of interest to also consider couplings between subdomains of equal dimension, for instance to implement domain decomposition solvers. This can be realized by a secondary partitioning of the subdomains. When such a strategy is applied, the above constraints should be applied only on the interface between subdomains of different dimensions. On interfaces between subdomains of the same dimension, standard continuity conditions can be applied.

3 Model problems

In this section, we use the modeling framework defined in Section 2 to present three sets of governing equations, each of which is of high relevance for fractured porous media: the elliptic pressure equation, fully coupled flow and transport, and fracture deformation coupled with poroelastic deformation of the host medium. Since most of the involved fixed-dimensional processes are well established, our main purpose is to apply the modeling framework described in Section 2 to the mixed-dimensional setting.

We introduce the following notation for variables and subdomains: Variables in a generic subdomain Ω_i are marked by the subscript i, while the subscript j identifies interface variables on Γ_j. For a subdomain Ω_i, the set of neighboring interfaces is split into interfaces towards subdomains of higher dimensions, denoted $\widehat{S}_i$, and interfaces towards subdomains of lower dimensions, denoted by $\check{S}_i$ (see Fig. 3).

Communication between an interface and its neighboring subdomains is handled by projection operators. In the subsequent parts, we will apply four different classes of projections. We indicate the mapping from an interface to the related subdomains by Ξ, with a subscript indicating the index of the interface and a superscript denoting the index of the subdomain, as illustrated in Fig. 4. We also introduce the projection operators from subdomains neighboring of an interface to the interface itself, denoted by the symbol Π with the same convention as before for sub- and superscripts. The actual definition of these objects is scope-dependent and will be specified when needed. The construction of the projection needs to consider the nature of the variable to project, being of intensive or extensive kind, that is, whether the projections should average or sum the variables, respectively.

3.1 Flow in fractured media

We first consider incompressible flow in mixed-dimensional geometries, where we assume a Darcy-type relation between

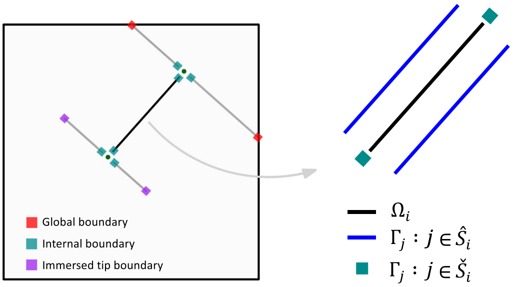


Fig. 3 Intersecting fractures, interfaces, and types of boundary conditions. The 2d domain contains three fractures (1d lines) that intersect in two intersection points (dots). The fractures have three types of boundaries: internal (green squares), immersed tips (purple squares), and endings at the external boundary (red squares). A close-up of the black fracture Ω_i shows the interfaces associated with its higher-dimensional (blue lines) and lower-dimensional (green squares) neighboring subdomains. The sets of such interfaces are denoted respectively by $\widehat{S}_i$ and $\check{S}_i$

the flux and the pressure gradient in all subdomains. The model has been presented several times before, see, e.g., [2, 42, 43].

First, consider a domain with a single interface Γ_j with neighboring subdomains Ω_h and Ω_l. In addition to the pressure p_i and flux q_i in each subdomain, we denote the flux on Γ_j by λ_j and formally write $\lambda_j = \Pi_j^h\, tr\; q_h \cdot n_h$, with n_h the unit normal on $\partial_j\Omega_h$ pointing from Ω_h to Ω_l, and *tr* a suitable trace operator mapping from Ω_h to $\partial_j\Omega_h$, referring to Fig. 4. The strong form of the Darcy problem for Ω_l reads: find (q_l, p_l) such that

$$\begin{array}{l} q_l + \dfrac{\mathcal{K}_l}{\mu_l}\nabla p_l = 0, \\ \nabla \cdot q_l - \Xi_j^l \lambda_j = f_l \end{array}, \qquad (3.1)$$

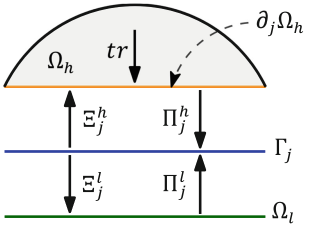


Fig. 4 Representation of a generic coupling between two subdomains. An interface Γ_j is coupled to a higher-dimensional subdomain Ω_h and a lower-dimensional subdomain Ω_l. The projection operators are denoted by Ξ (interface to subdomains) and Π (subdomains to interface) with subscripts indicating the interface and superscripts indicating the subdomain. The trace operator *tr* maps quantities from Ω_h to its boundary $\partial_j\Omega_h$

where the differential operators are defined on the tangent space of Ω_l and Ξ_j^l maps from Γ_j to Ω_l. We have indicated with f_l a source or sink term, μ_l is the fluid viscosity, while $\mathcal{K}_l$ represents the effective tangential permeability tensor scaled by the aperture as described in [42]. An analogous problem is written for (q_h, p_h), with the exception that $\Xi_j^h \lambda_j$ is mapped to a boundary condition on $\partial_j\Omega_h$,

$$q_h \cdot n_h|_{\partial_j\Omega_h} = \Xi_j^h \lambda_j. \qquad (3.2)$$

The flux λ_j is given by an interface condition on Γ_j, which reads

$$\lambda_j + \frac{\kappa_j}{\mu_j}\left(\Pi_j^l p_l - \Pi_j^h tr\; p_h\right) = 0. \qquad (3.3)$$

Here, κ_j indicates the normal effective permeability. Equation (3.3) can be seen as a Darcy law in the normal direction of Γ_j. Different types of boundary conditions can be imposed on the external boundary of Ω_h and Ω_l. Moreover, we impose null flux if Ω_l has an immersed tip boundary.

The extension to problems with many subdomains is now immediate: The flux on an interface is still formulated in terms of variables on its two neighboring subdomains, while for a subdomain Ω_i summation over all neighboring interfaces gives the problem: Find (q_i, p_i) so that

$$\begin{array}{l} q_i + \dfrac{\mathcal{K}_i}{\mu_i}\nabla p_i = 0, \\ \nabla \cdot q_i - \sum_{j\in S} \Xi_j^i \lambda_j = f_i, \\ q_i \cdot n_i|_{\partial j\Omega_i} = \Xi_j^i \lambda_j \quad \forall j \in \check{S}_i \end{array} \qquad (3.4)$$

In the case of $d = 0$, most of the above terms are void, and we are left with the balance between the source term and fluxes from higher dimensions, while for the case $d = 3$, the term involving interface fluxes from higher dimensions is void.

3.2 Fully coupled flow and transport

We next turn to modeling of fully coupled flow and transport, as an example of a multiphysics problem with variable coupling within and between subdomains. We consider a single-phase flow of an incompressible fluid with two components that mix ideally. We denote by c_i the mass fraction of a component associated with Ω_i; the closure relation for the mass fractions implies that we can calculate the other value by $1 - c_i$. The governing equation of the fluid is given by Darcy's law and the fluid mass conservation as in Eq. (3.4). However, we let the viscosity of the fluid depend on the mass fraction,

$$\mu_i = \mu_i(c_i). \tag{3.5}$$

The conservation equations for the components can be formulated as

$$\phi_i \frac{\partial c_i}{\partial t} + \nabla \cdot (c_i q_i - \mathcal{D}_i \nabla c_i) - \sum\nolimits_{j \in S_i} \Xi_j^i \left(\eta_j + \beta_j\right) = g_i. \tag{3.6}$$

Here, ϕ_i represents the effective porosity, $\mathcal{D}_i$ is the effective diffusivity, and g_i denotes sources and sinks. A sum of advective, η_j, and diffusive, β_j, fluxes from the higher-dimensional domains is included in the conservation equation. As for the flow problem, flow over lower-dimensional interfaces Γ_j, $j \in \check{S}_i$, enters as Neumann boundary conditions. We note that the governing equations are coupled via the mass fraction dependency of viscosity and the presence of the Darcy flux in the advective transport.

Let us now consider the interaction between two neighboring subdomains Ω_h and Ω_l via the common interface Γ_j. The flow over Γ_j, denoted by λ_j, is given by Eq. (3.3), where the interface viscosity μ_j is modeled as a function of the mean of the mass fractions on the two sides,

$$\mu_j = \mu_j \left(\frac{\Pi_j^l c_l + \Pi_j^h tr\, c_h}{2} \right). \tag{3.7}$$

The component flux over Γ_j is again governed by an advection-diffusion relation: The diffusion term β_j is, in analogy with the corresponding term for the Darcy flux, given by

$$\beta_j + \delta_j \left(\Pi_j^l c_l - \Pi_j^h tr\, c_h \right) = 0, \tag{3.8}$$

with δ_j representing the effective diffusivity over the interface Γ_j. For the advective term η_j, we introduce an upstream-like operator based on the Darcy interface flux:

$$Up\left(c_h, c_l; \lambda_j\right) = \begin{cases} \Pi_j^h tr\, c_h, & \text{if } \lambda_j \geq 0 \\ \Pi_j^l c_l, & \text{if } \lambda_j < 0. \end{cases} \tag{3.9}$$

With this, the advective interface flux η_j is given by the relation

$$\eta_j - \lambda_j Up\left(c_h, c_l; \lambda_j\right) = 0. \tag{3.10}$$

Finally, global boundary conditions are imposed in the standard way for elliptic and advection-diffusion problems, see, e.g., [44]. Equations (3.5)–(3.10) define the governing equations in all subdomains and on all interfaces, with the exception of 0d domains, where the diffusion operator again is void.

3.3 Poroelastic fracture deformation by contact mechanics

Our final set of model equations considers poroelastic deformation of a fractured medium, where the fractures may open or, if the frictional forces are insufficient to withstand tangential forces on the fracture surface, undergo slip. This process is important in applications such as geothermal energy extraction and CO_2 storage. Modeling of the process is non-trivial due to (i) the coupled poroelastic processes, (ii) the heterogeneous governing equations between subdomains, (iii) the need to use non-standard constitutive laws to relate primary variables during sliding, and (iv) the non-smooth behavior of the constitutive laws in the transition between sticking and sliding and between open and closed fractures. Modeling of this process is an active research field, see, e.g., [45–47], and thus represents an example where the availability of a flexible prototyping framework is highly useful. Due to the complexity in deformation of intersecting fractures, we limit our exposition to media with non-intersecting fractures.

Flow and deformation in the rock matrix, represented by the subdomain Ω_h, are governed by Biot's equations for poroelasticity [48].

$$\begin{aligned} &\nabla \cdot (\mathcal{C}_h \nabla_s u_h - \alpha_h p_h I) = b_h, \\ &\alpha_h \frac{\partial (\nabla \cdot u_h)}{\partial t} + \theta_h \frac{\partial p_h}{\partial t} - \nabla \cdot \left(\frac{\mathcal{K}_h}{\mu_h} \nabla p_h \right) = f_h \end{aligned} \tag{3.11}$$

Here, the first equation represents conservation of momentum, with the acceleration term neglected, while the second equation expresses conservation of mass. The primary variables are the displacement, u_h, and the fluid pressure, p_h. The stiffness matrix $\mathcal{C}_h$ can for linear isotropic media be expressed purely in terms of the first and second Lamé parameters, and the elastic stress can be computed as

$$\sigma_h = \mathcal{C}_h \nabla_s u_h,$$

where ∇_s is the symmetric gradient. Furthermore, α_h is the Biot constant, I the second-order identity tensor, b_h denotes body forces, and θ_h the effective storage term. We also assume boundary conditions are given on the global boundary.

Next, to model relative motion of the fracture walls, it is necessary to consider both interfaces between Ω_h and Ω_l. In a slight abuse of notation, we will let u_j denote the displacement variable on both interfaces. We emphasize that u_j is a vector in $\mathbb{R}^n$, that is, it represents the displacement in both the tangential and normal direction of Ω_l. We will require continuity between u_h and u_j, expressed as $\Pi_j^h tr\, u_h = u_j$, where we recall that the trace operator maps to $\partial_j\Omega_h$. We also introduce the jump in displacement, $[\![u_j]\!]$, between the two interfaces on opposing sides of Ω_l (see Fig. 5). The jump is decomposed into the tangential jump $[\![u_j]\!]_\tau$ and the normal jump $[\![u_j]\!]_n$.

The mechanical state in Ω_l is described by the contact traction σ_l, which also is a vector in $\mathbb{R}^n$, with normal and tangential components $\sigma_{l,n}$ and $\sigma_{l,\tau}$, respectively. Our model also includes fluid flow in the fracture Ω_l, which is governed by conservation of mass

$$\frac{\partial}{\partial t}\left(a\left([\![u_j]\!]\right)\right) + \theta_l \frac{\partial p_l}{\partial t} - \nabla\cdot\left(\frac{\mathcal{K}_l}{\mu_l}\nabla p_l\right) - \Xi_j^l \lambda_j = f_l. \quad (3.12)$$

Here, the time derivative of the aperture $a([\![u_j]\!]) = a_0 - [\![u_j]\!]_n$ represents changes in the available volume due to changes in the displacement jump, with a_0 denoting the residual hydraulic aperture. The negative sign on the normal jump is related to the sign convention in (3.14) below. As in the previous sections, the relation between the fluid pressures in Ω_h and Ω_l is governed by a flux law of the type (3.3).

The relation between σ_l and $[\![u_j]\!]$ is modeled by borrowing techniques from contact mechanics as summarized here (for a full discussion, see [49]). Balance of tractions between the poroelastic stress in Ω_h and the contact traction in Ω_l is for the two sides expressed as

$$\begin{aligned}
\Pi_{j_1}^h\, n_h\cdot(\sigma_h - \alpha_h p_h I) &= \Pi_{j_1}^l \sigma_l - \left(\Pi_{j_1}^h\, n_h\right)\cdot\left(I\Pi_{j_1}^l \alpha_l p_l\right)\\
\Pi_{j_2}^h\, n_h\cdot(\sigma_h - \alpha_h p_h I) &= -\,\Pi_{j_2}^l \sigma_l - \left(\Pi_{j_2}^h\, n_h\right)\cdot\left(I\Pi_{j_2}^l \alpha_l p_l\right)
\end{aligned} \quad (3.13)$$

The contact traction is zero whenever the normal displacement jump is nonzero, that is

$$[\![u_j]\!]_n \le 0, \qquad \sigma_{l,n} \le 0, \qquad [\![u_j]\!]_n \sigma_{l,n} = 0. \quad (3.14)$$

For closed fractures, the motion in the tangential direction is controlled by the ratio between the tangential traction $\sigma_{l,\tau}$ and the maximum available frictional traction $F\sigma_{l,n}$, where F is the friction coefficient. The time derivative of the displacement jump is zero until the frictional traction is overcome; for larger tangential tractions, the time derivative of the displacement jump and tangential traction are parallel:

$$\begin{aligned}
&\|\sigma_{l,\tau}\| \le -F\sigma_{l,n},\\
&\|\sigma_{l,\tau}\| < -F\sigma_{l,n} \rightarrow [\![u_j]\!]_\tau = 0,\\
&\|\sigma_{l,\tau}\| = -F\sigma_{l,n} \rightarrow \exists\gamma\in\mathbb{R}, \sigma_{l,\tau} = -\,\gamma^2 [\![u^j]\!]_\tau.
\end{aligned} \quad (3.15)$$

Here $\|\cdot\|$ represents the Euclidean norm, and $[\![u_j]\!]_\tau$ the sliding velocity. We emphasize that the tangential contact conditions are formulated in terms of the contact traction σ_l, with no contribution from the fluid pressure p_l.

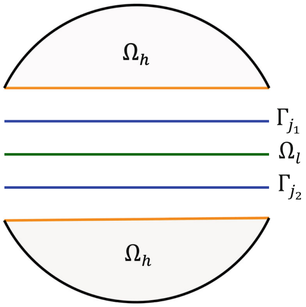


Fig. 5 Illustration of a lower-dimensional domain, Ω_l, that has two interfaces, Γ_{j_1} and Γ_{j_2}, with a higher-dimensional domain, Ω_h

4 Implementation

This section describes the implementation of the mixed-dimensional simulation framework outlined above in the open-source simulator PorePy. Our emphasis is on three topics that are particular to this type of DFM simulation models: Gridding, discretization of subdomain couplings, and how to deal with parameters, variables, and linear systems for multiphysics problems that are defined on an arbitrary number of subdomains and dimensions. The ability to treat these components with relatively simple input is the main distinguishing feature of PorePy, and thus, the section gives an overview of the important properties of the implemented simulator.

Figure 6 displays the main components of PorePy, with emphasis on the mixed-dimensional aspects of the code. The implementation follows the principles of locality of variables and equations described in the previous sections. Specifically, equations and discretizations are assigned on individual subdomains, and the implementation of specific discretization schemes closely resembles that applied to fixed-dimensional problems. Similarly, the stencil of interface couplings is

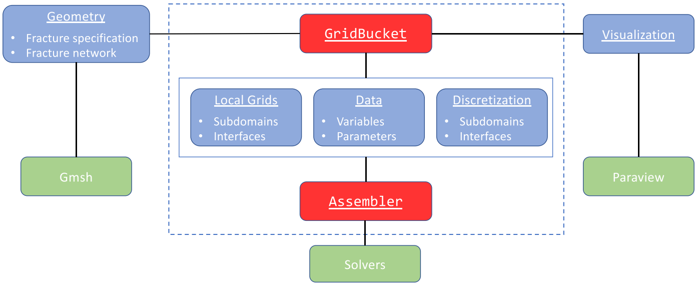


Fig. 6 Outline of the architecture of PorePy: The main mixed-dimensional components are the GridBucket class, which is a combined grid and data manager, and the Assembler class, which acts as a degree of freedom manager. Variables, parameters, and discretizations are local to subdomains and interfaces. Geometry specification and grid construction is handled in part by communication with Gmsh, while visualization is available through export to Paraview. Green boxes represent external dependencies

limited to the interface and the immediate neighboring subdomains. The connection between the subdomains is handled in a top-down manner and implemented in two core classes: The GridBucket class keeps track of the relation between neighboring subdomains and interfaces, and it also acts as a facility for storage of parameters and variables. The Assembler class can be considered a global degree of freedom manager which also has methods for global discretization and assembly. These core mixed-dimensional components are supplemented by functionality for grid construction, assisted by Gmsh, while visualization and linear solvers must be handled by external packages.

A typical workflow for a mixed-dimensional simulation will consist of the following steps:

1. Specify the problem geometry. Use this to create a GridBucket object, that is, a mixed-dimensional grid.
2. On the individual subdomains and interfaces in the GridBucket, specify variables, parameters, and discretizations (thus implicitly define governing equations).
3. Create an Assembler object, use this for initial discretization and assembly of linear system.
4. Solve the mixed-dimensional problem.

Depending on the problem characteristics, the last point can entail non-linear iterations, time stepping, etc.

The rest of this section presents design choices and concrete implementation details of the individual steps. As an illustration of the usage of the resulting simulation framework, Fig. 7 provides an example PorePy code for the setup, discretization, and solution of the mixed-dimensional compressible flow problem. We emphasize that to change the problem geometry, e.g., the fracture network, it is sufficient to change the pink section, while governing equations, parameters, and/or discretization schemes are altered by modifications to the green section. Several examples of the latter are given in Section 5.

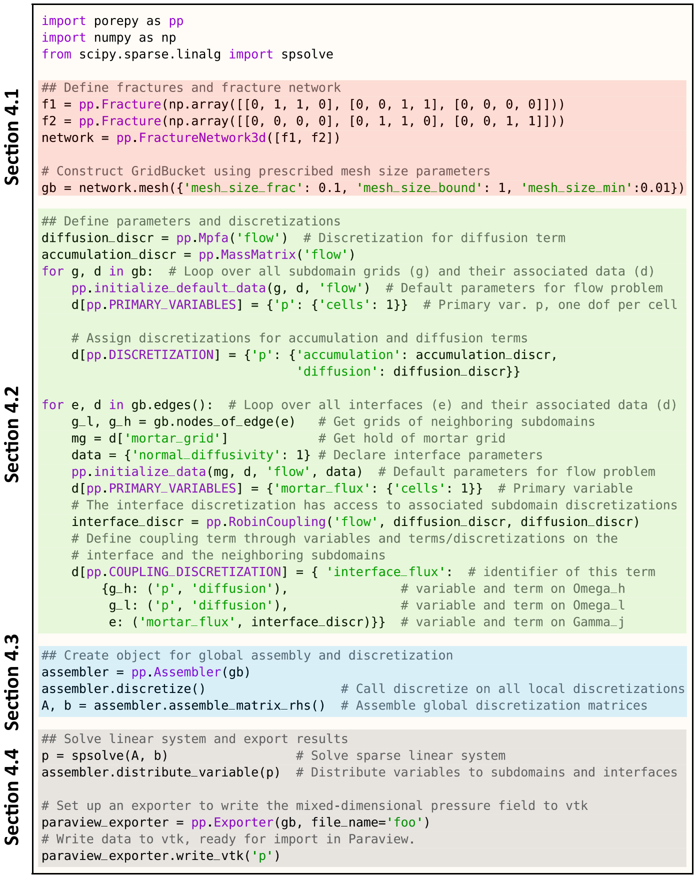
4.1 Mixed-dimensional geometry and gridding

Grid construction is one of the main technical bottlenecks for the application of conforming DFM models. The translation of a geometric description of the fracture network into a computational grid consists of three steps: Identification of intersection lines and points, construction of the mixed-dimensional grid, and post-processing of the grid into a format that is suited for the discretization approaches described in Section 4.2. The first and third of these tasks are technically challenging, and one of the strengths of PorePy is that it provides a robust implementation with a simple interface. The second item, grid construction, is a highly advanced research topic in its own; in PorePy, this is handled by a Gmsh backend.

4.1.1 Geometry processing

In PorePy, fractures are described as lines (for 2d domains) or convex planar polygons (in 3d). Curved objects are not supported, as this would significantly complicate the task of identifying intersections; however, piecewise linear approximations are possible. The fractures are specified by their endpoints (in 2d) or vertexes (in 3d). Individual fractures are collected into FractureNetwork2d and FractureNetwork3d classes.

Before passing the fracture network to a gridding software, all fracture intersections must be found. In principle, the computation of fracture intersections is straightforward, following


Section 4.1
Section 4.2
Section 4.3
Section 4.4

```python
import porepy as pp
import numpy as np
from scipy.sparse.linalg import spsolve

## Define fractures and fracture network
f1 = pp.Fracture(np.array([[0, 1, 1, 0], [0, 0, 1, 1], [0, 0, 0, 0]]))
f2 = pp.Fracture(np.array([[0, 0, 0, 0], [0, 1, 1, 0], [0, 0, 1, 1]]))
network = pp.FractureNetwork3d([f1, f2])

# Construct GridBucket using prescribed mesh size parameters
gb = network.mesh({'mesh_size_frac': 0.1, 'mesh_size_bound': 1, 'mesh_size_min':0.01})

## Define parameters and discretizations
diffusion_discr = pp.Mpfa('flow')  # Discretization for diffusion term
accumulation_discr = pp.MassMatrix('flow')
for g, d in gb:  # Loop over all subdomain grids (g) and their associated data (d)
    pp.initialize_default_data(g, d, 'flow')  # Default parameters for flow problem
    d[pp.PRIMARY_VARIABLES] = {'p': {'cells': 1}}  # Primary var. p, one dof per cell

    # Assign discretizations for accumulation and diffusion terms
    d[pp.DISCRETIZATION] = {'p': {'accumulation': accumulation_discr,
                                  'diffusion': diffusion_discr}}

for e, d in gb.edges():  # Loop over all interfaces (e) and their associated data (d)
    g_l, g_h = gb.nodes_of_edge(e)   # Get grids of neighboring subdomains
    mg = d['mortar_grid']            # Get hold of mortar grid
    data = {'normal_diffusivity': 1} # Declare interface parameters
    pp.initialize_data(mg, d, 'flow', data)  # Default parameters for flow problem
    d[pp.PRIMARY_VARIABLES] = {'mortar_flux': {'cells': 1}}  # Primary variable
    # The interface discretization has access to associated subdomain discretizations
    interface_discr = pp.RobinCoupling('flow', diffusion_discr, diffusion_discr)
    # Define coupling term through variables and terms/discretizations on the
    # interface and the neighboring subdomains
    d[pp.COUPLING_DISCRETIZATION] = { 'interface_flux':  # identifier of this term
        {g_h: ('p', 'diffusion'),              # variable and term on Omega_h
         g_l: ('p', 'diffusion'),              # variable and term on Omega_l
         e: ('mortar_flux', interface_discr)}}  # variable and term on Gamma_j

## Create object for global assembly and discretization
assembler = pp.Assembler(gb)
assembler.discretize()                  # Call discretize on all local discretizations
A, b = assembler.assemble_matrix_rhs()  # Assemble global discretization matrices

## Solve linear system and export results
p = spsolve(A, b)                 # Solve sparse linear system
assembler.distribute_variable(p)  # Distribute variables to subdomains and interfaces

# Set up an exporter to write the mixed-dimensional pressure field to vtk
paraview_exporter = pp.Exporter(gb, file_name='foo')
# Write data to vtk, ready for import in Paraview.
paraview_exporter.write_vtk('p')
```

Fig. 7 Setup of a full PorePy simulation, illustrated by a mixed-dimensional compressible flow problem solved with a single time step. The background colors indicate different simulation stages, which are discussed in detail in the indicated subsections

for instance [50]. However, to reduce the complexity of the grid construction and limit the number of cells in the resulting grid, it can be useful to alter the geometry to avoid small details, such as almost intersecting fractures. PorePy automatically merges objects that are closer than a user-specified tolerance, and also cuts dangling fracture ends. While such modifications can alter the connectivity of the network, we have found that it is a critical ingredient for dealing with fracture networks that originate from sources that have not removed such small details, for instance networks exported from geological processing software or stochastic fracture network generators.

4.1.2 Gridding

The computational grid should conform to all fractures, and by extension also to their intersection lines and points. This is a difficult problem; however, algorithms [51–53] and high-quality implementations [54, 55] are available. PorePy relies on Gmsh [39] for the grid construction, as this allows for a unified approach in both 2d and 3d domains. While Gmsh allows for a nuanced specification of grid sizes, only a limited set of this functionality is exposed in the PorePy interface: A grid size can be set for the fracture network and the far field; more advanced settings can be accessed by direct manipulations in Gmsh. Still, the specified geometry implicitly sets conditions on the grid size; if the fracture network contains fractures that are close relative to the specified grid size, Gmsh will attempt to construct a grid with reasonable quality, and thereby override the user preferences if necessary.

4.1.3 Construction of grids, mortar grids, and projection operators

The grids provided by Gmsh must be post-processed to be of use for our mixed-dimensional simulations. First, grids for individual subdomains must be extracted. Second, mortar grids must be constructed on the interface between subdomain grids, together with projection operators between the grids. Third, the resulting sets of grids must be arranged in the mixed-dimensional GridBucket.

Subdomains of different dimensions can be identified from Gmsh tags that for each cell identify the geometric object to which the cell belongs (matrix, fracture, or intersection). However, to avoid direct connection between cells that lie on different sides of lower-dimensional objects, faces must be split, and nodes duplicated before the grids are arranged in the GridBucket. This process is illustrated in Fig. 8, which also shows the resulting lower-dimensional grids. Note that while all (d–1)-dimensional faces are split in two, the number of duplicates of a node depends on whether it is located on an intersection, a fracture tip or a global boundary, or in the interior of the subdomain. After this modification, the cells that belong to the same geometric objects are collected into subdomain grids. These are implemented as standard fixed-dimensional grids, so that when a discretization scheme is applied to a subdomain, this is indistinguishable from the traditional fixed-dimensional operation. In this spirit, the grid structure used for individual grids is agnostic to spatial dimension, with an implementation heavily inspired by that of MRST [32].

The mortar grids constructed under post-processing of the Gmsh output are associated with the interfaces. They match with the lower-dimensional grid, and thereby also with the split faces of the higher-dimensional grid. The mortar grids also have methods for the construction of projection matrices between themselves and the lower- and higher-dimensional neighboring subdomains, with separate methods for the mapping of extensive and intensive quantities. Only the lowest order projection operators are available in PorePy, which for matching grids simply identify the split faces of Ω_h with cells in Γ_j, and cells in Γ_j with cells in Ω_l. However, non-matching grids can be introduced by replacing individual subdomain and mortar grids. Specifically, computational speedups can often be achieved by combining fine grids in fractures, which are often the main venue for dynamical processes, with relatively coarse grids in the matrix. During the replacement, the projection operators are automatically updated to account for the resulting non-matching grids.

The individual subdomains and mortar grids are collected in the GridBucket class. This is implemented as a graph, where each subdomain grid Ω_i defines a node, while the interface Γ_j is represented as an edge in the graph, and is identified by the pairing of its neighboring subdomains (Ω_h, Ω_l). In addition to keeping track of geometric information, the GridBucket also provides flexible data storage in the form of dictionaries on subdomains and interfaces. These are used for parameters, discretizations, simulation results, and other data if relevant.

4.2 Primary variables, parameters, and discretization

To define a problem to be discretized in PorePy, one must define primary variables, governing equations, and problem

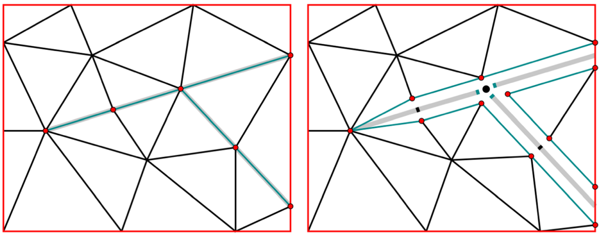


Fig. 8 The process of splitting the faces and nodes of the grid. The faces and nodes of the 2d grid that coincide with the 1d grids (gray lines) are split and define an internal boundary of the grid. Similarly, the faces and nodes of the 1d grids that coincide with the 0d grid (black dot) are split. Note that the split nodes and faces coincide geometrically but have been shifted in the right figure for illustrative purposes

parameters. PorePy is designed to allow for maximum flexibility in these specifications. Variables and parameters are defined on individual subdomains and interfaces. Governing equations are specified in terms of their discretizations: Each variable can be assigned one or several discretizations corresponding to different terms in the equation. As with the variable specification, discretizations are specified locally on subdomains and interfaces, thus heterogeneous governing equations or discretization schemes can readily be assigned. It is up to the user to ensure that the specified combination of variables, equations, and discretizations is mathematically well posed on the given mixed-dimensional grid.

In terms of implementation, the data structures for parameters and solution vectors are stored locally to each subdomain and interface. Specifically, variables are represented as numpy arrays and parameters as a combination of numpy arrays and dedicated classes.

4.2.1 Discretization classes

For the implementation of discretizations, it is useful to differ between the schemes themselves, their implementation, and the application of a discretization object to a specific grid and parameter set, which produces a discretization matrix. All discretization schemes are implemented as classes which are designed to act on individual subdomains or interfaces. In most cases, there is a one-to-one correspondence between terms in the governing equations and discretization. As an example, the compressible flow equation on a subdomain will be specified by assigning discretizations of the accumulation and diffusion term to a pressure variable, as is shown in Fig. 7.

A compatible discretization class should implement a method for discretization, which computes coefficients that will enter into a discretization matrix. Furthermore, the class needs a method for assembly of matrix and right-hand side. The act of discretization and assembly should together produce a local discretization matrix, usually in the form of a sparse matrix represented using the SciPy library and a right-hand side represented as a numpy array.

There are important differences between discretization classes for subdomains and interfaces: Subdomain discretizations have access only to the subdomain grid and its associated data and assemble a matrix local to the subdomain. An interface discretization is responsible for coupling variables on the neighboring subdomains, and it therefore has access to the relevant subdomain discretizations and data in addition to information local to the interface. Thus, an interface discretization may put additional requirements on a subdomain discretization, see Section 4.2.2 for an example. The assembly method in the interface discretization should treat both the interface equation and the discrete couplings of the interface law to the neighboring subdomains.

In PorePy, subdomain discretization schemes are available for diffusion, advection, and mechanical deformation, as well as mass matrices for accumulation terms. Specifically, diffusion processes can be discretized by the lowest order Raviart-Thomas mixed finite elements combined with a piecewise constant pressure approximation (RT0-P0) [56], the lowest order mixed virtual element method (MVEM) combined with a piecewise constant pressure approximation [57, 58], and by two finite volume schemes: the two- and multipoint flux approximations (TPFA and MPFA, respectively). Advection terms can be discretized by a first-order upstream scheme. Mechanical deformation is discretized by the multipoint stress approximation (MPSA) [59, 60], also extended to poroelasticity [61] and thermoporoelasticity [62].

On interfaces, discretization schemes in PorePy cover the interface diffusion law (3.3), and an upstream scheme for the advection term (3.9). The discretization of the contact mechanics (Eqs. (3.14) and (3.15)) is implemented by a semismooth Newton method to deal with the discontinuities in the solution, for details we refer to [49, 63]. The available discretizations on subdomains and interfaces can also be used as building blocks for more complex problems; for instance, the simulations of thermo-poroelasticity with fracture deformation reported in [64] utilized several of the discretization schemes mentioned above.

In the following, we present the implementation of two examples of combined subdomain and interface discretizations, allowing us to discuss different aspects in the design and implementation of mixed-dimensional problems.

4.2.2 Subdomain coupling for discretization of mixed-dimensional flow

3.1, focusing on the division of responsibilities between subdomain and interface discretizations. The discretization of the interface law (3.3) is implemented in the class RobinInterfaceLaw, which in itself is simple, but has an instructive approach to communication with the adjacent subdomain discretizations. From the model in Section 3.1, we see that for a discretization on a generic subdomain Ω_i to interact with the interface problem, we need to provide operators which:

1) Handle Neumann boundary data on the form $\Xi^i_j\lambda_j$ for all interfaces Γ_j for which Ω_i is the higher-dimensional neighbor.
2) Handle source terms $\Xi^i_j\lambda_j$ from interfaces Γ_j for which Ω_i is the lower-dimensional neighbor.
3) Provide a discrete operator $tr\, p_i$ to be combined with Π^i_j to project the pressure to interfaces $\Gamma_j,\ j\in\check{S}_i$.
4) Provide a pressure p_i that can be projected to interfaces Γ_j, $j\in\widehat{S}_i$ using Π^i_j.

RobinInterfaceLaw assumes that the subdomain discretization has dedicated methods, with specified names, that handle each of these four operations. Thus, any discretization class aimed at individual subdomains can be made compatible with RobinInterfaceLaw, and thus applicable to mixed-dimensional problems, provided the four required methods are implemented. Moreover, all of these are readily available in any reasonable implementation of a discretization scheme for elliptic equations. Examples of how RobinInterfaceLaw is set up to interact with subdomain discretizations can be found in Figs. 7 and 10.

It is instructive to write out the structure of the coupled system for our case with two subdomains Ω_h and Ω_l separated by an interface Γ_j. Denote by y_h, y_l, and ξ_j the vectors of discrete unknowns in Ω_h, Ω_l, and on Γ_j, respectively. As we make no assumptions that the same discretization scheme is applied in both subdomains, these may contain different sets of unknowns. The discrete system can then be represented on the generic form

$$\begin{pmatrix} A_h & 0 & N_h\Xi_j^h \\ 0 & A_l & S_l\Xi_j^l \\ -\Pi_j^h P_h & \Pi_j^l P_l & M_j \end{pmatrix} \begin{pmatrix} y_h \\ y_l \\ \xi_j \end{pmatrix} = \begin{pmatrix} f_h \\ f_l \\ 0 \end{pmatrix}. \tag{4.1}$$

Here, A_h and A_l are the fixed-dimensional discretizations on the subdomains and f_h and f_l the corresponding source and sink terms. N_h is the discretization of Neumann boundary conditions on Ω_h, and S_l is the discretization of source terms in Ω_l. Furthermore, P_h provides a discrete representation of the pressure trace operator on Ω_h and P_l gives the pressure unknowns in Ω_l; the latter is an identity operator for the integral formulations presented on primal form and strips away flux unknowns in the dual formulation. Finally, M_j represents the normal permeability term in (3.3) and is discretized directly by RobinCoupling. In accordance with the second constraint on mixed-dimensional modeling discussed in Section 2.2, there is no direct coupling between Ω_h and Ω_l as seen from the 0 entries in the matrix.

The PorePy implementation of the above method represents the mortar variable by piecewise constant functions. Our implementation for the coupled mixed-dimensional problem relies on the analysis carried out in [39], which provides a theoretical background to obtain a stable global scheme with full flexibility in choosing heterogeneous discretization schemes between the subdomains. We also note that the interface discretization for many other classes of equations, such as the advection-diffusion problem presented in Section 3.2, follows a similar approach.

4.2.3 Subdomain couplings for contact mechanics in poroelastic media

As a second example of the matrix structure produced by a subdomain and interface coupling, we consider the model for fracture deformation introduced in Section 3.3. This can be considered a complex model, in that the traction balance on the interface involves multiple variables on Ω_h, Ω_l, and Γ_j. Specifically, the equations for the momentum balance presented in Section 3.3 can be represented in matrix form as

$$\begin{pmatrix} A_h & B_h & D_h\Xi_j^h & 0 & 0 \\ 0 & 0 & U_l\Xi_j^l & 0 & T_l \\ \Pi_j^h T_h & \Pi_j^h G_h & \Pi_j^h S_h \Xi_j^h & -\Pi_j^l G_l & \pm\Pi_j^l \end{pmatrix} \begin{pmatrix} u_h \\ p_h \\ u_j \\ p_l \\ \sigma_l \end{pmatrix} = \begin{pmatrix} b_h \\ r \\ 0 \end{pmatrix}. \tag{4.2}$$

Here, the first row represents the momentum balance with the contribution of the mortar displacement variables on the momentum balance in Ω_h. In practice, this takes the form of a Dirichlet boundary condition discretized as D_h, while A_h, B_h, and b_h represent discretization of poroelasticity in Ω_h. In the second row, the matrices U_l and T_l represent the linearized fracture conditions, i.e., the relation between u_j and σ_l stated in Eqs. (3.14) and (3.15), with contributions from the previous Newton iteration and time step entering in $r = r(u_j, \sigma_l)$. The third row represents Newton's third law over the interfaces, and thus is a discretization of Eq. (3.13). The first three terms provide the traction on the two fracture walls reconstructed from the variables on $\partial_j\Omega_h$ and Γ_j, where S_h represents a mapping from the Dirichlet boundary condition to tractions. The two last terms relate these tractions to the variables in Ω_l, where G_l represents $n_h\alpha_l$, while the ± in the last term accounts for the fracture side. We emphasize that neither the inter-dimensional contributions to mass conservation nor the coupling for mass conservation is included in (4.2); this is handled by the corresponding internal subdomain discretizations and additional coupling discretizations in the form discussed in Section 4.2.2.

In terms of implementation, the interface equations in (4.2) are in fact split into three different classes: One which handles the interaction between u_h, u_j, and σ_l and two that represent the fluid traction on Γ_j from p_h and p_l, respectively. The most interesting of these classes is the first, termed PrimalContactCoupling, which is used for purely mechanical problems; the discretization of the contact problem that produces the matrices U_l and T_l for the current state of $[\![u_j]\!]$ and σ_l is outsourced to a separate class ColoumbContact. An illustration of how PrimalContactCoupling is set up to interact with the surrounding variables and discretizations is given in the context of Sneddon's problem of fracture deformation (see Fig. 16 in Section 5.3).

4.3 Global assembly of mixed-dimensional multiphysics problems

As discussed in Section 4.2, PorePy requires only specification of variables and discretizations locally on subdomains and interfaces. The global organization is left to the Assembler class, which has the following responsibilities: First, to assign a global numbering of the degrees of freedom of all local variables. Second, to apply all assigned discretization schemes. Third, to assemble the sparse global linear system. The user interface to the Assembler is simple;

numbering of degrees of freedom is handled in the object initialization, while the class has dedicated methods for discretization and assembly. The underlying implementation of these methods is elaborate and involves nested loops over the GridBucket. For global discretization, all local discretization objects are identified, and their respective discretization methods invoked. In the assembly operation, the local discretization matrices are placed in the global linear system according to the degree of freedom of the associated local variable(s).

It is instructive to consider the structure of the global linear system in the setting of a multiphysics problem with more than one primary variable. It has a double block structure, with one set of blocks stemming from the geometric division into subdomains and interfaces. Within each subdomain and interface, there is a second set of blocks, with one block per variable or variable pair (for off-diagonal blocks). This information, which is useful for design of tailored preconditioners and linear solvers as well as post-processing and visualization, can be accessed through the Assembler. We emphasize that the implementation of the Assembler is general in the sense that it can be applied to new discretizations and governing equations without modification.

The bottom-up approach to the assembly of variables and discretizations to some degree favors flexibility over computational speed. The overhead in construction and manipulation of matrices, independent of matrix size and separate from the cost of discretization, is minor but can become notable when repeated many times, e.g., in time-dependent and non-linear problems. For problems with many subdomains, the cost in using local assembly can become prohibitively high. Specifically, the cost has been pronounced in simulations of non-linearly coupled flow and transport, as reported in [65] and also in Section 6.1. As a remedy, which is also compatible with the automatic differentiation (AD) module in PorePy, the Assembler also provides methods to construct global discrete operators.

4.4 Solvers and visualization

PorePy has no native support for linear solvers, but instead relies on external libraries for solving linear systems. The structure of the linear systems obtained for mixed-dimensional is non-standard compared with that of similar fixed-dimensional problems. Thus, if the linear system is to be solved by iterative methods, traditional preconditioners cannot be expected to perform well, and specialized methods may be preferable. Preconditioners for mixed-dimensional problems are an immature research field, see however [66, 67] for examples on how PorePy can be combined with dedicated solvers for mixed-dimensional problems.

Finally, visualization is handled by an export filter to the vtk/vtu format, which can be read for instance by Paraview [68]. To aid analysis of simulation results, the export preserves the link between the data and its associated dimensions.

5 Validation

In this section, we validate our modeling framework and its implementation in PorePy by probing discretization schemes, multiphysics problems, and time-dependent problems through three test cases: a benchmark for flow problems in 2d fractured media, Mandel's problem for poroelasticity, and Sneddon's problem for fracture deformation in elastic media. The cases thus supplement previous testing of PorePy, reported in [38, 69–71]. The supplementary material provides detailed setups, including parameters, for all simulations in Sections 5 and 6. Scripts that reproduce all results reported herein can be accessed at [72], see that reference or the supplementary material for installation instructions.

5.1 Flow in 2d fractured porous media

To validate the mixed-dimensional flow discretization, we consider Benchmark 3 of [73], which describes the incompressible single-phase flow problem in a fractured domain presented in Section 3.1. The fracture network contains intersecting and isolated fractures (see Fig. 9). The network contains both highly conductive and blocking fractures, see the supplementary material for parameter details.

The aim of this case is twofold — we benchmark our code against well-established methods in the literature and illustrate PorePy's flexibility in assigning heterogeneous subdomain discretizations. We consider four groups of discretization schemes and simulation grids: first, three homogeneous (the same for all the subdomains) discretizations: TPFA, MPFA, and RT0-P0. Second, a case with the MVEM, where the cells of the rock matrix are constructed by a clustering procedure starting from a more refined simplicial grid, see [70] for details. Third, two heterogeneous discretizations where RT0-P0 and MVEM for the rock matrix are combined with TPFA for the fractures. Fourth, a case where the fracture grid is twice as fine as the matrix grid, with the mortar grids non-conforming to the surrounding grids (labeled Non-Matching) discretized using the RT0-P0 scheme. We use simplex grids in all cases that do not involve MVEM. A code snippet that highlights the assignment of heterogeneous discretizations is given in Fig. 10.

Figure 9 shows the domain with fractures, boundary conditions, and a representative numerical solution. The figure also depicts a plot of the pressure along the line (0, 0.5) − (1, 0.9). We observe good agreement between the solutions obtained in PorePy and the reference solution of [73], which is a solution of the equi-dimensional problem computed on a very fine grid. We also perform a refinement study using a sequence of three grids to compute the error relative to the

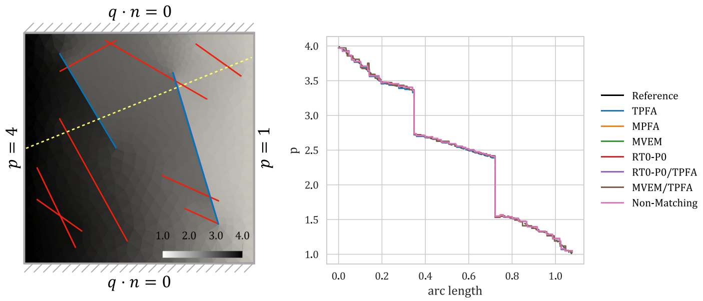


Fig. 9 Left: A solution obtained with MPFA on the coarsest grid showing the fracture network and the problem setup. The red lines represent conductive fractures whereas the blue lines are blocking fractures. The yellow line indicates the line of the pressure profile. Right: Pressure profiles for the discretization schemes used in the validation

reference solution, as done in the original benchmark. Figure 11 shows the decay of the normalized L^2 error for the rock matrix and the union of the fracture subdomains. In the former, we notice a first order of convergence for all the considered methods. The convergence rate for the fracture subdomains is sublinear, as was also observed in the original benchmark.

5.2 Mandel's problem in poroelasticity

The next test case considers a poroelastic material, with a setup defined by Mandel's problem [74, 75], for which an analytical solution is available. While the problem geometry does not include lower-dimensional objects, the case tests the implementation of the poroelastic code and shows the framework's flexibility to

```python
matrix_discr = pp.RT0('flow')     # Discretization in the matrix
fracture_discr = pp.Tpfa('flow')  # Discretization in fractures and intersections

for g, d in gb:
    if g.dim == 2:  # This is the matrix grid
        # RT0 has both cell and face unknowns
        d[pp.PRIMARY_VARIABLES] = {'p': {'cells': 1, 'faces': 1}}
        d[pp.DISCRETIZATION] = {'p': {'diffusion': matrix_discr}}
    else:  # Fracture or fracture intersection grid.
        # TPFA has only cell unknowns
        d[pp.PRIMARY_VARIABLES] = {'p': {'cells': 1}}
        d[pp.DISCRETIZATION] = {'p': {'diffusion': fracture_discr}}

for e, d in gb.edges():  # Loop over all interfaces (e) and their associated data (d)
    g_l, g_h = gb.nodes_of_edge(e)  # Get grids of neighboring subdomains
    d[pp.PRIMARY_VARIABLES] = {'mortar_flux': {'cells': 1}}  # Primary variable

    if g_h.dim == 2:  # Adapt interface discretization to neighboring subdomains
        interface_discr = pp.RobinCoupling('flow', matrix_discr, fracture_discr)
    else:  # Omega_l is an intersection point
        interface_discr = pp.RobinCoupling('flow', fracture_discr, fracture_discr)

    # Define coupling
    d[pp.COUPLING_DISCRETIZATION] = { 'interface_flux':  # Identifier of this term
        {g_h: ('p', 'diffusion'),
         g_l: ('p', 'diffusion'),
         e: ('mortar_flux', interface_discr)}}
```

Fig. 10 Code snippet of the discretization assignment for the combination of RT0-P0 and TPFA. The code can be used as a partial replacement of the green section in Fig. 7. Note that the parameter definition is not included in the snippet

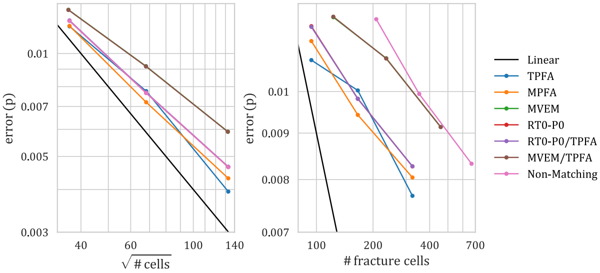


Fig. 11 Left: Convergence of the pressure unknown for the matrix subdomain for the simulations reported in Section 5.1. Right: Convergence for the pressure unknown for the fracture subdomains

deal with coupled problems and time-dependent mixed boundary conditions. The original problem consists of an isotropic poroelastic slab of width $2a$ and height $2b$ sandwiched by two rigid plates (Fig. 12). Initially, two compressive constant loads of intensity $2F$ are applied to the slab at $y = \pm b$. At $x = \pm a$, fluid is free to drain, and edges are stress free. Gravity contributions are neglected.

The problem is modeled using the quasi-static Biot equations, as presented in Section 3.3. Exploiting the symmetry of the problem, we focus on the positive quarter domain Ω', rather than the full domain Ω, see Fig. 12 for an illustration and for boundary conditions. Note that the vertical displacement at the top of the domain is time-dependent and given by the exact solution, see [76].

The simulation parameters were taken from [77], see also the supplementary material for details. The coupled problem is discretized in space using MPSA and MPFA for the mechanics and flow, respectively. For the time discretization, we use implicit Euler. The computational grid is unstructured and composed of 622 triangular elements. The results are shown in Fig. 13 in terms of dimensionless quantities and are in good agreement with [77] for both pressure and displacement.

In Fig. 14, we show a code snippet illustrating the assembly of a generic poroelastic problem using MPSA/MPFA in PorePy. One primary variable for each subproblem must be specified, namely displacement for the mechanics (variable 0) and pressure for the flow (variable 1). There are five terms (plus one stabilization term) involved in the discretization of the Biot equations. We label them with subscripts kl identifying the impact on variable k from variable l. The numbering also corresponds to the placement in the 2 × 2 block discretization matrix, with the first row representing the momentum balance and the second row the mass balance.

The Mpsa class is used to obtain the divergence of the stress (term_00), which corresponds to the first diagonal block. For the second diagonal block, term_11_0 and term_11_1 refer to the discretization of the fluid accumulation and fluid flux (after applying implicit Euler) obtained using the classes ImplicitMassMatrix and ImplicitMpfa, respectively. In addition, term_11_2 is a stabilization term arising naturally from the discretization process [61].

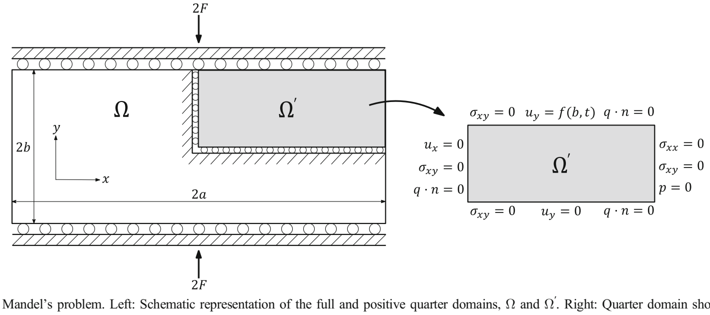


Fig. 12 Mandel's problem. Left: Schematic representation of the full and positive quarter domains, Ω and Ω'. Right: Quarter domain showing the boundary conditions

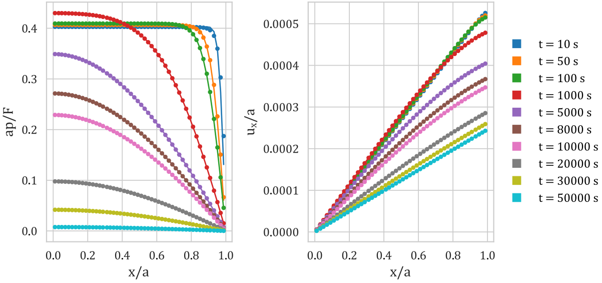


Fig. 13 Analytical (solid lines) and MPSA/MPFA (dots) solutions to Mandel's problem. The dimensionless profiles for the pressure (left) and the horizontal displacement (right) are shown for several times

Lastly, term_01 and term_10 are the off-diagonal coupling blocks representing respectively the terms involving the pressure gradient (obtained with GradP) and the divergence of the displacement field (obtained with DivU).

5.3 Sneddon's problem of fracture deformation

In this example, a square domain with a single fracture located in the middle is considered. The fracture forms an angle β with the horizontal direction (see Fig. 15) and is subjected to a constant pressure p_0, which can be interpreted as a pair of normal forces acting on either side of the fracture. An analytical solution for the relative normal displacement along the fracture was derived by Sneddon [78] for an infinite domain, and has the following form:

$$[\![u_j]\!]_n(d_f) = \frac{(1-\nu)p_0 L}{G}\sqrt{1-\frac{d_f^2}{\left(\frac{L}{2}\right)^2}}, \tag{5.1}$$

```
## Primary variables for poroelasticity problems
v_0 = 'u'  # displacement
v_1 = 'p'  # pressure

## Name of the five terms of the equation + additional stabilization term
term_00 = 'stress_divergence'       # div symmetric grad u
term_11_0 = 'fluid_mass'            # d/dt beta p
term_11_1 = 'fluid_flux'            # div (rho g - K grad p)
term_11_2 = 'stabilization'
term_01 = 'pressure_gradient'       # alpha grad p
term_10 = 'displacement_divergence' # d/dt alpha div u

## Store in the data dictionary and specify discretization objects
d[pp.PRIMARY_VARIABLES] = {v_0: {'cells': g.dim}, v_1: {'cells': 1}}
d[pp.DISCRETIZATION] = {
    # Momentum balance equation
    v_0: {term_00: pp.Mpsa(kw_m)},
    # Mass conservation equation
    v_1: {
        term_11_0: pp.ImplicitMassMatrix(kw_f, v_1),
        term_11_1: pp.ImplicitMpfa(kw_f),
        term_11_2: pp.BiotStabilization(kw_f, v_1),
    },
    # Pressure contribution to the momentum balance equation
    v_0 + '_' + v_1: {term_01: pp.GradP(kw_m)},
    # Displacement contribution to the mass conservation equation
    v_1 + '_' + v_0: {term_10: pp.DivU(kw_m, kw_f, v_0)},
}
```

Fig. 14 Code snippet illustrating the terms involved in the assembly of a poroelastic problem using MPSA/MPFA in PorePy. The snippet highlights assignment of discretizations for multiphysics problems within a subdomain

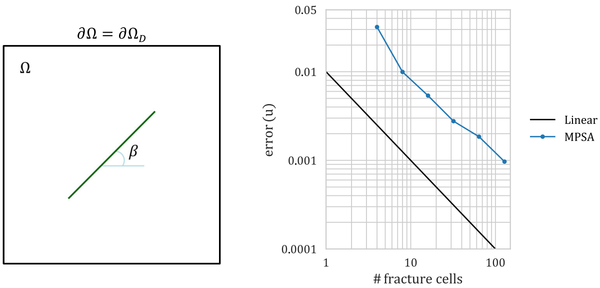


Fig. 15 Setup and convergence of Sneddon's problem. Left: Schematic representation of the domain. Right: Average convergence behavior of the relative normal displacement along the fracture. Each dot corresponds to the average of 140 simulations

where ν and G are the Poisson's ratio and shear modulus, respectively, L is the fracture length, and d_f denotes the distance from the center of the fracture.

In our calculations, the condition of infinite domain is replaced with a Dirichlet boundary, where the prescribed displacement is set equal to the analytical solution calculated using the procedure illustrated in [79]. The accuracy of the numerical solution is very sensitive to the discretization, specifically the cell configuration at the fracture tips [46]. To reduce the dependency on specific grid realizations, the values of the numerical solution reported in Fig. 16 are the average of a group of $20 \times 7 = 140$ computations per level of grid resolution, with 7 different fracture angles β in the range 0°–30° and 20 grid realizations per fracture. With six levels of grid refinement, the full study contains $20 \times 7 \times 6 = 840$ simulations. Figure 16 summarizes the results in the form of the error in relative normal displacement between the analytical solution (5.1) and the numerical solution as a function of the fracture resolution, i.e., number of fracture elements. The method provides first-order convergence on average.

Finally, the code snippet in Fig. 16 indicates the key parts of the variable and discretization assignment for the contact mechanics problem. The classes to note are ColoumbContact, which represents Eqs. (3.14) and (3.15), and the interface discretization PrimalContactCoupling, see also the discussion in Section 4.2.3.

6 Applications: multiphysics simulations

Having established the accuracy of PorePy for central test cases that involve mixed-dimensional geometries, we proceed to present two multiphysics cases of high application relevance: A non-linearly coupled flow and transport problem, and fracture reactivation caused by fluid injection. The motivation for the simulations is to illustrate further capabilities of the modeling framework and its PorePy implementation, including simulations on complex 3d fracture networks, automatic differentiation applied to non-linear problems, non-matching grids, and simulation of fracture deformation in a poroelastic setting.

6.1 Fully coupled flow and transport

We consider the injection of a more viscous fluid into a domain initially filled with a less viscous fluid. The two fluids are miscible and have equal densities; thus, they can be modeled as two components in a single-phase system, as described in Section 3.2. The viscosity of the mixture of fluids given by $\mu_i(c_i) = \exp(c_i)$, for the mass fraction $c_i \in [0, 1]$, which is 0 if only the less viscous fluid is present and 1 if only the more viscous fluid is present. In the parameter regime studied in this example, the transport in the fractures is advection dominated, while the transport in the rock matrix is dominated by diffusion, see the supplementary material for details.

The time derivative is approximated using an implicit Euler method, which gives a fully implicit scheme for the primary variables pressure and mass fraction. The spatial terms are discretized by a finite volume method, with simple upstream for advective terms, and TPFA for fluxes and diffusive terms. We apply forward automatic differentiation implemented in PorePy to obtain the Jacobian of the global system of equations, which is then used in a standard Newton method to solve the non-linear problem. The convergence criterion is given by the maximum norm of the residual vector with a tolerance 10^{-9}.

The mixed-dimensional domain considered in this example consists of one 3d domain, 15 2d fracture domains, 62 1d domains, and 9 0d domains. On this geometry, two computational grids are constructed: The first has matching grids in all dimensions, with in total 20,812 cells, out of which 16,766 are 3d cells and 3,850 are 2d fracture cells. The second mixed-dimensional grid has a 3d grid identical to the first grid, whereas the lower-dimensional objects are assigned refined grids with in total 13,839 2d fracture cells; thus, the 3d-2d interfaces have non-matching grids. The combination of the

```
Nd = gb.dim_max()  # Get ambient dimension
# For the 2d domain we solve linear elasticity with MPSA
mpsa = pp.Mpsa('mechanics')

# The assembler expects that all variables are assigned a discretization. Create a
# void discretization object; the friction discretization is set via the interface law
empty_discr = pp.VoidDiscretization('friction', Nd)

# Define discretization parameters
for g, d in gb:
    if g.dim == Nd:  # Omega_i is the matrix
        d[pp.PRIMARY_VARIABLES] = {'u': {'cells': Nd}}
        d[pp.DISCRETIZATION] = {'u': {'mpsa': mpsa}}
    else:  # Omega_i is a fracture
        d[pp.PRIMARY_VARIABLES] = {'contact': {'cells': Nd}}
        d[pp.DISCRETIZATION] = {'contact': {'empty': empty_discr}}

# For the 1d domain we define a contact condition
coloumb = pp.ColoumbContact('friction', Nd)
# Define a contact condition on the mortar grid
contact = pp.PrimalContactCoupling('friction', mpsa, coloumb)

# Loop over all interfaces (e) and their associated data (d)
for e, d in gb.edges():
    g_l, g_h = gb.nodes_of_edge(e)

    d[pp.PRIMARY_VARIABLES] = {'interface_u': {'cells': Nd}}
    d[pp.COUPLING_DISCRETIZATION] = {'friction': {
        g_h: ('u', 'mpsa'),
        g_l: ('contact', 'empty'),
        (g_h, g_l): ('interface_u', contact)}}
```

Fig. 16 Code snippet that illustrates variable and discretization assignment for Sneddon's problem, discretized using the contact mechanics functionality in PorePy. The code can be used as a partial replacement of the green section in Fig. 7

non-linearity and the non-matching grids provides a challenging test for the robustness of the PorePy implementation of subdomain couplings and provides an illustration of the framework's flexibility.

Figure 17 shows the average mass fraction profile in the fractures for the two grids. There are no significant differences between the two cases, indicating the stability of the implementation of the non-matching case. Figure 18 shows a snapshot of the mass fraction in the fractures and the rock matrix at time $t = 20$. The diffusive front in the rock matrix has only moved a few grid cells at the break-through; however, due to the diffusion and advection from the fractures to the rock matrix, the mass fraction has increased in considerable parts of the rock matrix. We observe no irregularities for the solution produced on the non-matching grid in this case, suggesting PorePy's ability to deal with non-standard grid couplings also for challenging physical regimes.

6.2 Poroelasticity and fracture deformation

The final example aims at demonstrating the modeling framework's and PorePy's applicability to non-standard combinations of physical processes in different domains and thereby its potential for method development and prototyping. With the critical events taking place on individual fractures as a result of processes in the rock matrix, it also serves as an example of the importance of incorporating dynamics of both the matrix and explicitly represented fractures, as done in DFM models.

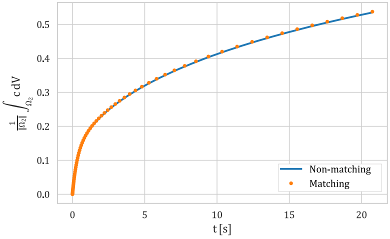


Fig. 17 Fully coupled flow and transport: Comparison of average mass fraction in the fracture network for a simulation with matching grids and a simulation with non-matching grids

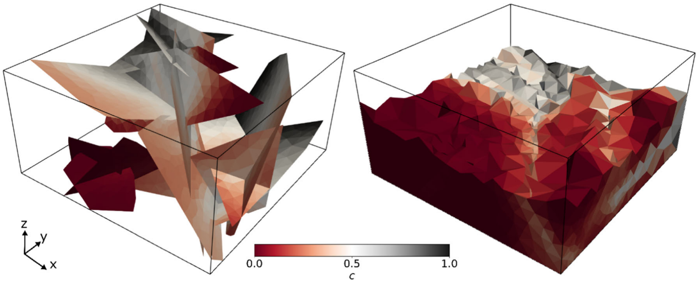


Fig. 18 Fully coupled flow and transport: Mass fraction in the fractures (left) and in the rock matrix (right) for the coupled flow and transport problem given in Section 3.2 at the end time of the simulation ($t = 20$). In the right figure, the rock matrix domain is cropped, and the fractures removed to reveal the mass fraction inside the domain. The black lines indicate the domain boundary. Non-matching grids are used with the fracture grids being much finer than the grid in the rock matrix

Specifically, we consider the model equations for coupled poroelasticity and fracture deformation presented in Section 3.3. The poroelastic deformation of the host rock is discretized with MPSA, while the fluid flow in the fractures is discretized with MPFA. The discretization of the contact mechanics follows the structure outlined in Section 4.2.3, and temporal discretization is performed using implicit Euler.

We consider a reservoir of idealized geometry containing three non-intersecting fractures numbered from 1 through 3, whereof the first contains an injection well (see Fig. 19). On this geometry, we solve the governing equations presented in Section 3.3. We impose injection over a 25-day period and an anisotropic background stress regime, producing a scenario well suited to demonstrate different fracture dynamics. We investigate the dynamics both during the injection phase and during the subsequent 25-day relaxation phase, at the end of which the pressure has almost reached equilibrium once more. The full set of parameters may be found in the supplementary material.

The dynamics on the fractures throughout the simulation are summarized in Fig. 19, while the spatial distribution of the fracture displacement jumps at the end of the injection phase is shown in Fig. 20. The figures show how the simulation captures the complex dynamics both during and after injection, and thus highlight how the explicit fracture representation allows for detailed studies of fracture deformation.

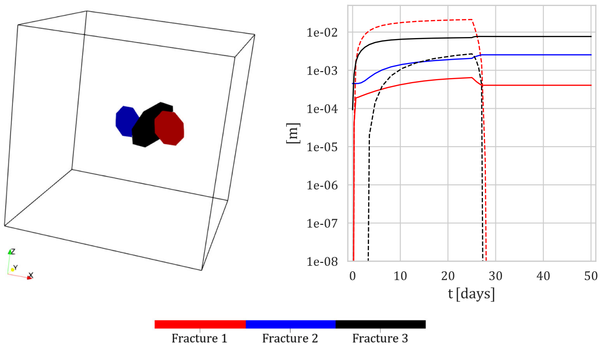


Fig. 19 Left: Domain geometry with numbering of the three fractures. Fluid is injected in fracture 1 during the first 25 days, after which the well is shut. Right: L^2-norm normalized by fracture area of the normal (dashed lines) and tangential (solid lines) displacement jumps for each fracture

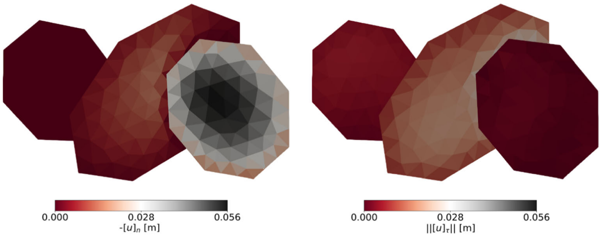


Fig. 20 Normal and tangential displacements jump on the fractures at the end of the injection phase to the left and right, respectively. The orientation of the fracture network corresponds to that in Fig. 12, with the injection fracture to the right

7 Conclusions

The complexity in modeling and simulation of multiphysics processes in fractured porous media, combined with a strong current research focus and corresponding developments, calls for flexible simulation tools that facilitate rapid prototyping of models and discretization methods. This paper presents design principles for such software together with their implementation in the open-source simulation tool PorePy. The combined framework for modeling and simulation is based on the discrete fracture matrix model, where fractures and their intersections are represented as separate lower-dimensional geometric objects. The framework facilitates flexibility for multiphysics dynamics and reuse of existing code written for non-fractured domains; hence, it is well suited for extending other software packages to mixed-dimensional problems.

The open-source software PorePy demonstrates the capabilities of the suggested framework: It provides automatic gridding of complex fracture networks in two and three dimensions, and contains implemented numerical methods for flow, transport, poroelastic deformation of the rock, and fracture deformation modeled by contact mechanics. The implementation performs well for benchmark problems in flow, poroelastic deformation, and fracture deformation. Furthermore, multiphysics simulations of fully coupled flow and non-linear transport and of fracture deformation under poroelastic deformation of a domain demonstrate the versatility of the software.

Acknowledgments The authors thank two anonymous reviewers for the comments and suggestions that helped to improve the quality of the paper.

Funding Open Access funding provided by University of Bergen. This work has been funded in part by Norwegian Research Council grant 250223, 244129/E20, 267908/E20, and 274883, and by a VISTA Scholarship from the Norwegian Academy of Science and Letters.

References

1. Berkowitz, B.: Characterizing flow and transport in fractured geological media: a review. Adv. Water Resour. **25**(8–12), 861–884 (2002). https://doi.org/10.1016/S0309-1708(02)00042-8
2. Martin, V., Jaffré, J., Roberts, J.E.: Modeling fractures and barriers as interfaces for flow in porous media. SIAM J. Sci. Comput. **26**(5), 1667–1691 (2005). https://doi.org/10.1137/S1064827503429363
3. Barton, N., Bandis, S., Bakhtar, K.: Strength, deformation and conductivity coupling of rock joints. Int. J. Rock Mech. Min. Sci. Geomech. Abstr. **22**(3), 121–140 (1985). https://doi.org/10.1016/0148-9062(85)93227-9
4. Frih, N., Roberts, J.E., Saada, A.: Modeling fractures as interfaces: a model for Forchheimer fractures. Comput. Geosci. **12**(1), 91–104 (2008). https://doi.org/10.1007/s10596-007-9062-x
5. Rutqvist, J., Wu, Y.-S., Tsang, C.-F., Bodvarsson, G.: A modeling approach for analysis of coupled multiphase fluid flow, heat transfer, and deformation in fractured porous rock. Int. J. Rock Mech. Min. Sci. **39**(4), 429–442 (2002). https://doi.org/10.1016/S1365-1609(02)00022-9
6. Burnell, J., et al.: Geothermal supermodels: the next generation of integrated geophysical, chemical and flow simulation modelling tools. Proc World Geotherm. Congr. 7 (2015)
7. Pruess, K.: TOUGH2: a general numerical simulator for multiphase fluid and heat flow. Report LBL-29400 (1991)
8. Hammond, G.E., Lichtner, P.C., Mills, R.T.: Evaluating the performance of parallel subsurface simulators: an illustrative example with PFLOTRAN: evaluating the parallel performance of Pflotran. Water Resour. Res. **50**(1), 208–228 (2014). https://doi.org/10.1002/2012WR013483
9. Barenblatt, G.I., Zheltov, I.P., Kochina, I.N.: Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]. J. Appl. Math. Mech. **24**(5), 1286–1303 (1960). https://doi.org/10.1016/0021-8928(60)90107-6

10. Arbogast, T., Douglas Jr., J., Hornung, U.: Derivation of the double porosity model of single phase flow via homogenization theory. SIAM J. Math. Anal. **21**(4), 823–836 (1990). https://doi.org/10.1137/0521046
11. Lemonnier, P., Bourbiaux, B.: Simulation of naturally fractured reservoirs. State of the art: part 1 – physical mechanisms and simulator formulation. Oil Gas Sci. Technol. Rev. L'Institut Fr. Pétrole. **65**(2), 239–262 (2010). https://doi.org/10.2516/ogst/2009066
12. Lemonnier, P., Bourbiaux, B.: Simulation of naturally fractured reservoirs. State of the art: part 2 – matrix-fracture transfers and typical features of numerical studies. Oil Gas Sci. Technol. – Rev. L'Institut Fr. Pétrole. **65**(2), 263–286 (2010). https://doi.org/10.2516/ogst/2009067
13. Hyman, J.D., Karra, S., Makedonska, N., Gable, C.W., Painter, S.L., Viswanathan, H.S.: dfnWorks: a discrete fracture network framework for modeling subsurface flow and transport. Comput. Geosci. **84**, 10–19 (2015). https://doi.org/10.1016/j.cageo.2015.08.001
14. Erhel, J., de Dreuzy, J.-R., Poirriez, B.: Flow simulation in three-dimensional discrete fracture networks. SIAM J. Sci. Comput. **31**(4), 2688–2705 (2009). https://doi.org/10.1137/080729244
15. Berrone, S., Pieraccini, S., Scialò, S.: On simulations of discrete fracture network flows with an optimization-based extended finite element method. SIAM J. Sci. Comput. **35**(2), A908–A935 (2013). https://doi.org/10.1137/120882883
16. Berre, I., Doster, F., Keilegavlen, E.: Flow in fractured porous media: a review of conceptual models and discretization approaches. Transp. Porous Media. **130**, 215–236 (2018). https://doi.org/10.1007/s11242-018-1171-6
17. Noorishad, J., Mehran, M.: An upstream finite element method for solution of transient transport equation in fractured porous media. Water Resour. Res. **18**(3), 588–596 (1982). https://doi.org/10.1029/WR018i003p00588
18. Baca, R.G., Arnett, R.C., Langford, D.W.: Modelling fluid flow in fractured-porous rock masses by finite-element techniques. Int. J. Numer. Methods Fluids. **4**(4), 337–348 (1984). https://doi.org/10.1002/fld.1650040404
19. Reichenberger, V., Jakobs, H., Bastian, P., Helmig, R.: A mixed-dimensional finite volume method for two-phase flow in fractured porous media. Adv. Water Resour. **29**(7), 1020–1036 (2006). https://doi.org/10.1016/j.advwatres.2005.09.001
20. Li, L., Lee, S.H.: Efficient field-scale simulation of black oil in a naturally fractured reservoir through discrete fracture networks and homogenized media. SPE Reserv. Eval. Eng. **11**(04), 750–758 (2008). https://doi.org/10.2118/103901-PA
21. Fumagalli, A., Scotti, A.: A reduced model for flow and transport in fractured porous media with non-matching grids. In: Cangiani, A., Davidchack, R.L., Georgoulis, E., Gorban, A.N., Levesley, J., Tretyakov, M.V. (eds.) Numerical Mathematics and Advanced Applications 2011, pp. 499–507. Springer, Berlin (2013)
22. Flemisch, B., Fumagalli, A., Scotti, A.: A review of the XFEM-based approximation of flow in fractured porous media. In: Ventura, G., Benvenuti, E. (eds.) Advances in Discretization Methods, vol. 12, pp. 47–76. Springer International Publishing, Cham (2016)
23. Schwenck, N., Flemisch, B., Helmig, R., Wohlmuth, B.I.: Dimensionally reduced flow models in fractured porous media: crossings and boundaries. Comput. Geosci. **19**(6), 1219–1230 (2015). https://doi.org/10.1007/s10596-015-9536-1
24. Jiang, J., Younis, R.M.: An improved projection-based embedded discrete fracture model (pEDFM) for multiphase flow in fractured reservoirs. Adv. Water Resour. **109**, 267–289 (2017). https://doi.org/10.1016/j.advwatres.2017.09.017
25. Flemisch, B., Darcis, M., Erbertseder, K., Faigle, B., Lauser, A., Mosthaf, K., Müthing, S., Nuske, P., Tatomir, A., Wolff, M., Helmig, R.: DuMux: DUNE for multi-{phase,component,scale, physics,…} flow and transport in porous media. Adv. Water Resour. **34**(9), 1102–1112 (2011). https://doi.org/10.1016/j.advwatres.2011.03.007
26. Matthäi, S.K., Geiger, S., Roberts, S.G., Paluszny, A., Belayneh, M., Burri, A., Mezentsev, A., Lu, H., Coumou, D., Driesner, T., Heinrich, C.A.: Numerical simulation of multi-phase fluid flow in structurally complex reservoirs. Geol. Soc. Lond. Spec. Publ. **292**(1), 405–429 (2007). https://doi.org/10.1144/SP292.22
27. Gaston, D., Newman, C., Hansen, G., Lebrun-Grandié, D.: MOOSE: a parallel computational framework for coupled systems of nonlinear equations. Nucl. Eng. Des. **239**(10), 1768–1778 (2009). https://doi.org/10.1016/j.nucengdes.2009.05.021
28. Breede, K., Dzebisashvili, K., Liu, X., Falcone, G.: A systematic review of enhanced (or engineered) geothermal systems: past, present and future. Geotherm. Energy. **1**(1), 4 (2013). https://doi.org/10.1186/2195-9706-1-4
29. Wang, W., Kolditz, O.: Object-oriented finite element analysis of thermo-hydro-mechanical (THM) problems in porous media. Int. J. Numer. Methods Eng. **69**(1), 162–201 (2007). https://doi.org/10.1002/nme.1770
30. Březina, J., Stebel, J.: Analysis of model error for a continuum-fracture model of porous media flow. In: Kozubek, T., Blaheta, R., Šístek, J., Rozložník, M., Čermák, M. (eds.) High Performance Computing in Science and Engineering, vol. 9611, pp. 152–160. Springer International Publishing, Cham (2016)
31. Lie, K.-A.: An Introduction to Reservoir Simulation Using MATLAB/GNU Octave: User Guide for the MATLAB Reservoir Simulation Toolbox (MRST), 1st edn. Cambridge University Press (2019)
32. Lie, K.-A., Krogstad, S., Ligaarden, I.S., Natvig, J.R., Nilsen, H.M., Skaflestad, B.: Open-source MATLAB implementation of consistent discretisations on complex grids. Comput. Geosci. **16**(2), 297–322 (2012). https://doi.org/10.1007/s10596-011-9244-4
33. Alnæs, M., et al.: The FEniCS Project Version 1.5. Arch. Numer. Softw. **3**, (2015). https://doi.org/10.11588/ans.2015.100.20553
34. Blatt, M., et al.: The distributed and unified numerics environment, Version 2.4. Arch. Numer. Softw. **4**, (2016). https://doi.org/10.11588/ans.2016.100.26526
35. Rathgeber, F., Ham, D.A., Mitchell, L., Lange, M., Luporini, F., Mcrae, A.T.T., Bercea, G.T., Markall, G.R., Kelly, P.H.J.: Firedrake: automating the finite element method by composing abstractions. ACM Trans. Math. Softw. **43**(3), 1–27 (2016). https://doi.org/10.1145/2998441
36. Boon, W.M., Nordbotten, J.M., Vatne, J.E.: Functional analysis and exterior calculus on mixed-dimensional geometries. Ann. Mat. (2020). https://doi.org/10.1007/s10231-020-01013-1
37. Boon, W.M., Nordbotten, J.M.: Stable mixed finite elements for linear elasticity with thin inclusions. arXiv. **1903.01757**, (2019)
38. Nordbotten, J.M., Boon, W.M., Fumagalli, A., Keilegavlen, E.: Unified approach to discretization of flow in fractured porous media. Comput. Geosci. **23**(2), 225–237 (2019). https://doi.org/10.1007/s10596-018-9778-9
39. Geuzaine, C., Remacle, J.-F.: Gmsh: a 3-D finite element mesh generator with built-in pre- and post-processing facilities. Int. J. Numer. Methods Eng. **79**(11), 1309–1331 (2009). https://doi.org/10.1002/nme.2579
40. Karimi-Fard, M.: An efficient discrete-fracture model applicable for general-purpose reservoir simulators. SPE J. **9**(2), (2004). https://doi.org/10.2118/88812-PA
41. Hui, M.-H., Mallison, B., Lim, K.-T.: An innovative workflow to model fractures in a giant carbonate reservoir. Proc. Int. Pet. Tech. Conf. 15 (2008)
42. Berre, I., et al.: Verification benchmarks for single-phase flow in three-dimensional fractured porous media. arXiv. **2002.07005**, (2020)
43. Boon, W.M., Nordbotten, J.M., Yotov, I.: Robust discretization of flow in fractured porous media. SIAM J. Numer. Anal. **56**(4), 2203–2233 (2018). https://doi.org/10.1137/17M1139102

44. Quarteroni, A., Valli, A.: Numerical approximation of partial differential equations, 2nd edn. Springer, Berlin (1997)
45. Garipov, T.T., Karimi-Fard, M., Tchelepi, H.A.: Discrete fracture model for coupled flow and geomechanics. Comput. Geosci. **20**(1), 149–160 (2016). https://doi.org/10.1007/s10596-015-9554-z
46. Ucar, E., Keilegavlen, E., Berre, I., Nordbotten, J.M.: A finite-volume discretization for deformation of fractured media. Comput. Geosci. **22**(4), 993–1007 (2018). https://doi.org/10.1007/s10596-018-9734-8
47. McClure, M.W., Horne, R.N.: An investigation of stimulation mechanisms in Enhanced Geothermal Systems. Int. J. Rock Mech. Min. Sci. **72**, 242–260 (2014). https://doi.org/10.1016/j.ijrmms.2014.07.011
48. Coussy, O.: Poromechanics. Chichester, Wiley (2003)
49. Berge, R.L., Berre, I., Keilegavlen, E., Nordbotten, J.M., Wohlmuth, B.: Finite volume discretization for poroelastic media with fractures modeled by contact mechanics. Int. J. Numer. Methods Eng. **121**(4), 644–663 (2020). https://doi.org/10.1002/nme.6238
50. Dong, S., Zeng, L., Dowd, P., Xu, C., Cao, H.: A fast method for fracture intersection detection in discrete fracture networks. Comput. Geotech. **98**, 205–216 (2018). https://doi.org/10.1016/j.compgeo.2018.02.005
51. Mallison, B.T., Hui, M.H., Narr, W.: Practical gridding algorithms for discrete fracture modeling workflows. presented at the 12th European Conference on the Mathematics of Oil Recovery, Oxford, UK (2010). https://doi.org/10.3997/2214-4609.20144950
52. Holm, R., Kaufmann, R., Heimsund, B.-O., Øian, E., Espedal, M.S.: Meshing of domains with complex internal geometries. Numer. Linear Algebra Appl. **13**(9), 717–731 (2006). https://doi.org/10.1002/nla.505
53. Berge, R.L., Klemetsdal, Ø.S., Lie, K.-A.: Unstructured Voronoi grids conforming to lower dimensional objects. Comput. Geosci. **23**(1), 169–188 (2019). https://doi.org/10.1007/s10596-018-9790-0
54. Shewchuk, J.R.: Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Applied Computational Geometry: Towards Geometric Engineering, vol. 1148, pp. 203–222 (1996)
55. Si, H.: TetGen, a Delaunay-based quality tetrahedral mesh generator. ACM Trans. Math. Softw. **41**(2), 1–36 (2015). https://doi.org/10.1145/2629697
56. Boffi, D., Brezzi, F., Fortin, M.: Mixed finite element methods and applications. Springer, Berlin (2013)
57. da Veiga, L.B., Brezzi, F., Marini, L.D., Russo, A.: Mixed virtual element methods for general second order elliptic problems on polygonal meshes. ESAIM Math. Model. Numer. Anal. **50**(3), 727–747 (2016). https://doi.org/10.1051/m2an/2015067
58. da Veiga, L.B., Brezzi, F., Marini, L.D., Russo, A.: H(div) and H(curl) -conforming virtual element methods. Numer. Math. **133**(2), 303–332 (2016). https://doi.org/10.1007/s00211-015-0746-1
59. Nordbotten, J.M.: Convergence of a cell-centered finite volume discretization for linear elasticity. SIAM J. Numer. Anal. **53**(6), 2605–2625 (2015). https://doi.org/10.1137/140972792
60. Keilegavlen, E., Nordbotten, J.M.: Finite volume methods for elasticity with weak symmetry. Int. J. Numer. Methods Eng. **112**(8), 939–962 (2017). https://doi.org/10.1002/nme.5538
61. Nordbotten, J.M.: Stable cell-centered finite volume discretization for Biot equations. SIAM J. Numer. Anal. **54**(2), 942–968 (2016). https://doi.org/10.1137/15M1014280
62. Nordbotten, J.M., Keilegavlen, E.: An introduction to multi-point flux (MPFA) and stress (MPSA) finite volume methods for thermo-poroelasticity. arXiv. **2001.01990**, (2020)
63. Hüeber, S., Stadler, G., Wohlmuth, B.I.: A primal-dual active set algorithm for three-dimensional contact problems with coulomb friction. SIAM J. Sci. Comput. **30**(2), 572–596 (2008). https://doi.org/10.1137/060671061
64. Stefansson, I., Berre, I., Keilegavlen, E.: A fully coupled numerical model of thermo-hydro-mechanical processes and fracture contact mechanics in porous media. **arXiv:2008.06289**, (2020)
65. Berge, R.L., Berre, I., Keilegavlen, E., Nordbotten, J.M.: Viscous fingering in fractured porous media. **arXiv:1906.10472**, (2019)
66. Budisa, A., Boon, W., Hu, X.: Mixed-dimensional auxiliary space preconditioners. **arXiv:1910.04704**, (2019)
67. Budiša, A., Hu, X.: Block preconditioners for mixed-dimensional discretization of flow in fractured porous media. Comput. Geosci. (2020). https://doi.org/10.1007/s10596-020-09984-z
68. Ahrens, J., Geveci, B., Law, C.: ParaView: an end-user tool for large data visualization
69. Fumagalli, A., Keilegavlen, E., Scialò, S.: Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations. J. Comput. Phys. **376**, 694–712 (2019). https://doi.org/10.1016/j.jcp.2018.09.048
70. Fumagalli, A., Keilegavlen, E.: Dual virtual element methods for discrete fracture matrix models. Oil Gas Sci. Technol. – Rev. D'IFP Energ. Nouv. **74**, 41 (2019). https://doi.org/10.2516/ogst/2019008
71. Stefansson, I., Berre, I., Keilegavlen, E.: Finite-volume discretisations for flow in fractured porous media. Transp. Porous Media. **124**(2), 439–462 (2018). https://doi.org/10.1007/s11242-018-1077-3
72. PorePy implementation with runscripts. https://doi.org/10.5281/zenodo.3374624. (2019)
73. Flemisch, B., Berre, I., Boon, W., Fumagalli, A., Schwenck, N., Scotti, A., Stefansson, I., Tatomir, A.: Benchmarks for single-phase flow in fractured porous media. Adv. Water Resour. **111**, 239–258 (2018). https://doi.org/10.1016/j.advwatres.2017.10.036
74. Mandel, J.: Consolidation des sols (étude mathématique). Geotechnique. **3**(7), 287–299 (1953)
75. Abousleiman, Y., Cheng, A.-D., Cui, L., Detournay, E., Rogiers, J.-C.: Mandel's problem revisited. Geotechnique. **46**(2), 187–195 (1996)
76. Cheng, A.H.-D., Detournay, E.: A direct boundary element method for plane strain poroelasticity. Int. J. Numer. Anal. Methods Geomech. **12**(5), 551–572 (1988). https://doi.org/10.1002/nag.1610120508
77. Mikelić, A., Wang, B., Wheeler, M.F.: Numerical convergence study of iterative coupling for coupled flow and geomechanics. Comput. Geosci. **18**(3–4), 325–341 (2014). https://doi.org/10.1007/s10596-013-9393-8
78. Sneddon, I.N.: Fourier Transforms. Dover Publications, New York (1995)
79. Crouch, S.L., Starfield, A.M.: Boundary Element Methods in Solid Mechanics: with Applications in Rock Mechanics and Geological Engineering. Allen & Unwin, London (1983)

Paper C

A posteriori error estimates for hierarchical mixed-dimensional elliptic equations

Varela J., Ahmed E., Keilegavlen E., Nordbotten J. M., Radu, F. A.

Under review in the Journal Of Numerical Mathematics.

A posteriori error estimates for hierarchical mixed-dimensional elliptic equations

Jhabriel Varela[*,†] Elyes Ahmed[‡] Eirik Keilegavlen[†]
Jan Martin Nordbotten[†] Florin Adrian Radu[†]

April 21, 2022

Abstract

Mixed-dimensional elliptic equations exhibiting a hierarchical structure are commonly used to model problems with high aspect ratio inclusions, such as flow in fractured porous media. We derive general abstract estimates based on the theory of functional *a posteriori* error estimates, for which guaranteed upper bounds for the primal and dual variables and two-sided bounds for the primal-dual pair are obtained. We improve on the abstract results obtained with the functional approach by proposing four different ways of estimating the residual errors based on the extent the approximate solution has conservation properties, i.e.: (1) no conservation, (2) subdomain conservation, (3) grid-level conservation, and (4) exact conservation. This treatment results in sharper and fully computable estimates when mass is conserved either at the grid level or exactly, with a comparable structure to those obtained from grid-based *a posteriori* techniques. We demonstrate the practical effectiveness of our theoretical results through numerical experiments using four different discretization methods for synthetic problems and applications based on benchmarks of flow in fractured porous media.

Keywords: mixed-dimensional geometry, functional a posteriori error estimates, fractured porous media

Classification: 65N15, 76S05, 35Q86

[*] Corresponding author e-mail: jhabriel.varela@uib.no.

[†] Center for Modeling of Coupled Subsurface Dynamics, University of Bergen, P.O. Box 7800, N-5020 Bergen, Norway.

[‡] SINTEF Digital, Mathematics and Cybernetics, P.O. Box 124 Blindern, N-0314 Oslo, Norway.

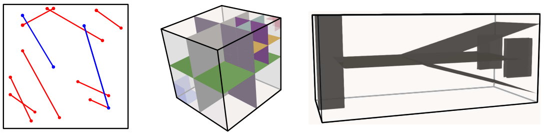


Figure 1: Example geometries falling within the context of hierarchical mixed-dimensional geometries studied herein. Left figure corresponds to a 2d benchmark problem [13] while the two remaining correspond to 3d benchmark problems [14].

1 Introduction

Mixed-dimensional partial differential equations (mD-PDEs) arise when partial differential equations interact on domains of different topological dimensions [1]. Prototypical examples include models of thin inclusions in elastic materials [2, 3, 4], blood flow in human vasculature [5, 6, 7], root water uptake systems [8], and flow in fractured porous media [9, 10, 11]. The latter example has an appealing mathematical structure, in that the model equations allow for a hierarchical representation where each subdomain (matrix, fractures, fracture intersections, and intersection points) only has direct interaction with subdomains of topological dimension one higher or one lower [12]. Such hierarchical mD-PDEs are the topic of the current paper.

mD-PDEs are intrinsically linked to the underlying geometric representation, which, in a certain sense, generalizes the usual notion of the domain. One can then define sets of suitable functions (and function spaces) on this geometry, and these sets are then naturally interpreted as mixed-dimensional (mD) functions. Exploiting this concept, one can generalize the standard differential operators to mappings between mD functions and thus obtain an mD calculus. The fact that this mD calculus inherits standard properties of calculus, particularly partial integration (relative to suitable inner products), a de Rham complex structure, and a Poincaré-Friedrichs inequality, was recently established using the language of exterior calculus on differential forms [15].

The inherent geometric generality of hierarchical mD-PDEs also demand the same level of abstraction of *a posteriori* error estimation techniques. This requirement makes error estimates of the functional type particularly well-suited for the task [16, 17, 18, 19, 20, 21]. The most attractive feature of this approach is that error estimates are derived using purely functional methods [20]. The bounds are therefore agnostic to the way approximated solutions are obtained in the energy space, and the only undetermined constants arise from Poincaré-type inequalities [22].

However, unlike other types of error estimates [23, 24, 25, 26, 27], this generality makes standard functional estimates of limited applicability to hierarchical elliptic mD-PDEs due to the following reasons: (1) for general fracture networks, the mixed-dimensional Poincaré constant is not easily computable, and (2) since Poincaré constants are proportional to the diameter of the physical domain, residual estimators cannot exhibit superconvergent properties.

To circumvent the aforementioned issues, we exploit the fact that Poincaré-type inequalities imply weighted norms [28, 29], and use spatially-dependent weights to control the residual norms. We show both theoretically and numerically that this treatment leads to sharper estimates when approximations to the exact solution satisfy mass conservation in a given partition of the domain.

In view of the preceding discussion, our aim is therefore to obtain *a posteriori* error estimates for the approximate solution to the mD scalar elliptic equation [12, 30, 15], where the mD Laplace equation for geometries such as those illustrated in Figure 1 is described in detail in Section 3.

We remark that while a broad range of *a posteriori* error techniques are available for mono-dimensional problems, existing error bounds for mD models are far more scarce. Moreover, the ones available, are restricted to specific cases (e.g., in the context of mortar methods [31, 32, 33, 34] and fractured porous media [35, 36, 37]) with far less geometric generality than what we present here. Thus, for practical problems, *a posteriori* error bounds for mD geometries have until now essentially not been available.

The rest of the paper is structured as follows: Section 2 is devoted to introducing the model problem, functional spaces, and variational formulations for the case of a single 1d fracture embedded in a 2d matrix. The section is concluded by providing a first upper bound for the primal variable. In Section 3, we generalize the results from Section 2 to the case of fracture networks and introduce the necessary tools to perform the *a posteriori* analysis in an mD setting. After reviewing necessary tools from functional analysis in Section 4, in Section 5, we provide our main results starting from a generic abstract estimate and then considering specific cases depending upon the degree of accuracy at which residual terms are approximated. In Section 6, we introduce the approximated problem using mixed-finite element methods and thus make the estimates concrete. Sections 7 and 8 deal, respectively, with numerical validations and practical applications of the derived bounds. Finally, in Section 9, we present our concluding remarks.

2 Upper bounds for a single fracture

In this section, we introduce the model problem together with functional spaces and the variational formulations for the case of a single 1d line embedded in a 2d matrix, as illustrated in Figure 2. Furthermore, a first upper bound for the primal variable is derived following the classical functional approach. We remark that the case of a single fracture embedded in a matrix has been analyzed before. For example, [35] and [37] proposed error estimators based on the residual approach, whereas [36] obtained guaranteed a posteriori error estimates using the approach of Vohralík [26].

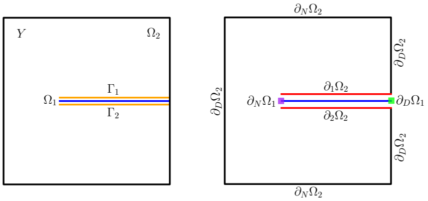


Figure 2: A horizontal 1d fracture embedded in a 2d matrix. Left: Subdomains and interfaces. Right: Boundary conditions. For the fracture, the purple square denotes a no-flux boundary condition, whereas the green square a Dirichlet boundary condition. Note that $\partial_1\Omega_2$, Γ_1, Ω_1, Γ_2, $\partial_2\Omega_2$, all coincide spatially. For illustrative purposes, however, they are placed in different locations.

2.1 The model problem for a single fracture

Before writing the set of equations describing general fracture networks, let us first introduce the governing equations of a simpler configuration; that is, a unit square domain $Y \subset \mathbb{R}^2$ decomposed as a 1d fracture Ω_1 embedded in a 2d matrix Ω_2 as shown in the left Figure 2. Interfaces Γ_1 and Γ_2, at each side of Ω_1, establish the link between Ω_2 and Ω_1. The model presented below is well-established for these problems, and we point the reader to the references for further justification of this system [38, 12, 30].

The strong form of the governing equations in Ω_2 reads

$$\begin{aligned}
\nabla \cdot u_2 &= f_2, && \text{in}\, \Omega_2, && \text{(1a)}\\
u_2 &= -\mathcal{K}_2 \nabla p_2, && \text{in}\, \Omega_2, && \text{(1b)}\\
u_2 \cdot n_2 &= \lambda_1, && \text{on}\, \partial_1\Omega_2, && \text{(1c)}\\
u_2 \cdot n_2 &= \lambda_2, && \text{on}\, \partial_2\Omega_2, && \text{(1d)}\\
u_2 \cdot n_2 &= 0, && \text{on}\, \partial_N\Omega_2, && \text{(1e)}\\
p_2 &= g_{D,2}, && \text{on}\, \partial_D\Omega_2. && \text{(1f)}
\end{aligned}$$

Here, (1a) is the mass conservation equation, u_2 is the matrix velocity, and f_2 an external source. The fluid velocity is given by the standard Darcy's law (1b), where $\mathcal{K}_2$ is the matrix permeability; a bounded, symmetric, and positive-definite 2×2 tensor, and p_2 is the fluid pressure.

Equations (1c) and (1d) require that at each side of the internal boundary of Ω_2, the normal component of u_2 to match the interface (mortar) fluxes λ_1 and λ_2. To fix the direction of the normal vector on internal boundaries, we require n_2 pointing from the higher- to the lower-dimensional subdomain. No flux conditions are prescribed in (1e), where $u_2 \cdot n_2$ represents the outer normal flux across $\partial_N \Omega_2$. Finally, Dirichlet boundary conditions are imposed in (1f), where $g_{D,2}$ is a prescribed function on the Dirichlet boundary.

In the fracture Ω_1, the equations are given by

$$\begin{aligned} \nabla_1 \cdot u_1 - (\lambda_1 + \lambda_2) &= f_1, && \text{in}\, \Omega_1, && \text{(2a)} \\ u_1 &= -\mathcal{K}_1 \nabla_1 \, p_1, && \text{in}\, \Omega_1, && \text{(2b)} \\ u_1 \cdot n_1 &= 0, && \text{on}\, \partial_N \Omega_1, && \text{(2c)} \\ p_1 &= g_{D,1}, && \text{on}\, \partial_D \Omega_1. && \text{(2d)} \end{aligned}$$

In (2a), $\nabla_1 \cdot (\cdot) = \frac{d}{dx}(\cdot) = \nabla_1(\cdot)$ are the divergence and gradient operators acting in the tangent space of Ω_1, u_1 is the tangential fracture velocity, the term in parentheses represents the jump in normal fluxes from the adjacent interfaces Γ_1 and Γ_2 onto Ω_1, and f_1 is an external source.

The tangential velocity u_1 is again expressed via Darcy's law (2b), where in a slight abuse of notation, we use $\mathcal{K}_1$ to refer to the tangential component of the fracture permeability, which is again assumed to be positive and bounded from above. Finally, (2c) and (2d) are the Neumann and Dirichlet boundary conditions, respectively. Again, we use $g_{D,1}$ to denote a prescribed function on the Dirichlet part of the fracture boundary.

To close the system of equations, we must specify a constitutive relationship for the interface fluxes. Here, we use a Darcy-type law [38], where mortar fluxes are linearly related to pressure jumps

$$\begin{aligned} \lambda_1 &= -\kappa_1 \left(p_1 - p_2 \right), && \text{on}\, \Gamma_1, && \text{(3a)} \\ \lambda_2 &= -\kappa_2 \left(p_1 - p_2 \right), && \text{on}\, \Gamma_2, && \text{(3b)} \end{aligned}$$

with κ_1 and κ_2 representing the effective normal permeability on Γ_1 and Γ_2, respectively. We restrict our analysis to the case where κ_1 and κ_2 are non-degenerate. Thus, following [12], we further require the existence of two constants γ_1 and γ_2 such that $0 < \gamma_1 \leq \kappa_j^{-1} \leq \gamma_2 < \infty$ for $j \in \{1, 2\}$.

2.2 Functional spaces and variational formulations

Let us now present the primal weak formulation of the single fracture model from the previous section. To this aim, consider first the energy space with vanishing traces on Dirichlet boundaries

$$H_0^1(\Omega_i) = \{ q_i \in H^1(\Omega_i) : \mathrm{tr}_{\partial_D \Omega_i} \; q_i = 0 \}, \tag{4}$$

and the product spaces

$$H^1(\Omega) = H^1(\Omega_1) \times H^1(\Omega_2) \qquad \text{and} \qquad H_0^1(\Omega) = H_0^1(\Omega_1) \times H_0^1(\Omega_2). \tag{5}$$

Furthermore, let $\langle \cdot, \cdot \rangle_{\Omega_i}$ and $\langle \cdot, \cdot \rangle_{\Gamma_j}$ denote respectively the L^2–inner products on Ω_i and Γ_j, and $\|\cdot\|_{\Omega_i}$ and $\|\cdot\|_{\Gamma_j}$ the relevant L^2–norms. Finally, we denote by $g = [g_1, g_2] \in H^1(\Omega)$ two functions extending the boundary data into the domains, and thus satisfying $\mathrm{tr}_{\partial_D \Omega_i} g_i = g_{D,i}$. We now state the primal weak problem as:

Definition 1 (Primal weak formulation for a single fracture)**.** *Let $p = [p_1, p_2]$ and $g = [g_1, g_2] \in H^1(\Omega)$. Then find $p \in H_0^1(\Omega) + g$ such that*

$$\sum_{i=1}^{2} \langle \mathcal{K}_i \nabla_i p_i, \nabla_i q_i \rangle_{\Omega_i} + \sum_{j=1}^{2} \left\langle \kappa_j \left(p_1 - \mathrm{tr}_{\partial_j \Omega_2} p_2\right), q_1 - \mathrm{tr}_{\partial_j \Omega_2} q_2 \right\rangle_{\Gamma_j} \\ = \sum_{i=1}^{2} \langle f_i, q_i \rangle_{\Omega_i}, \quad \forall\, q = [q_1, q_2] \in H_0^1(\Omega). \tag{6}$$

Refer to Appendix A.1 for the derivation of the primal weak form from the strong form in Section 2.1. We see directly from equation (6) that the primal weak form has a minimization structure subject to the stated conditions on $\mathcal{K}_i$ and κ_j, and well-posedness follows by standard arguments.

A dual weak form for the model problem, with explicit representation of the subdomain velocities and mortar fluxes, can also be constructed. We first define the space $H(\mathrm{div}; \Omega_i, \partial_X \Omega)$ as the space of L^2-vector functions on Ω_i with weak divergence in $L^2(\Omega_i)$ and zero trace on the part of the boundary indicated by $\partial_X \Omega$. Then, we denote the product spaces of $H(\mathrm{div})$-functions that are zero on Neumann, and on Neumann and internal boundaries as:

$$V = H(\mathrm{div}; \Omega_1, \partial_N \Omega_1) \times H(\mathrm{div}; \Omega_2, \partial_N \Omega_2), \tag{7}$$

$$V_0 = H(\mathrm{div}; \Omega_1, \partial_N \Omega_1) \times H(\mathrm{div}; \Omega_2, \partial_N \Omega_2 \cup \partial_1 \Omega_2 \cup \partial_2 \Omega_2). \tag{8}$$

Furthermore, we define the L^2-product spaces on the domains:

$$L^2(\Omega) = L^2(\Omega_1) \times L^2(\Omega_2), \qquad L^2(\Gamma) = L^2(\Gamma_1) \times L^2(\Gamma_2). \tag{9}$$

With these spaces in hand, we consider the standard linear extension operators from internal boundaries onto domains denoted $\mathcal{R}_j := L^2(\Gamma_j) \to H(\mathrm{div}; \Omega_2, \partial_N \Omega_2)$, such that $\mathcal{R}_j$ satisfies for all $\lambda_j \in L^2(\Gamma_j)$

$$\mathrm{tr}_{\partial_j \Omega_2} (\mathcal{R}_j \lambda_j) \cdot n_2 = \begin{cases} \lambda_j & \text{on } \partial_j \Omega_2 \\ 0 & \text{on } \partial\Omega \setminus \partial_j \Omega_2 \end{cases}. \tag{10}$$

The precise choice of the extension operator $\mathcal{R}_j$ is not important; however, the natural choice based on the solution of an auxiliary elliptic equation is reasonable [12]. We naturally extend the definition of $\mathcal{R}_j$ to $\mathcal{R} := L^2(\Gamma) \to V$ by requiring that for $[\lambda_1, \lambda_2] \in L^2(\Gamma)$, then $[u_1, u_2] = \mathcal{R}\lambda$ satisfies $u_1 = 0$ and $u_2 = \mathcal{R}_1 \lambda_1 + \mathcal{R}_2 \lambda_2$.

The above constructions allow us to represent subdomain fluxes as

$$u = u_0 + \mathcal{R}\lambda, \tag{11}$$

where $u_0 \in V_0$ and $\lambda \in L^2(\Gamma)$. This motivates the construction of a compound $H(\mathrm{div})$-type spaces, as

$$H(\mathrm{div}; \Omega, \Gamma) = V_0 \times L^2(\Gamma). \tag{12}$$

This construction will become key when we generalize to more complex geometries in the next section.

Remark 1 (On the regularity of $H(\mathrm{div};\Omega,\Gamma)$)**.** *It is worth remarking that the restriction of space $H(\mathrm{div};\Omega,\Gamma)$ to the domain Ω_2 has slightly enhanced regularity relative to the standard space $H(\mathrm{div};\Omega_2)$, as this latter space has normal traces which do not lie in $L^2(\Gamma_1)$ nor $L^2(\Gamma_2)$.*

Definition 2 (Dual weak formulation for a single fracture.)**.** *Let $u_0 = [u_{0,1}, u_{0,2}]$, $\lambda = [\lambda_1, \lambda_2]$, $p = [p_1, p_2]$. Then find $(u_0, \lambda, p) \in H(\mathrm{div};\Omega,\Gamma) \times L^2(\Omega)$ such that*

$$\begin{aligned}\left\langle \mathcal{K}_2^{-1}\left(u_{0,2} + \mathcal{R}_1\lambda_1 + \mathcal{R}_2\lambda_2\right), v_{0,2}\right\rangle_{\Omega_2} + \left\langle \mathcal{K}_1^{-1} u_{0,1}, v_{0,1}\right\rangle_{\Omega_1} - \sum_{i=1}^{2} \langle p_i, \nabla_i \cdot v_{0,i}\rangle_{\Omega_i} \\ = -\sum_{i=1}^{2} \langle g_{D,i}, \mathrm{tr}\, v_{0,i} \cdot n_i\rangle_{\partial_D\Omega_i}, \qquad \forall\, v_0 = [v_{0,1}, v_{0,2}] \in V_0,\end{aligned} \tag{13a}$$

$$\begin{aligned}\left\langle \mathcal{K}_2^{-1}\left(u_{0,2} + \mathcal{R}_1\lambda_1 + \mathcal{R}_2\lambda_2\right), \mathcal{R}_1\nu_1 + \mathcal{R}_2\nu_2\right\rangle_{\Omega_2} - \langle p_2, \nabla_2 \cdot (\mathcal{R}_1\nu_1 + \mathcal{R}_2\nu_2)\rangle_{\Omega_2} \\ + \sum_{j=1}^{2} \left\langle \kappa_j^{-1}\lambda_j, \nu_j\right\rangle_{\Gamma_j} + \langle p_1, \nu_1 + \nu_2\rangle_{\Omega_1} = 0, \qquad \forall\, \nu = [\nu_1, \nu_2] \in L^2(\Gamma),\end{aligned} \tag{13b}$$

$$\begin{aligned}\langle \nabla_2 \cdot (u_{0,2} + \mathcal{R}_1\lambda_1 + \mathcal{R}_2\lambda_2), q_2\rangle_{\Omega_2} + \langle \nabla_1 \cdot u_{0,1}, q_1\rangle_{\Omega_1} - \langle \lambda_1 + \lambda_2, q_1\rangle_{\Omega_1} \\ = \sum_{i=1}^{2} \langle f_i, q_i\rangle_{\Omega_i}, \qquad \forall\, q = [q_1, q_2] \in L^2(\Omega).\end{aligned} \tag{13c}$$

Refer to Appendix A.2 for the derivation.

Remark 2 (Well-posedness)**.** *The variational formulation from Definition 2 can be classified as a saddle point structure, for which well-posedness results have been established for fracture networks, see e.g. Theorem 2.5 from [12].*

2.3 A first *a posteriori* error estimate for the primal variable

Having the functional spaces and weak formulations formally introduced, in this section, we provide a first upper bound for an approximation to the primal variable $q = [q_1, q_2] \in H_0^1(\Omega) + g$ for the case of a single fracture in the energy norm

$$|||q|||^2 := \sum_{i=1}^{2} \left\| \mathcal{K}_i^{\frac{1}{2}} \nabla_i\, q_i \right\|_{\Omega_i}^2 + \sum_{j=1}^{2} \left\| \kappa_j^{\frac{1}{2}} \left(q_1 - \mathrm{tr}_{\partial_j\Omega_2}\, q_2\right) \right\|_{\Gamma_j}^2. \tag{14}$$

Theorem 1 (A first upper bound for the primal variable)**.** *Let $p \in H_0^1(\Omega) + g$ be the solution to the primal weak form (6) with $\partial_D\Omega_1$ non-empty. Then for any $q \in H_0^1(\Omega) + g$, it holds that*

$$|||p - q||| \le \sum_{i=1}^{2} \eta_{\mathrm{DF},\Omega_i} + \sum_{j=1}^{2} \eta_{\mathrm{DF},\Gamma_j} + \sum_{i=1}^{2} \eta_{\mathrm{R},\Omega_i}, \quad \forall\ [v_0, \nu] \in H(\mathrm{div};\Omega,\Gamma), \tag{15}$$

with

$$\eta_{\mathrm{DF},\Omega_1} = \left\| \mathcal{K}_1^{-\frac{1}{2}} \left(v_{0,1} + \mathcal{K}_1 \nabla_1 q_1 \right) \right\|_{\Omega_1}, \tag{16a}$$

$$\eta_{\mathrm{DF},\Omega_2} = \left\| \mathcal{K}_2^{-\frac{1}{2}} \left(v_{0,2} + \mathcal{R}_1 \nu_1 + \mathcal{R}_2 \nu_2 + \mathcal{K}_2 \nabla_2 q_2 \right) \right\|_{\Omega_2}, \tag{16b}$$

$$\eta_{\mathrm{DF},\Gamma_1} = \left\| \kappa_1^{-\frac{1}{2}} \left(\nu_1 + \kappa_1 \left(q_1 - \mathrm{tr}_{\partial_1 \Omega_2} q_2 \right) \right) \right\|_{\Gamma_1}, \tag{16c}$$

$$\eta_{\mathrm{DF},\Gamma_2} = \left\| \kappa_2^{-\frac{1}{2}} \left(\nu_2 + \kappa_2 \left(q_1 - \mathrm{tr}_{\partial_2 \Omega_2} q_2 \right) \right) \right\|_{\Gamma_2}, \tag{16d}$$

$$\eta_{\mathrm{R},\Omega_1} = C_{\Omega_1} \left\| f_1 - \nabla_1 \cdot v_{0,1} + \nu_1 + \nu_2 \right\|_{\Omega_1}, \tag{16e}$$

$$\eta_{\mathrm{R},\Omega_2} = C_{\Omega_2} \left\| f_2 - \nabla_2 \cdot \left(v_{0,2} + \mathcal{R}_1 \nu_1 + \mathcal{R}_2 \nu_2 \right) \right\|_{\Omega_2}, \tag{16f}$$

where C_{Ω_1} and C_{Ω_2} are the permeability-weighted Poincaré-Friedrichs constants for Ω_1 and Ω_2:

$$C_{\Omega_i} := \sup_{q \in H^1_{0,D}(\Omega_i)} \frac{\|q\|_{\Omega_i}}{\left\| \mathcal{K}_i^{\frac{1}{2}} \nabla_i q \right\|_{\Omega_i}}. \tag{17}$$

Proof. Refer to Appendix B for the proof. □

Remark 3 (Nature of the estimators)**.** *The upper bound (15) is a guaranteed upper bound for the deviation between the primal solution $p \in H^1_0(\Omega) + g$ and an arbitrary approximation $q \in H^1_0(\Omega) + g$ in the energy space. There are three types of contributions to the upper bound: (1) diffusive flux estimators (16a) and (16b) measuring the difference between the approximate fluxes $v_0 + \mathcal{R}\nu \in V$ and fluxes obtained from $H^1_0(\Omega)$-potentials q, (2) domain coupling estimators (16c) and (16d) measuring how close the approximate normal fluxes $\nu \in L^2(\Gamma)$ are to the jump in $H^1_0(\Omega)$-potentials q, and (3) residual estimators (16e) and (16f) measuring the difference between the exact source term and the divergence of the approximate flux plus the jump in adjacent approximate normal fluxes. An important detail is that the approximate cross-domain fluxes ν_1 and ν_2 enter into the residual estimators of both the higher- and lower-dimensional subdomain.*

Remark 4 (Sharpness of the estimates)**.** *The estimates above are in principle sharp, as can be shown by standard arguments [20]. However, in practice, we will often have access to additional information about the approximate solution (most commonly if it is derived with a local conservation property). This allows for improvements in the residual estimators (16f) and (16e), as we will show in Section 5.2.*

It is clear that even for this fairly simple configuration, the variational formulations (and the analysis in general) can be quite cumbersome. The situation escalates in complexity when intersecting fractures (see Figure 3) are part of the geometric configuration, in particular as the proof of Theorem 1 relies on all subdomains having some non-vanishing Dirichlet boundary. Indeed, when floating subdomains (e.g., fully embedded fractures or isolated rock domains) are present in the fracture network, the standard procedure used in Theorem 1

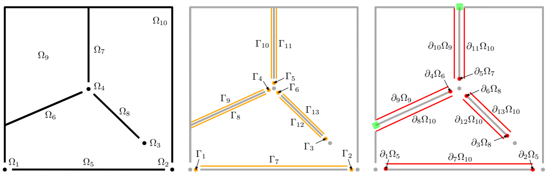


Figure 3: Mixed-dimensional geometric decomposition of a fracture network. Left: The domain Y is decomposed into two 2d matrices (Ω_9 and Ω_{10}), four 1d fractures (Ω_5, Ω_6, Ω_7, and Ω_8), one 0d fracture intersection point (Ω_4), and three 0d fracture end-points (Ω_1, Ω_2, Ω_3). Note that we allow fractures and other lower-dimensional subdomains to form parts of the boundary of the domain (e.g., Ω_5 with its endpoints Ω_1 and Ω_2). Center: Interfaces between subdomains. Right: Subdomain boundaries. Internal boundaries are depicted in red, whereas fracture's boundaries touching the ambient boundary are depicted in green.

can no longer be applied directly. Thus, in the remainder of the paper, we deal with these challenges in a more general framework.

3 Extension to fracture networks

In this section, we extend the single fracture model to account for several subdomains as part of a general fracture network. Our vocabulary is motivated by the physical case of $n = 3$, where the surrounding rock is composed of simply connected 3d subdomains, fractures are simply connected planar 2d subdomains, the intersection between such fractures are 1d lines, and the intersection between fracture intersections are 0d points (see Figure 3 for an example with $n = 2$).

We start with the classical description and then introduce the mD notation. The rest of the section is devoted to introducing key tools that are necessary to perform the analysis in an mD setting.

3.1 Mixed-dimensional geometric representation

The derivation of *a posteriori* estimates for generic fracture networks greatly benefits from an mD decomposition of the domain of interest, and we therefore follow the approach of [12]. We start by considering an n–dimensional contractible domain $Y \subset \mathbb{R}^n$, $n \in \{2, 3\}$, decomposed into m planar, open and non-intersecting subdomains Ω_i of different dimensionality $d_i = d(i)$, such that $Y = \cup_{i=1}^{m} \Omega_i$ (see left Figure 3). The partitioning is constrained such that any d-dimensional subdomain (for $d < n$) is always either the intersection of the closure of two or more subdomains of dimension $d+1$, or a cut in a domain of dimension $d+1$. This hierarchical structure excludes e.g., a 1d line or a 0d point embedded directly in a 3d domain.

We adopt a structure where neighboring subdomains one dimension apart are connected via interfaces, denoted by Γ_j for $j \in \{1, \dots, M\}$. To be precise, let Γ_j be the interface between subdomains indexed by $\check{\jmath}$ and $\hat{\jmath}$ of dimension d and $d+1$, respectively. Then $\Gamma_j = \Omega_{\check{\jmath}}$ (see center Figure 3), and furthermore, we denote the adjacent boundary of $\Omega_{\hat{\jmath}}$ by $\Gamma_j = \partial_j \Omega_{\hat{\jmath}}$. We emphasize that while the internal boundary $\partial_j \Omega_{\hat{\jmath}}$ is defined to spatially coincide with the interface Γ_j, which in turn coincides with the lower-dimensional subdomain $\Omega_{\check{\jmath}}$, their distinction is crucial to define variables properly.

To keep track of the connections from subdomains to interfaces, we introduce the sets $\hat{\mathcal{S}}_i$ and $\check{\mathcal{S}}_i$, containing the indices of the higher-dimensional (respectively lower-dimensional) neighboring interfaces of Ω_i, as illustrated in the right panel of Figure 3. These sets are dual to $\check{\jmath}$ and $\hat{\jmath}$ defined in the previous paragraph, thus for all $j \in \hat{\mathcal{S}}_i$, it holds that $\check{\jmath} = i$, while for all $j \in \check{\mathcal{S}}_i$, it holds that $\hat{\jmath} = i$.

We will be interested in defining functions on the above stated partition of the domain and the interfaces. This motivates us to define the disjoint unions

$$\Omega = \bigsqcup_{i=1}^{m} \Omega_i \qquad \text{and} \qquad \Gamma = \bigsqcup_{j=1}^{M} \Gamma_j. \tag{18}$$

A complete mixed-dimensional partitioning, including both subdomain and interfaces, is given by $\Omega \sqcup \Gamma$.

In order to speak of boundary conditions, we introduce the decomposition of the boundary of Ω. Let $\partial\Omega$ be partitioned into its Neumann, Dirichlet, and internal parts. That is, we define $\partial\Omega = \partial_N\Omega \cup \partial_D\Omega \cup \partial_I\Omega$, where $\partial_N\Omega = \cup_{i=1}^{m} \partial_N \Omega_i$, $\partial_D\Omega = \cup_{i=1}^{m} \partial_D\Omega_i$, and $\partial_I\Omega = \cup_{i=1}^{m} \cup_{j \in \check{\mathcal{S}}_i} \partial_j \Omega_i$. Finally, to ensure the existence of a unique solution, we require $\partial_D\Omega \neq \emptyset$.

3.2 The model problem for a fracture network

Let us now present the model problem valid for m subdomains of dimensionality 0 to n, and M interfaces of dimensionality 0 to $n-1$. Our model summarizes the derivations given in recent literature [12, 30, 39]. For all domains Ω_i, we consider a scalar pressure p_i together with a flux u_i in the tangent space of the domain. On all interfaces Γ_j, we consider a scalar coupling flux λ_j, oriented as positive for flow from the higher dimensional domain $\Omega_{\hat{\jmath}}$. We will, in this section, assume sufficient regularity that the strong form makes sense, and return to the weak formulation in later sections. The governing equations from the previous section then generalize as

$$\begin{aligned}
\nabla_i \cdot u_i - \textstyle\sum_{j \in \hat{S}_i} \lambda_j &= f_i, && \text{in } \Omega_i, && i \in \{1, \ldots, m\}, && && \text{(19a)}\\
u_i &= -\mathcal{K}_i \nabla_i \, p_i, && \text{in } \Omega_i, && i \in \{1, \ldots, m\}, && d_i \neq 0, && \text{(19b)}\\
\lambda_j &= -\kappa_j \left(p_{\check{j}} - p_{\hat{j}} \right), && \text{on } \Gamma_j, && j \in \{1, \ldots, M\}, && && \text{(19c)}\\
u_{\hat{j}} \cdot n_{\hat{j}} &= \lambda_j, && \text{on } \partial_j \Omega_{\hat{j}}, && j \in \{1, \ldots, M\}, && && \text{(19d)}\\
u_i \cdot n_i &= 0, && \text{on } \partial_N \Omega_i, && i \in \{1, \ldots, m\}, && && \text{(19e)}\\
p_i &= g_{D,i}, && \text{on } \partial_D \Omega_i, && i \in \{1, \ldots, m\}. && && \text{(19f)}
\end{aligned}$$

In (19a), the summation captures the contribution of fluxes from the adjacent interfaces to Ω_i, and can be seen as a generalization of the second term in (2a). Note that for $d_i = n$, the set $\hat{S}_i = \emptyset$, and thus the jump operator, evaluates to zero in the highest-dimensional domains. Conversely, in (19a), the differential term $\nabla_i \cdot u_i$ is void whenever $d_i = 0$, as there is no tangent space to a point in all subdomains, and indeed, we will not consider the u_i defined on these domains, which justifies why equation (19b) are not applied to 0d domains.

We are now ready to recast the model problem in mD notation, building on the product space structures introduced in Section 2.2. Let us start by defining the mD pressure as the ordered collection of subdomain pressures $\mathfrak{p} := [p_i] \in C\Omega$, i.e., scalar functions on Ω. We now decompose the fluxes as in (11), so that

$$u_i = u_{0,i} + \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \lambda_j \tag{20}$$

such that $u_{0,i}$ satisfies $u_{0,i} \cdot n_i = 0$ for all $j \in \check{\mathcal{S}}_i$, and where the reconstruction operator is generalized as $\mathcal{R}_j : C\Gamma_j \to C\Omega_{\hat{j}}$ satisfying:

$$\operatorname{tr}_{\partial_j \Omega_{\hat{j}}} \left(\mathcal{R}_j \, \lambda_j \right) \cdot n_{\hat{j}} = \begin{cases} \lambda_j & \text{on } \partial_j \Omega_{\hat{j}} \\ 0 & \text{on } \partial \Omega_{\hat{j}} \setminus \partial_j \Omega_{\hat{j}} \end{cases}. \tag{21}$$

This allows us to define the mD flux as the internal (tangential) domain fluxes and (normal) interface fluxes $\mathfrak{u} := [u_{0,i}, \lambda_j] \in C_0 T\Omega \times C\Gamma$, i.e., the pairing of sections of the tangent bundle $T\Omega$ together with scalar functions on Γ. By the subscript $C_0 T\Omega$, we indicate that both $u_i \cdot n_i = 0$ on all $\partial_j \Omega_i$, where $j \in \check{\mathcal{S}}_i$, and also $u_i \cdot n_i = 0$ on $\partial_N \Omega_i$.

We now define a generalized divergence operator $\mathfrak{D} \cdot (\cdot) : C_0 T\Omega \times C\Gamma \to C\Omega$ which acts on the mD flux in accordance with the left-hand side of (19a):

$$\mathfrak{D} \cdot \mathfrak{u} = \mathfrak{D} \cdot [u_{0,i}, \lambda_j] = \mathfrak{q}, \tag{22}$$

where $\mathfrak{q} = [q_i] \in C\Omega$ is a scalar function for each domain Ω_i, defined by:

$$q_i := \nabla_i \cdot \left(u_{0,i} + \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \lambda_j \right) - \textstyle\sum_{j \in \hat{\mathcal{S}}_i} \lambda_j \tag{23}$$

Similarly, we define an mD gradient operator $\mathbb{D}\,(\cdot) : C\Omega \to CT\Omega \times C\Gamma$ acting on the mD pressure in accordance with the right-hand sides of equations (19b) and (19c):

$$\mathbb{D}\, \mathfrak{p} = \mathbb{D}\, [p_i] = \mathfrak{v}, \tag{24}$$

where $\mathfrak{v} = [v_{0,i}, \nu_j] \in CT\Omega \times C\Gamma$ has the same structure as the mD flux (but without the boundary conditions), such that for all $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, M\}$, it holds that

$$\nu_j := p_{\check{j}} - p_{\hat{j}}, \qquad v_{0,i} := \nabla_i \, p_i - \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j. \tag{25}$$

Recalling that the full flux v_i is recovered from equation (20), we note that the second term above is simply the gradient on each subdomain. We will, in Section 3.3, further justify the terminology "divergence" and "gradient" due to the fact that these operators satisfy an integration-by-parts property with respect to the suitable inner products, and are thus adjoints (subject to appropriate boundary conditions).

Material parameters are collected into the mD permeability $\mathfrak{K} : CT\Omega \times C\Gamma \to CT\Omega \times C\Gamma$, defined such that for

$$-\,\mathfrak{K}\,\mathfrak{v} = -\mathfrak{K}\,[v_{0,i}, \nu_j] = \mathfrak{u}, \tag{26}$$

then from the model given in equation (19), we recognize the desired relationships

$$\lambda_j = -\kappa_j \nu_j, \qquad u_i = -\mathcal{K}_i v_i. \tag{27}$$

The second term, corresponding to Darcy's law, can be rewritten in terms of the decomposition $\mathfrak{u} = [u_{0,i}, \lambda_j]$ from equation (20) as:

$$u_{0,i} = -\mathcal{K}_i \left(v_{0,i} + \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j \right) - \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \lambda_j. \tag{28}$$

The presence of the extra terms arising from the decomposition is analogous to that in (19).

We note that the restriction $\mathfrak{u} \in C_0T\Omega \times C\Gamma$, implicitly places constraints (depending on the material constants $\mathfrak{K}$ and via the definition of $\mathbb{D}$) on the admissible pressures $\mathfrak{p}$. This space of admissible pressures can be understood as the domain of the restricted operator $\mathfrak{K}\mathbb{D} : C\Omega \to C_0T\Omega \times C\Gamma$.

In view of the mD variables and operators defined above, and subject to the right-hand side data $\mathfrak{f} = [f_i] \in C\Omega$ and the boundary data $\mathfrak{g}_D = [g_{D,i}] \in C\partial_D\Omega$, a straightforward substitution of definitions shows that problem (19) is equivalent to the concisely stated mD elliptic problem

$$\mathfrak{u} = -\mathfrak{K}\,\mathbb{D}\,\mathfrak{p}, \qquad \text{in}\, \Omega \times \Gamma, \tag{29a}$$

$$\mathfrak{D} \cdot \mathfrak{u} = \mathfrak{f}, \qquad \text{in}\, \Omega, \tag{29b}$$

$$\mathfrak{p} = \mathfrak{g}_D, \qquad \text{on}\, \partial_D\Omega, \tag{29c}$$

defined for $\mathfrak{u} \in C_0T\Omega \times C\Gamma$ and $\mathfrak{p} \in C\Omega$.

Remark 5 (Internal Neumann boundaries)**.** *For simplicity of exposition, the domain Y is taken as contractible, and Ω_i is considered a partitioning of Y. However, the reader will appreciate that these assumptions can be relaxed. Most importantly, from the perspective of applications (as discussed in Section 2.1),* *some internal interfaces may be modeled as impermeable, i.e.* $\lambda_j = 0$. *We refer to the remaining (permeable) interfaces as* $\Xi \subset \{0, \dots, M\}$. *The impermeable interfaces are then excluded from the problem, and considered as internal Neumann interfaces. To be precise, we define a reduced disjoint union of interface domains*

$$\Gamma = \bigsqcup_{j \in \Xi} \Gamma_j.$$

The internal Neumann boundaries may partition the domain into disconnected parts. We refer to a subdomain as "Dirichlet-connected", denoted $i \in \xi$ *if either (1)* $\partial_D \Omega_i \neq \emptyset$, *or (2) there exists some* $j \in \hat{\mathcal{S}}_i$ *such that* $\hat{\jmath} \in \xi$, *or (3) there exists some* $j \in \check{\mathcal{S}}_i$ *such that* $\check{\jmath} \in \xi$. *This allows us to construct a reduced disjoint union of subdomains*

$$\Omega = \bigsqcup_{i \in \xi} \Omega_i.$$

All the derivations in the continuation are equally valid for these reduced product domains.

Remark 6 (Extensions to the model equations)**.** *The results of this paper can with minor modifications be extended to non-zero Neumann boundary conditions, and with some additional effort to the class of non-planar geometries considered in [15]. However, as this generality is typically not needed for applications, we restrict the presentation as indicated above.*

3.3 Variational formulations in mixed-dimensional notation

Before writing the variational formulations in mD notation, let us first define the relevant mD inner products and norms. Consider the following inner-products

$$\langle \mathfrak{q}, \mathfrak{r} \rangle_\Omega = \sum_{i=1}^{m} \langle q_i, r_i \rangle_{\Omega_i} \qquad \forall\ \mathfrak{q} = [q_i], \mathfrak{r} = [r_i] \in L^2\Omega, \tag{30}$$

$$\begin{aligned} \langle \mathfrak{v}, \mathfrak{w} \rangle_{\Omega,\Gamma} = \sum_{i=1}^{m} \Bigg(&\Big\langle \Big(v_{0,i} + \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j\Big), \Big(w_{0,i} + \textstyle\sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \mu_j\Big) \Big\rangle_{\Omega_i} \\ &+ \sum_{j \in \check{S}_i} \langle \nu_j, \mu_j \rangle_{\Gamma_j} \Bigg) \qquad \forall\ \mathfrak{v} = [v_{0,i}, \nu_j], \mathfrak{w} = [w_{0,i}, \mu_j] \in L^2 T\Omega \times L^2\Gamma, \end{aligned} \tag{31}$$

$$\langle \mathfrak{q}, \mathfrak{r} \rangle_{\partial_X \Omega} = \sum_{i=1}^{m} \langle q_i, r_i \rangle_{\partial_X \Omega_i} \quad \forall\ \mathfrak{q} = [q_i], \mathfrak{r} = [r_i] \in L^2 \partial_X \Omega, \tag{32}$$

and their respective induced norms

$$\|\mathfrak{q}\|_\Omega^2 = \langle \mathfrak{q}, \mathfrak{q} \rangle_\Omega, \qquad \|\mathfrak{v}\|_{\Omega,\Gamma}^2 = \langle \mathfrak{v}, \mathfrak{v} \rangle_{\Omega,\Gamma}, \qquad \|\mathfrak{q}\|_{\partial_X \Omega}^2 = \langle \mathfrak{q}, \mathfrak{q} \rangle_{\partial_X \Omega}. \tag{33}$$

With these inner products, the previously defined mD divergence satisfy the following integration-by-parts formula [12, 15] whenever $\mathfrak{v} \in CT\Omega \times C\Gamma$ and $\mathfrak{q} \in C\Omega$.

$$\langle \mathfrak{q}, \mathfrak{D} \cdot \mathfrak{v} \rangle_{\Omega} + \langle \mathbb{D}\, \mathfrak{q}, \mathfrak{v} \rangle_{\Omega,\Gamma} = \langle \mathfrak{T}_D \mathfrak{q}, \mathfrak{T}_D \mathfrak{v} \rangle_{\partial_D \Omega} + \langle \mathfrak{T}_N \mathfrak{q}, \mathfrak{T}_N \mathfrak{v} \rangle_{\partial_N \Omega}. \tag{34}$$

In the above the restriction to the boundary is denoted $\mathfrak{T}_X(\cdot)$ (for $X = D, N$), which depending on context acts as the boundary values of pressure variables, $\mathfrak{T}_X(\cdot) : C\Omega \to C\partial_X \Omega$, or the normal component of flux variables, $\mathfrak{T}_X(\cdot) : CT\Omega \times C\Gamma \to C\partial_X \Omega$.

From the product structure in the definition of the C and L^2 spaces, the continuous spaces inherit their density from the individual subdomains to the product spaces on Ω and Γ. We can thus follow standard procedures to obtain weak extensions of the mD differential operators, the boundary restriction (trace) operators, and the corresponding function spaces [40, 41, 42]. We elaborate this below.

Due to the density of $C_0 T\Omega \times C\Gamma$ in $L^2 T\Omega \times L^2 \Gamma$, the mD divergence from Section 3.2 is a densely defined unbounded linear operator on the latter space $\mathfrak{D}\cdot : L^2\Omega \to L^2 T\Omega \times L^2\Gamma$. Let us now (temporarily) use the notation $(T, \mathrm{dom}(T))$ to emphasize that an operator T has domain of definition $\mathrm{dom}(T)$, and we denote the adjoint operator with respect to the L^2 inner product by an asterisk.

We recall that the Neumann boundary is incorporated into the definition of the continuous flux spaces $C_0 T\Omega \times C\Gamma$, thus the last term in the integration-by-parts formula (34), is zero. Hence, we can define a weak mD gradient and the corresponding space of weakly mD differentiable functions with zero trace on the Dirichlet boundary H_0^1 by considering the adjoint:

$$(\mathbb{D}\,, H_0^1(\Omega)) := (\mathfrak{D}\cdot, C_0 T\Omega \times C\Gamma)^*. \tag{35}$$

Clearly, $C_0\Omega \subseteq H_0^1(\Omega)$, and thus it is appropriate to consider $(\mathbb{D}\,, H_0^1(\Omega))$ as a weak gradient. Moreover, the domain of definition simply corresponds to the standard $H_0^1(\Omega_i)$ on each domain, where the subscript zero indicates zero trace on all Dirichlet boundaries. Thus $H_0^1(\Omega) = \prod_{i=1}^m H_0^1(\Omega_i)$, which generalizes (5).

Considering the integration-by-parts formula again, the weak mD divergence and the corresponding space of flux functions with divergence in L^2 and zero trace on the Neumann boundary $H(\mathrm{div}; \Omega, \Gamma)$ can be defined as

$$(\mathfrak{D}\cdot, H(\mathrm{div}; \Omega, \Gamma)) := (\mathbb{D}\,, H_0^1)^*. \tag{36}$$

Again $C_0 T\Omega \times C\Gamma \subseteq H(\mathrm{div}; \Omega, \Gamma)$, and it is appropriate to consider $(\mathfrak{D}\cdot, H(\mathrm{div}; \Omega, \Gamma))$ as a weak divergence. This domain of definition of the weak divergence has the interpretation of $H_0(\mathrm{div}; \Omega_i)$ on all subdomains Ω_i (where the subscript zero indicates zero trace on all boundaries except for Dirichlet boundaries), and $L^2(\Gamma_j)$ spaces on all interfaces Γ_j. Thus $H(\mathrm{div}; \Omega, \Gamma) = \prod_{i=1}^m H_0(\mathrm{div}; \Omega_i) \times \prod_{i=1}^M L^2(\Gamma_j)$, which generalizes (12).

Due to the above identification of $H^1(\Omega)$ and $H(\mathrm{div}; \Omega, \Gamma)$ in terms of product spaces of standard function spaces on subdomains, we extend the definition of the boundary restriction operators $\mathfrak{T}_X(\cdot)$ to trace operators on the weak spaces by requiring that they coincide with the standard trace operators on subdomains.

In the continuation, we will always consider the weak mD gradient and divergence, and denote these simply by $\mathbb{D}$ and $\mathfrak{D}\cdot$, respectively. Similarly, we will always consider the boundary restrictions as trace operators. The above definitions of weak mD gradient and divergence operators, and their adjoint property on the above weak spaces, has the following statements of the primal and dual weak formulations of equations (29) as a direct consequence:

Definition 3 (Mixed-dimensional primal weak form)**.** *Let* $\mathfrak{g} \in H^1(\Omega)$*. Then find* $\mathfrak{p} \in H_0^1(\Omega) + \mathfrak{g}$ *such that*

$$\langle \mathfrak{K}\mathbb{D}\,\mathfrak{p}, \mathbb{D}\,\mathfrak{q}\rangle_{\Omega,\Gamma} = \langle \mathfrak{f}, \mathfrak{q}\rangle_{\Omega} \qquad \forall\, \mathfrak{q} \in H_0^1(\Omega). \tag{37}$$

Definition 4 (Mixed-dimensional dual weak form)**.** *Find* $(\mathfrak{u}, \mathfrak{p}) \in H(\mathrm{div}; \Omega, \Gamma) \times L^2(\Omega)$ *such that*

$$\left\langle \mathfrak{K}^{-1}\mathfrak{u}, \mathfrak{v}\right\rangle_{\Omega,\Gamma} - \langle \mathfrak{p}, \mathfrak{D}\cdot\mathfrak{v}\rangle_{\Omega} = \langle \mathfrak{g}_D, \mathfrak{T}_D\mathfrak{v}\rangle_{\partial_D\Omega} \qquad \forall\, \mathfrak{v} \in H(\mathrm{div}; \Omega, \Gamma), \tag{38a}$$

$$\langle \mathfrak{D}\cdot\mathfrak{u}, \mathfrak{q}\rangle_{\Omega} = \langle \mathfrak{f}, \mathfrak{q}\rangle_{\Omega} \qquad \forall\, \mathfrak{q} \in L^2(\Omega). \tag{38b}$$

The above weak forms of the mixed-dimensional elliptic problem are well-posed for bounded coefficients [15], in the sense that there exist positive constants $\mathfrak{K}_0$ and $\mathfrak{K}_\infty$ such that:

$$\sup_{\mathfrak{v}\in H(\mathrm{div};\Omega,\Gamma)} \frac{\langle \mathfrak{K}\,\mathfrak{v}, \mathfrak{v}\rangle_{\Omega,\Gamma}}{\mathfrak{K}_\infty \|\mathfrak{v}\|^2_{\Omega,\Gamma}} \leq 1 \leq \inf_{\mathfrak{v}\in H(\mathrm{div};\Omega,\Gamma)} \frac{\langle \mathfrak{K}\,\mathfrak{v}, \mathfrak{v}\rangle_{\Omega,\Gamma}}{\mathfrak{K}_0 \|\mathfrak{v}\|^2_{\Omega,\Gamma}}. \tag{39}$$

The solutions of the primal and dual weak formulations are equivalent, and define true solutions $\mathfrak{p} \in H_0^1(\Omega) + \mathfrak{g}$ and $\mathfrak{u} \in H(\mathrm{div}; \Omega, \Gamma)$ against which the approximate solutions will be measured in later sections.

4 Functional analysis tools

In this section, we summarize the main functional analysis tools we will exploit for the *a posteriori* analysis.

4.1 Poincaré-type inequalities

We recall the following weighted Poincaré inequalities:

Lemma 1 (Permeability-weighted Poincaré-Friedrichs inequalities)**.** *There exist* *constants $C_\Omega \geq C_{\Omega_i} \geq C_K$ such that*

$$
\begin{aligned}
\|\mathfrak{q}\|_{\Omega,\Gamma} &\leq C_{\Omega,\Gamma} \left\|\mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,\mathfrak{q}\right\|_{\Omega,\Gamma} && \forall\ \mathfrak{q} \in H_0^1(\Omega), && && \text{(40a)}\\
\|q\|_{\Omega_i} &\leq C_{\Omega_i} \left\|\mathcal{K}_i^{\frac{1}{2}}\nabla_i q\right\|_{\Omega_i} && \forall\ q \in H_0^1(\Omega_i), && \text{if } \partial_D\Omega_i \neq \emptyset, && \text{(40b)}\\
\|q - \tilde{q}_{\Omega_i}\|_{\Omega_i} &\leq C_{\Omega_i} \left\|\mathcal{K}_i^{\frac{1}{2}}\nabla_i q\right\|_{\Omega_i} && \forall\, q \in H^1(\Omega_i), && \text{if } \partial_D\Omega_i = \emptyset, && \text{(40c)}\\
\|q - \tilde{q}_K\|_K &\leq C_K \left\|\mathcal{K}_i^{\frac{1}{2}}\nabla_i q\right\|_K && \forall\ q \in H^1(K), && \text{where } K \subset \Omega_i. && \text{(40d)}
\end{aligned}
$$

Here, we denote by $\tilde{q}_{\Omega_i}$ and $\tilde{q}_K$ the mean value of q over the subdomain Ω_i and an arbitrary d_i-simplex $K \subset \Omega_i$, respectively.

We refer to $C_{\Omega,\Gamma}$ as the mixed-dimensional permeability-weighted Poincaré-Friedrichs constant (whose existence was shown in [15]), C_{Ω_i} is the standard subdomain permeability-weighted Poincaré-Friedrichs constant, and C_K is a local permeability-weighted Poincaré-Friedrichs constant.

It is important to mention that concrete values of C_{Ω_i} are available only for a limited set of geometries, see e.g., [43, 44, 45]. An upper bound exists for convex domains, and thus for a simplex $K \subset \Omega_i$ we have [46, 47]

$$C_K \leq \frac{\operatorname{diam}(K)}{\pi c_K} \tag{41}$$

where c_K is the lower bound on the permeability within K:

$$c_K = \inf_{\substack{x\in K\\ v\in TK_x}} \frac{(\mathcal{K}_i(x)v)\cdot v^{\,2}}{\|v\|} \tag{42}$$

The importance of this is understood if K is an element of a simplicial partition of Ω_i, in which case C_K scales with the mesh size $h_K = \operatorname{diam}(K)$. This allows for super-convergent properties of residual estimators for some locally mass-conservative approximations [26, 48, 49]. We analyze these cases with further details in Section 5.2 and Remark 15.

4.2 Conforming flux spaces

It is often possible to verify that an approximate solution $\mathfrak{v} \in H(\operatorname{div};\Omega,\Gamma)$ satisfies some stronger conservation property, that is to say, that there is some space $U \subseteq L^2$ such that

$$\mathfrak{D}\cdot\mathfrak{v} - \mathfrak{f} \in U \tag{43}$$

This allows for the construction of stronger *a posteriori* estimates, and as such, we formalize this concept as a generalization of $H(\operatorname{div};\Omega,\Gamma)$ to "U-conforming flux spaces":

Definition 5 (Conforming mD flux space)**.** *Let $H(\operatorname{div};\Omega,\Gamma;U) \subset H(\operatorname{div};\Omega,\Gamma)$ be a U-conforming flux space, in the sense of*

$$H(\operatorname{div};\Omega,\Gamma;U) = \{\mathfrak{v} \in H(\operatorname{div};\Omega,\Gamma) : \mathfrak{f} - \mathfrak{D}\cdot\mathfrak{v} \in U\}\,. \tag{44}$$

To exploit the conforming flux spaces, we must construct certain projected $H^1(\Omega)$ spaces. Consider therefore U as some subspace of $L^2(\Omega)$ and define $U^\perp$ to be its orthogonal complement:

$$U^\perp := \{\mathfrak{q} \in L^2(\Omega) : \langle \mathfrak{q}, \mathfrak{r} \rangle_\Omega = 0 \quad \forall\ \mathfrak{r} \in U\}. \tag{45}$$

Moreover, let $\pi_{U^\perp}$ be the L^2–projection onto $U^\perp$, such that for any $\mathfrak{r} \in L^2(\Omega)$, $\pi_{U^\perp}\mathfrak{r} \in U^\perp$ satisfies the orthogonality property:

$$\langle \mathfrak{r} - \pi_{U^\perp}\mathfrak{r}, \mathfrak{q} \rangle_\Omega = 0 \quad \forall\ \mathfrak{q} \in U^\perp. \tag{46}$$

Consider now the projected $H_0^1(\Omega)$ space denoted $W \subset L^2(\Omega)$, defined as the range of $\pi_W := (I - \pi_{U^\perp}) : H_0^1(\Omega) \to L^2(\Omega)$, and let the norm of W be defined as a weighted L^2-norm with nonnegative weights $\mu \in L^\infty(\Omega)$

$$\|\mathfrak{q}\|_{W,\mu} := \|\mu\mathfrak{q}\|_\Omega \qquad \forall\ \mathfrak{q} \in W, \tag{47}$$

which are defined within the class $\mathcal{C}_W$ with unit Poincaré constants:

$$\mathcal{C}_W = \left\{ \mu \in L^\infty(\Omega) : \sup_{\mathfrak{q} \in H_0^1(\Omega)} \frac{\|\pi_W \mathfrak{q}\|_{W,\mu}}{\left\|\mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,\mathfrak{q}\right\|_{\Omega,\Gamma}} \leq 1 \right\}. \tag{48}$$

Indeed, such classes exist in the literature of Poincaré inequalities for weighted norms, see e.g., [29, 28]. Note that a trivial member of $\mathcal{C}_W$ is the inverse of the permeability-weighted mD Poincaré-Friedrichs constant $\mu(x) = C_{\Omega,\Gamma}^{-1}$. As we will see in Sections 5.1 and 5.2, the concrete choice of the space U and the corresponding weights μ will directly impact the strength of the estimates.

Remark 7 (On the space $H(\mathrm{div};\Omega,\Gamma;U)$)**.** *The conforming mD flux spaces allow us to obtain sharper estimates in Section 5. However, it is important to note that the standard case $U = L^2(\Omega)$ is included in our definition, for which the orthogonal complement is void, and the projection $\pi_W = I$; thus $W = H_0^1(\Omega)$. This and other cases are elaborated in more detail in Sections 5.2.1 to 5.2.4.*

4.3 Bilinear forms and energy norms

For the *a posteriori* analysis, we will need the next two mD bilinear forms and their induced energy norms

$$\mathfrak{B}(\mathfrak{q},\mathfrak{r}) = \langle \mathfrak{K}\mathbb{D}\,\mathfrak{q}, \mathbb{D}\,\mathfrak{r} \rangle_{\Omega,\Gamma}, \quad |||\mathfrak{q}|||^2 = \mathfrak{B}(\mathfrak{q},\mathfrak{q}) = \left\|\mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,\mathfrak{q}\right\|_{\Omega,\Gamma}^2 \quad \forall\ \mathfrak{q},\mathfrak{r} \in H_0^1(\Omega), \tag{49}$$

$$\mathfrak{A}(\mathfrak{v},\mathfrak{w}) = \left\langle \mathfrak{v}, \mathfrak{K}^{-1}\mathfrak{w} \right\rangle_{\Omega,\Gamma}, \quad |||\mathfrak{v}|||_*^2 = \mathfrak{A}(\mathfrak{v},\mathfrak{v}) = \left\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{v}\right\|_{\Omega,\Gamma}^2 \quad \forall\ \mathfrak{v},\mathfrak{w} \in L^2T\Omega \times L^2\Gamma, \tag{50}$$

which are related via

$$|||\mathfrak{q}||| = |||\mathfrak{K}\mathbb{D}\,\mathfrak{q}|||_* \qquad \forall\,\mathfrak{q} \in H_0^1(\Omega). \tag{51}$$

We also define the *full* norm for a mixed-dimensional pair of primal and dual variables as

$$|||[\mathfrak{q},\mathfrak{v}]||| := |||\mathfrak{q}||| + |||\mathfrak{v}|||_* + \left\|\mu^{-1}\mathfrak{D}\cdot\mathfrak{v}\right\|_\Omega \quad \forall\,(\mathfrak{q},\mathfrak{v}) \in H_0^1(\Omega)\times H(\mathrm{div};\Omega,\Gamma;U). \tag{52}$$

Note that the last norm will depend on the eventual choice of μ^{-1}, which we emphasize must be from the class $\mu \in \mathcal{C}_W$, as defined in the preceding section.

5 A posteriori error estimates

This section is devoted to obtaining the error bounds for our model problem. First, we provide general abstract estimates, and later we focus on the evaluation of the different bounds.

5.1 General abstract estimates

Let us now present the general abstract bounds. We formalize the main results presented in Section 3 and extend the ones presented in Theorem 1 in the following theorem.

Theorem 2 (General abstract a posteriori error bounds)**.** *Let the error majorant be defined as*

$$\mathcal{M}(\mathfrak{q},\mathfrak{v},\mathfrak{f},\mu) := \eta_{\mathrm{DF}}(\mathfrak{q},\mathfrak{v}) + \eta_{\mathrm{R}}(\mathfrak{v},\mathfrak{f},\mu), \tag{53}$$

where

$$\eta_{\mathrm{DF}}(\mathfrak{q},\mathfrak{v}) := |||\mathfrak{v} + \mathfrak{K}\mathbb{D}\,\mathfrak{q}|||_* \quad \text{and} \quad \eta_{\mathrm{R}}(\mathfrak{v},\mathfrak{f},\mu) := \left\|\mu^{-1}(\mathfrak{f} - \mathfrak{D}\cdot\mathfrak{v})\right\|_\Omega, \tag{54}$$

valid for all $\mathfrak{q} \in H_0^1(\Omega) + \mathfrak{g}$ *and* $\mathfrak{v} \in H(\mathrm{div};\Omega,\Gamma;U)$. *Then, the following a posteriori error estimates hold.*
(1) Let $\mathfrak{p} \in H_0^1(\Omega) + \mathfrak{g}$ *be the solution to* (37) *and* $\mathfrak{q} \in H_0^1(\Omega) + \mathfrak{g}$ *be arbitrary. Then*

$$|||\mathfrak{p} - \mathfrak{q}||| \leq \mathcal{M}_{\mathfrak{p}}^{\oplus} = \mathcal{M}(\mathfrak{q},\mathfrak{v},\mathfrak{f},\mu) \qquad \forall\ \mathfrak{v} \in H(\mathrm{div};\Omega,\Gamma;U), \tag{55}$$

where $\mathcal{M}_{\mathfrak{p}}^{\oplus}$ *is the upper bound of the error for the primal variable.*
(2) Let $\mathfrak{u} \in H(\mathrm{div};\Omega,\Gamma)$ *be the solution to* (38) *and* $\mathfrak{v} \in H(\mathrm{div};\Omega,\Gamma;U)$ *be arbitrary. Then*

$$|||\mathfrak{u} - \mathfrak{v}|||_* \leq \mathcal{M}_{\mathfrak{u}}^{\oplus} = \mathcal{M}(\mathfrak{q},\mathfrak{v},\mathfrak{f},\mu) \qquad \forall\ \mathfrak{q} \in H_0^1(\Omega) + \mathfrak{g}, \tag{56}$$

where $\mathcal{M}_{\mathfrak{u}}^{\oplus}$ *is the upper bound of the error for the dual variable.*
(3) Let $\mathfrak{p} \in H_0^1(\Omega)+\mathfrak{g}$ *be the solution to* (37) *and* $\mathfrak{u} \in H(\mathrm{div};\Omega,\Gamma)$ *be the solution to* (38)*, and let* $(\mathfrak{q},\mathfrak{v}) \in (H_0^1(\Omega) + \mathfrak{g}) \times H(\mathrm{div};\Omega,\Gamma;U)$ *be arbitrary. Then,*

$$\mathcal{M}(\mathfrak{q},\mathfrak{v},\mathfrak{f},\mu) = \mathcal{M}_{\mathfrak{p},\mathfrak{u}}^{\ominus} \leq |||[\mathfrak{p}-\mathfrak{q},\mathfrak{u}-\mathfrak{v}]||| \leq \mathcal{M}_{\mathfrak{p},\mathfrak{u}}^{\oplus} = 2\mathcal{M}(\mathfrak{q},\mathfrak{v},\mathfrak{f},\mu) + \eta_{\mathrm{R}}(\mathfrak{v},\mathfrak{f},\mu), \tag{57}$$

where $\mathcal{M}_{\mathfrak{p},\mathfrak{u}}^{\ominus}$ *and* $\mathcal{M}_{\mathfrak{p},\mathfrak{u}}^{\oplus}$ *are the lower and upper bounds of the error for the primal-dual variable.*

Proof. Due to the construction of mixed-dimensional product spaces and the adjoint property of the weak differential operators, the proof from the mono-dimensional case can (to a large extent) be applied directly [19]. A notable deviation from the standard proofs is the use of conforming flux spaces, and the inclusion of the Poincaré-constants in the weights $\mathcal{C}_W$. The full proof is included for completeness in Appendix C. □

Remark 8 (Non-conforming approximations)**.** *Referring again to the general setting of mD calculus, it has been shown that the differential operators form part of a cochain complex, and that an mD Helmholtz decomposition exists [15]. Thus, by realizing the above constructions as Hilbert complexes, the above error bounds can be extended also to non-conforming approximations following, e.g., Theorem 4.7 of [21]. However, as a main objective of our work is to obtain bounds based on conforming properties of the approximations, we will not pursue non-conforming approximations in this work.*

5.2 Evaluation of the majorant

The aim of this section is to provide concrete forms of the majorant $\mathcal{M}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}, \mu)$ from Theorem 2 depending upon the choices of the weights μ. For this purpose, consider once again the definition of the majorant

$$\begin{aligned}&\mathcal{M}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}, \mu) = \eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v}) + \eta_{\mathrm{R}}(\mathfrak{v}, \mathfrak{f}, \mu)\\ &\qquad \forall\ \mathfrak{q} = [q_i] \in H_0^1(\Omega) + \mathfrak{g},\ \ \mathfrak{v} = [v_{0,i}, \nu_j] \in H(\mathrm{div}; \Omega, \Gamma; U).\end{aligned} \tag{58}$$

The estimation of the first term $\eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v})$ is independent of the weights μ. Indeed, by applying (50), it is straightforward to see that

$$\begin{aligned}\eta_{\mathrm{DF}}^2(\mathfrak{q}, \mathfrak{v}) &= \sum_{i=1}^{m} \left(\sum_{K \in \mathcal{T}_{\Omega_i}} \left\| \mathcal{K}_i^{-\frac{1}{2}} \left(v_{0,i} + \sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j \right) + \mathcal{K}_i^{\frac{1}{2}} \nabla_i q_i \right\|_K^2 \right.\\ &\qquad \left. + \sum_{j \in \check{\mathcal{S}}_i} \sum_{K \in \mathcal{T}_{\Gamma_j}} \left\| \kappa_j^{-\frac{1}{2}} \nu_j + \kappa_j^{\frac{1}{2}} \left(q_{\check{j}} - \mathrm{tr}\, q_{\hat{j}} \right) \right\|_K^2 \right)\\ &= \sum_{i=1}^{m} \left(\sum_{K \in \mathcal{T}_{\Omega_i}} \eta_{\mathrm{DF}_{\parallel}, K}^2 + \sum_{j \in \check{\mathcal{S}}_i} \sum_{K \in \mathcal{T}_{\Gamma_j}} \eta_{\mathrm{DF}_{\perp}, K}^2 \right).\end{aligned} \tag{59}$$

The terms $\eta_{\mathrm{DF}_{\parallel}, K}$ and $\eta_{\mathrm{DF}_{\perp}, K}$ measure the diffusive flux error in the tangential and normal directions associated with the subdomain element $K \in \mathcal{T}_{\Omega_i}$ and the mortar element $K \in \mathcal{T}_{\Gamma_j}$, respectively.

To complete the evaluation of the majorant, we are left with the estimation of $\eta_{\mathrm{R}}(\mathfrak{v}, \mathfrak{f}, \mu)$, which depends on the choices of μ. Recall that this term measures the mismatch in satisfying the conservation equation in each subdomain. To be precise, there are four main types of conforming fluxes; Standard L^2-conforming, subdomain conservation, grid level (local) conservation, and point-wise. The quality of the residual balance can be verified explicitly *before* applying the *a posteriori* estimates, and thus is not considered an assumption in the theory. Below, we make precise the aforementioned cases.

5.2.1 No mass-conservation

Assume nothing is known about the approximation of the residual terms beyond the L^2 structure. We indicate this case by the abbreviation "NC", and set $U_{\mathrm{NC}} = L^2$, and $\mathfrak{v} \in H(\mathrm{div}; \Omega, \Gamma; U_{\mathrm{NC}}) = H(\mathrm{div}; \Omega, \Gamma)$. Then $U_{\mathrm{NC}}^{\perp} = 0$, which implies that $\pi_W = I$, and $W = H_0^1(\Omega)$. Then, *a priori*, we only know the global (mixed-dimensional) Poincaré (40a), i.e., we have no better weight than setting $\mu(x) = C_{\Omega,\Gamma}^{-1}$ for $x \in \Omega$.

Using (49) and the mD Poincaré inequality (40a), one obtains the following bound, which is the weakest bound available within the class of bounds considered in this paper:

$$\begin{aligned} \eta_{\mathrm{R}}^2 &\leq C_{\Omega,\Gamma}^2 \sum_{i=1}^{m} \sum_{K \in \mathcal{T}_{\Omega_i}} \left\| f_i - \nabla_i \cdot \left(v_{0,i} + \sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j \right) + \sum_{j \in \hat{\mathcal{S}}_i} \nu_j \right\|_K^2 \\ &= \sum_{i=1}^{m} \sum_{K \in \mathcal{T}_{\Omega_i}} \eta_{\mathrm{R},K;\mathrm{NC}}^2 = \eta_{\mathrm{R;NC}}^2, \end{aligned} \tag{60}$$

Here, $\eta_{\mathrm{R},K;\mathrm{NC}}$ denotes the local residual error for non-conservative approximations. The majorant when mass conservation cannot be guaranteed at any level is then given by,

$$\mathcal{M}_{\mathrm{NC}}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}) = \eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v}) + \eta_{\mathrm{R;NC}}(\mathfrak{v}, \mathfrak{f}), \tag{61}$$

and it follows from the above that this is an upper bound, $\mathcal{M} \leq \mathcal{M}_{\mathrm{NC}}$.

5.2.2 Subdomain mass-conservation

Due to the structure of the equations, where interface fluxes are stated explicitly, many approximations will have mass conservation satisfied in a subdomain level, which is in a sense a compatibility condition on the floating domains Ω_i. We indicate this case by the abbreviation "SC". In particular, the divergence $\mathfrak{r} = [r_i] = \mathfrak{D} \cdot \mathfrak{v} \in U_{\mathrm{SC}}$ satisfies for all $i \in \{1, \ldots, m\}$ where $\partial_D \Omega_i = \emptyset$,

$$\langle r_i, 1 \rangle_{\Omega_i} = \langle f_i, 1 \rangle_{\Omega_i}. \tag{62}$$

Thus, by definition $U_{\mathrm{SC}}^{\perp}$ is the space of constants over the floating subdomains Ω_i, and the space W is the space of $\mathring{H}^1(\Omega_i)$ functions, with zero mean if $\partial_D \Omega_i = \emptyset$.

This case represents an improvement relative to the previous one, in the sense that we can now employ the subdomain Poincaré constants instead of the mD constant. Let us make this point precise in the following lemma.

Lemma 2. *Let $W = \prod_{i=1}^{m} \mathring{H}^1(\Omega_i)$, where*

$$\mathring{H}^1(\Omega_i) = \left\{ q_i \in H_0^1(\Omega_i) \mid \langle q_i, 1\rangle_{\Omega_i} = 0 \text{ if } \partial_D \Omega_i = \emptyset \right\}. \tag{63}$$

Then, $\mu(x) = C_{\Omega_i}^{-1}$ for $x \in \Omega_i$ belongs to the class $\mathcal{C}_W$, where C_{Ω_i} is the permeability-weighted Poincaré-Friedrichs constants defined in Lemma 1.

Proof. Using the Poincaré inequality (40c) and the fact that the sum of broken norms is weaker than the full norm, the following result holds

$$\begin{aligned}
\sup_{\mathfrak{q} \in H_0^1(\Omega)} \frac{\|\pi_W \mathfrak{q}\|_{W,\mu}}{\left\|\mathfrak{K}^{\frac{1}{2}} \mathbb{D}\, \mathfrak{q}\right\|_{\Omega,\Gamma}} &= \sup_{\substack{\mathfrak{q} \in H_0^1(\Omega) \\ \left\|\mathfrak{K}^{\frac{1}{2}} \mathbb{D}\, \mathfrak{q}\right\|_{\Omega,\Gamma} = 1}} \|\pi_{W_\Omega} \mathfrak{q}\|_{W_\Omega,\mu} \\
&= \sup_{\substack{\mathfrak{q} \in H_0^1(\Omega) \\ \left\|\mathfrak{K}^{\frac{1}{2}} \mathbb{D}\, \mathfrak{q}\right\|_{\Omega,\Gamma} = 1}} \left(\sum_{\substack{i=1 \\ \partial_D \Omega_i \neq \emptyset}}^{m} \left\|C_{\Omega_i}^{-1} q_i\right\|_{\Omega_i} + \sum_{\substack{i=1 \\ \partial_D \Omega_i = \emptyset}}^{m} \left\|C_{\Omega_i}^{-1} \left(q_i - \frac{1}{|\Omega_i|}\langle q_i, 1\rangle_{\Omega_i}\right)\right\|_{\Omega_i} \right) \\
&\leq \sup_{\substack{\mathfrak{q} \in H_0^1(\Omega) \\ \left\|\mathfrak{K}^{\frac{1}{2}} \mathbb{D}\, \mathfrak{q}\right\|_{\Omega,\Gamma} = 1}} \sum_{i=1}^{m} \left\|\mathcal{K}^{\frac{1}{2}} \nabla_i q_i\right\|_{\Omega_i} \leq 1.
\end{aligned}$$

□

In view of Lemma 2, η_{R} can be bounded as

$$\begin{aligned}
\eta_{\mathrm{R}}^2 &\leq \sum_{i=1}^{m} C_{\Omega_i}^2 \sum_{K \in \mathcal{T}_{\Omega_i}} \left\| f_i - \nabla_i \cdot \left(v_{0,i} + \sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j \right) + \sum_{j \in \hat{\mathcal{S}}_i} \nu_j \right\|_K^2 \\
&= \sum_{i=1}^{m} \sum_{K \in \mathcal{T}_{\Omega_i}} \eta_{\mathrm{R},K;\mathrm{SC}}^2 = \eta_{\mathrm{R;SC}}^2,
\end{aligned} \tag{64}$$

where $\eta_{\mathrm{R},K;\mathrm{SC}}$ are the local residual estimators for subdomain mass-conservative approximations. The majorant for this case is given by

$$\mathcal{M}_{\mathrm{SC}}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}) = \eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v}) + \eta_{\mathrm{R;SC}}(\mathfrak{v}, \mathfrak{f}). \tag{65}$$

This estimate is sharper than that in the preceding section, since $C_{\Omega_i} \leq C_{\Omega,\Gamma}$, thus whenever the assumptions of this section are satisfied, it holds that $\mathcal{M} \leq \mathcal{M}_{\mathrm{SC}} \leq \mathcal{M}_{\mathrm{NC}}$.

Note that (65) is identical in structure to the residual estimators (16e) and (16f) obtained in Theorem 1. However, they are fundamentally different in the sense that (65) do not require all subdomains to have a non-empty Dirichlet part but rather mass to be conserved in each subdomain Ω_i.

5.2.3 Local mass-conservation

By choice of numerical method, it is often easy to verify that mass is conserved on an element basis in a subdomain partition. We indicate this case by the abbreviation "LC". As in the preceding section, this implies that the divergence $\mathfrak{r} = [r_i] = \mathfrak{D} \cdot \mathfrak{v} \in U_{\mathrm{LC}}$ then satisfies for all $K \subset \mathcal{T}_{\Omega_i}$ that

$$\langle r_i, 1 \rangle_K = \langle f_i, 1 \rangle_K, \tag{66}$$

where $\mathcal{T}_{\Omega_i}$ denotes a finite partition of Ω_i (typically a simplicial grid). In this case, U_{LC} contain functions having zero mean on each element $K \in \mathcal{T}_{\Omega_i}$, and from (66) we see that $U_{\mathrm{LC}}^{\perp} = \prod_{i=1}^{m} \mathbb{P}_0(\mathcal{T}_{\Omega_i})$.

We will consider the slightly weaker case, where (66) is only required to hold for all "non-Dirichlet boundary" elements, that is for all elements where $\partial K \cap \partial_D \Omega = \emptyset$. This is sufficient for the results from Lemma 2 to be extendable to the grid level by considering the space $W_\Omega = \prod_{i=1}^{m} \prod_{K \in \mathcal{T}_{\Omega_i}} \mathring{H}^1(K)$, where $\mathring{H}^1(K)$ is defined in (63).

Lemma 2 now applies without modification, and weights $\mu(x) \geq C_K^{-1}$ for $x \in K$ are therefore in $\mathcal{C}_W$. Moreover, thanks to convexity of simplicial grid elements, the local permeability-weighted Poincaré-Friedrichs constants are now fully computable. This allows us to bound $\eta_{\mathrm{R},\Omega}$ as follows:

$$\begin{aligned}
\eta_{\mathrm{R},\Omega}^2 &\leq \sum_{i=1}^{m} \sum_{K \in \mathcal{T}_{\Omega_i}} \frac{h_K^2}{\pi^2 c_K^2} \left\| f_i - \nabla_i \cdot \left(v_{0,i} + \sum_{j \in \check{\mathcal{S}}_i} \mathcal{R}_j \nu_j \right) + \sum_{j \in \hat{\mathcal{S}}_i} \nu_j \right\|_K^2 \\
&= \sum_{i=1}^{m} \sum_{K \in \mathcal{T}_{\Omega_i}} \eta_{\mathrm{R},K;\mathrm{LC}}^2 = \eta_{\mathrm{R},\Omega;\mathrm{LC}}^2,
\end{aligned} \tag{67}$$

where $\eta_{\mathrm{R},K;\mathrm{LC}}$ are the local residual estimators for locally mass-conservative approximations. Using the above results, the majorant for locally mass-conservative approximations reads

$$\mathcal{M}_{\mathrm{LC}}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}) = \eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v}) + \eta_{\mathrm{R;LC}}(\mathfrak{v}, \mathfrak{f}). \tag{68}$$

The local residual estimates $\eta_{\mathrm{R},\Omega;\mathrm{LC}}$ correspond to the ones previously obtained by [26, 49] for mono-dimensional problems subject to a flux equilibration step. Since $C_K \leq C_{\Omega_i}$, then, as before, whenever the assumptions of this section are satisfied, it holds that $\mathcal{M} \leq \mathcal{M}_{\mathrm{LC}} \leq \mathcal{M}_{\mathrm{SC}} \leq \mathcal{M}_{\mathrm{NC}}$.

Remark 9 (Fully computable residual estimators)**.** *Unlike estimators obtained with residual methods (containing unknown constants [50, 51]) or a purely functional approach such as in Sections 5.2.1 and 5.2.2 (containing constants that are generally difficult to determine [20]), estimators such as* (67) *contain only known local constants depending on the mesh size and material parameters. This justifies the claim that these estimators are fully computable.*

5.2.4 Exact mass-conservation

Methods with local mass conservation, as discussed in the previous section, when applied to problems where the RHS data $\mathfrak{f}$ is zero or piecewise constant, can then often be verified to have an exact (pointwise) conservation property. We indicate this case by the abbreviation "EC", for which $\mathfrak{f} = \mathfrak{D} \cdot \mathfrak{v}$, so that $U_{\mathrm{EC}} = 0$ and $U_{\mathrm{EC}}^{\perp} = L^2(\Omega)$. Now, $\pi_W = 0$ and $W = 0$. Thus, any finite weights μ are admissible, yet the choice is immaterial since the residual term $\left\|\mu^{-1}(\mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v})\right\|_{\Omega}$ always evaluates to zero. Consequently, only diffusive-type errors are present in the *a posteriori* estimation, and the majorant takes the form

$$\mathcal{M}_{\mathrm{EC}}(\mathfrak{q}, \mathfrak{v}) = \eta_{\mathrm{DF}}(\mathfrak{q}, \mathfrak{v}). \tag{69}$$

This case can also be seen as the limiting case of local mass conservation for a family of grid partitions where $h_K \to 0$.

5.2.5 Summary of majorants and subdomain errors

With the obtained majorants, we can define the corresponding upper bounds for the errors of the primal, dual, and primal-dual variables.

Definition 6. *Let $\alpha = \mathrm{NC}, \mathrm{SC}, \mathrm{LC}, \mathrm{EC}$, corresponding to the flux conformity spaces U_α discussed in the preceding sections. Then, in view of the results from Theorem 2 and the majorants (61), (65), (68), and (69), the upper bounds for the error in the primal, dual, and primal-dual pair, for arbitrary approximations $\mathfrak{q} \in H_0^1(\Omega) + \mathfrak{g}$ and $\mathfrak{v} \in H(\mathrm{div}; \Omega, \Gamma; U_\alpha)$, are*

$$\mathcal{M}^{\oplus}_{\mathfrak{p};\alpha} := \mathcal{M}_\alpha, \qquad \mathcal{M}^{\oplus}_{\mathfrak{u};\alpha} := \mathcal{M}_\alpha, \qquad \mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\alpha} := 2\mathcal{M}_\alpha + \eta_{\mathrm{R};\alpha}. \tag{70}$$

while the lower bound for the error in the primal-dual pair is

$$\mathcal{M}^{\ominus}_{\mathfrak{p},\mathfrak{u};\alpha} := \mathcal{M}_\alpha. \tag{71}$$

It is our interest not only to measure local errors, but also to distinguish between subdomain and interface errors. This motivates the definition of the following errors estimators.

Definition 7 (Subdomain and interface error indicators)**.** *Let $\alpha = \mathrm{EC}, \mathrm{LC}, \mathrm{SC}, \mathrm{NC}$. Then, we will denote by $\varepsilon_{\Omega_i;\alpha}$ and ε_{Γ_j} the subdomain and interface error indicators, defined by*

$$\varepsilon^2_{\Omega_i;\alpha} := \varepsilon^2_{\mathrm{DF},\Omega_i} + \varepsilon^2_{\mathrm{R},\Omega_i;\alpha} := \sum_{K \in \mathcal{T}_{\Omega_i}} \eta^2_{\mathrm{DF}_{\parallel},K} + \sum_{K \in \mathcal{T}_{\Omega_i}} \eta^2_{\mathrm{R},K;\alpha},$$

$$\varepsilon^2_{\Gamma_j} := \varepsilon^2_{\mathrm{DF},\Gamma_j} := \sum_{K \in \mathcal{T}_{\Gamma_j}} \eta^2_{\mathrm{DF}_{\perp},K}.$$

We emphasize that while the majorants provide guaranteed bounds, the subdomain and interface error indicators can only be expected to correlate with the error.

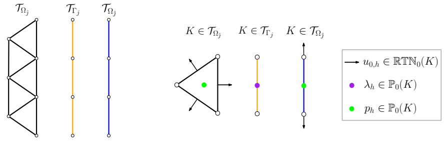

$\mathcal{T}_{\Omega_{\hat{j}}}$
$\mathcal{T}_{\Gamma_j}$
$\mathcal{T}_{\Omega_{\check{j}}}$
$K \in \mathcal{T}_{\Omega_{\hat{j}}}$
$K \in \mathcal{T}_{\Gamma_j}$
$K \in \mathcal{T}_{\Omega_{\check{j}}}$
$u_{0,h} \in \mathbb{RTN}_0(K)$
$\lambda_h \in \mathbb{P}_0(K)$
$p_h \in \mathbb{P}_0(K)$

Figure 4: Left: Matching coupling between the grids $\mathcal{T}_{\Omega_{\hat{j}}}$, $\mathcal{T}_{\Gamma_j}$, and $\mathcal{T}_{\Omega_{\check{j}}}$. Right: Degrees of freedom involved in the coupling between a 2d higher-dimensional cell, a 1d mortar-cell, and a 1d lower-dimensional cell. Locally, tangential fluxes are approximated using $\mathbb{RTN}_0(K)$, whereas mortar fluxes and pressures using $\mathbb{P}_0(K)$.

6 Concrete bounds for locally mass-conservative approximations

In this section, we will make the evaluation of the bounds concrete by providing explicit approximations to (38) using the lowest-order mixed-finite element method (MFEM).

6.1 Grid partitions

Ultimately, *a posteriori* estimates are primarily applied to approximations that are defined on computational grids. We therefore, in this section, summarize the relevant notation for grids and the mapping operators between subdomains and interfaces.

Let us start by defining the partitions of the domains of interest. To this aim, denote by $\mathcal{T}_{\Omega_i}$, $\mathcal{T}_{\Gamma_j}$, and $\mathcal{T}_{\partial_i \Omega_j}$ the partitions of Ω_i, Γ_j, and $\partial_j \Omega_i$, respectively. Moreover, let $\mathcal{T}_{\Omega} = \cup_{i=1}^{m} \mathcal{T}_{\Omega_i}$, $\mathcal{T}_{\Gamma} = \cup_{j=1}^{M} \mathcal{T}_{\Gamma_j}$, and $\mathcal{T}_{\partial_I \Omega} = \cup_{i=1}^{m} \cup_{j \in \check{\mathcal{S}}_i} \partial_j \Omega_i$ represent the union of all subdomain, mortar, and internal boundary grids.

Here, we only consider simplicial partitions. In particular, we require all elements $K \subset \Omega_i$ to be strictly non-overlapping simplices of dimension $d_K = d_i$. We use h_K to denote the diameter of K, and define $h_{\Omega_i} = \max_{h_K} \mathcal{T}_{\Omega_i}$, $h_{\Gamma_j} = \max_{h_K} \mathcal{T}_{\Gamma_j}$, and $h_{\partial_j \Omega_i} = \max_{h_K} \mathcal{T}_{\partial_j \Omega_i}$.

We will not at this point place any conditions on the grid partitions, although several aspects of this will be advantageous from the perspective of computation.

6.2 Finite element spaces and the approximated problem

Let us introduce the finite element spaces necessary to write the approximated problem. We start by defining a local space for the approximated pressures, mortar fluxes, and tangential fluxes. They are given, respectively by

$$
\begin{aligned}
Q_{h,i} &:= \left\{q_{h,i} \in L^2(\Omega_i) : q_{h,i}|_K \in \mathbb{P}_0(K)\ \forall\, K \in \mathcal{T}_{\Omega_i}\right\}, && d_i \in \{0, \dots, n\}, \\
\Lambda_{h,j} &:= \left\{\nu_{h,j} \in L^2(\Gamma_j) : \nu_{h,j}|_K \in \mathbb{P}_0(K)\ \forall\, K \in \mathcal{T}_{\Gamma_j}\right\}, && d_j \in \{0, \dots, n-1\}, \\
V_{h,i} &:= \left\{v_{h,i} \in H(\mathrm{div}; \Omega_i) : v_{h,i}|_K \in \mathbb{RTN}_0(K)\ \forall\, K \in \mathcal{T}_{\Omega_i}\right\}, && d_i \in \{1, \dots, n\},
\end{aligned}
$$

where $\mathbb{P}_0$ and $\mathbb{RTN}_0$ denote the spaces of constants and lowest-order Raviart-Thomas(-Nédélec) spaces of vector functions [52, 53]. See also Figure 4 for the degrees of freedom involved in the generic coupling between a (higher-dimensional) triangle, a mortar line segment, and a (lower-dimensional) line segment.

The composite space for the approximated mD pressure $Q_h \subset L^2(\Omega)$ and the approximated mD flux $X_h \subset H(\mathrm{div}; \Omega, \Gamma)$ are defined respectively by

$$
Q_h := \prod_{i=1}^{m} Q_{h,i} \qquad \text{and} \qquad X_h := \prod_{i=1}^{m} \left(H_0(\mathrm{div}; \Omega_i) \cap V_{h,i} \times \prod_{j \in \check{\mathcal{S}}_i} \mathcal{R}_{h,j}\, \Lambda_{h,j} \right). \tag{72}
$$

While not strictly necessary from a theoretical perspective, in the discrete setting, it is often useful to choose a finite-dimensional reconstruction operator based on the discrete spaces, and we allow for this through the notation $\mathcal{R}_{h,j} : \Lambda_{h,j} \to H(\mathrm{div}; \Omega_i)$, which in practice is often further restricted to $\mathcal{R}_{h,j} : \Lambda_{h,j} \to V_{h,\hat{j}}$. Such discrete reconstruction operators are natural for matching grids, and can also be constructed in the more general case of non-matching grids, see e.g., [12, 54, 55]. Here $\Pi_h : \Lambda_{h,j} \to \tilde{\Lambda}_{h,j}$ is the L^2 projection from the mortar grid on Γ_j to the boundary simplicial partition of $\Omega_{\hat{j}}$.

We have now all the elements necessary to write the finite-dimensional approximation to the dual mixed problem (38).

Definition 8 (Approximated mD dual mixed formulation)**.** *Find* $(\mathfrak{u}_h, \mathfrak{p}_h) \in X_h \times Q_h$ *such that*

$$
\begin{aligned}
\left\langle \mathfrak{K}^{-1}\mathfrak{u}_h, \mathfrak{v}_h \right\rangle_{\Omega,\Gamma} - \langle \mathfrak{p}_h, \mathfrak{D} \cdot \mathfrak{v}_h \rangle_{\Omega} &= \langle \mathfrak{g}_D, \mathfrak{T}_D \mathfrak{v}_h \rangle_{\partial_D \Omega} && \forall\, \mathfrak{v}_h \in X_h, && \text{(73a)} \\
\langle \mathfrak{D} \cdot \mathfrak{u}_h, \mathfrak{q}_h \rangle_{\Omega} &= \langle \mathfrak{f}, \mathfrak{q}_h \rangle_{\Omega} && \forall\, \mathfrak{q}_h \in Q_h. && \text{(73b)}
\end{aligned}
$$

Due to the presence of the discrete reconstruction operator, this approximation is conforming whenever $\Lambda_{h,j} = \tilde{\Lambda}_{h,j}$, i.e., for matching grids. For non-matching grids, the approximation is still convergent, subject to normal conditions on the mortar grids [12].

Remark 10 (Conservation properties)**.** *Whenever equation* (73b) *is satisfied exactly, then equation* (66) *holds, and we have local mass conservation for matching grids. Thus, the fluxes lie in the smaller space* $X_h \cap H(\mathrm{div}; \Omega, \Gamma; Q_{h,i}^{\perp})$, *and the results from section 5.2.3 apply. Furthermore, if* $f_i \in Q_{h,i}$ *and* $\mathcal{R}_{h,j} : \Lambda_{h,j} \to V_{h,\hat{j}}$, *then the projection of the source term, and hence the residual error, onto* $Q_{h,i}^{\perp}$ *vanishes. Thus, the local conservation is verified to be pointwise, the fluxes lie in* $X_h \cap H(\mathrm{div}; \Omega, \Gamma; 0)$ *and the results from Section 5.2.4 apply.*

Remark 11 (Well-posedness and a priori estimates)**.** *The stability and a priori approximation properties of the finite-dimensional system given in* (73) *has been previously established [12].*

6.3 Pressure reconstruction

Recall that Theorem 2 requires any approximation to the mD flux to be in $H(\mathrm{div};\Omega,\Gamma)$, whereas approximations to the mD pressure must lie in $H_0^1(\Omega)+\mathfrak{g}$. By the condition that $\mathfrak{u}_h \in X_h \subset H(\mathrm{div};\Omega,\Gamma)$, the solution of equations (73) by definition satisfy the first condition. On the other hand, the approximated mD pressure $\mathfrak{p}_h$ is only in $L^2(\Omega)$. We therefore need to enhance the regularity of the approximated pressure and thus obtain a reconstructed pressure.

Definition 9 (Reconstructed pressure)**.** *We will call reconstructed pressure* $\tilde{\mathfrak{p}}_h$ *to any function constructed from the mD pair* $(\mathfrak{p}_h, \mathfrak{u}_h) \in L^2(\Omega) \times H(\mathrm{div};\Omega,\Gamma)$ *such that*

$$\tilde{\mathfrak{p}}_h \in H_0^1(\Omega) + \mathfrak{g}. \tag{74}$$

Remark 12 (On potential reconstruction)**.** *Several techniques for obtaining* $\tilde{\mathfrak{p}}_h$ *are available in the literature. Arguably, the simplest option is to perform an average of the* $\mathbb{P}_0(K)$ *pressures on local patches and from there construct local affine* $\mathbb{P}_1(K)$ *functions [56]. Other techniques aim at solving first a local Neumann problem to obtain a* $\mathbb{P}_2(K)$ *post-processed pressure, and then apply interpolation techniques to get energy-conforming potentials [27, 57, 26, 58, 59]. Any of these choices are compatible with the bounds derived herein.*

Remark 13 (Computable estimates)**.** *Computable versions of the majorants are now readily available by setting* $(\mathfrak{q}, \mathfrak{v}) = (\tilde{\mathfrak{p}}_h, \mathfrak{u}_h)$ *in* (61), (65), (68), *and* (69).

Remark 14 (Other locally mass-conservative methods)**.** *In addition to the MFEM scheme of the lowest-order (RT0-P0), other flux-based numerical methods such as the Mixed Virtual Element Method (MVEM) [60, 61] and Cell Centered Finite Volume Methods (CCFVM), including the Two-Point Flux Approximation (TPFA) and the Multi-Point Flux Approximation (MPFA) [62, 63], can be analyzed with our framework provided that the fluxes are interpolated in* X_h *and the pressures reconstructed as indicated above. For methods without an explicit flux representation, an additional flux reconstruction step may be needed.*

Remark 15 (Superconvergence of the residual estimators)**.** *Due to Remark 10, the residual estimators* $\eta_{\mathrm{R,K,LC}}$ *are superconvergent for lowest-order locally mass-conservative approximations. This property is guaranteed since: (1) local Poincaré constants decay as* $\mathcal{O}(h_K)$ *for simplicial elements and (2) the norm of the residual* $\left\| f_i - \nabla_i \cdot v_i + \sum_{j \in \hat{S}_i} \nu_j \right\|_K$ *also decays as* $\mathcal{O}(h_K)$ *[64]; leading to an overall rate of* $\mathcal{O}(h_K^2)$.

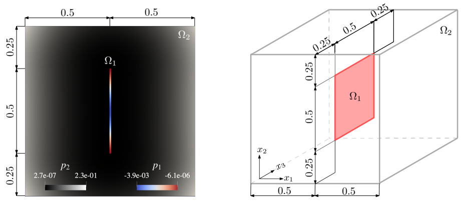


Figure 5: Geometric setups used for the numerical validations. Left: A 1d fracture embedded in a 2d matrix and the exact pressure solution. Right: A 2d fracture embedded in a 3d matrix.

7 Numerical validations

In this section, we test the performance of our estimators by conducting an efficiency analysis using four different numerical methods, namely those mentioned in Remark 14: RT0-P0, MVEM-P0, MPFA, and TPFA. The numerical examples are implemented in the Python-based open-source software `PorePy` [39], using the extension package `mdestimates` [65], which includes the scripts of all numerical examples considered here. In these numerical validations, we only consider matching grids, and use a low-order pressure reconstruction (recall Remark 12 for further discussion).

We validate the *a posteriori* bounds and assess their efficiency on a 1d/2d problem (Section 7.2) and a 2d/3d problem (Section 7.3), both with manufactured solutions. The geometric configuration for both problems is shown in Figure 5. Let us denote the fracture as Ω_1, the matrix as Ω_2, the left interface as Γ_1, and the right interface as Γ_2. Further, assume the existence of an exact, smooth pressure $p_2(x)$ in Ω_2. Refer to Table 7 and Table 8 from the Appendix D for the analytical expressions of all variables of interest.

7.1 Efficiency indices

Efficiency indices are used to assess the performance of the approximations when exact solutions are available. They are defined as the ratio between the estimated and the exact errors. Here, we consider the following efficiency indices.

Definition 10 (Efficiency indices)**.** *Let* $\alpha = \mathrm{NC}, \mathrm{SC}, \mathrm{LC}, \mathrm{EC}$ *and let* $\mathfrak{p} \in H_0^1(\Omega) + \mathfrak{g}$ *and* $\mathfrak{u} \in H(\mathrm{div}; \Omega, \Gamma)$ *be the solutions to* (37) *and* (38)*, respectively. Then, in view of Theorem* 2*, the efficiency indices for the primal, dual, and primal-dual*

Table 1: Two-dimensional validation: Majorants and efficiency indices.

	h_{coup}	$\mathcal{M}^{\oplus}_{\mathfrak{p};\text{NC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p};\text{LC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\text{NC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\text{LC}}$	$I_{\mathfrak{p};\text{NC}}$	$I_{\mathfrak{p};\text{LC}}$	$I_{\mathfrak{u};\text{NC}}$	$I_{\mathfrak{u};\text{LC}}$	$I_{\mathfrak{p},\mathfrak{u};\text{NC}}$	$I_{\mathfrak{p},\mathfrak{u};\text{LC}}$
RT0-P0	0.05	5.86e-02	4.36e-02	1.33e-01	8.83e-02	1.46	1.08	4.09	3.04	1.89	1.59
	0.025	3.01e-02	2.17e-02	6.89e-02	4.38e-02	1.49	1.07	4.18	3.02	1.91	1.58
	0.0125	1.52e-02	1.08e-02	3.48e-02	2.17e-02	1.50	1.07	4.22	3.00	1.92	1.57
	0.00625	7.65e-03	5.37e-03	1.76e-02	1.08e-02	1.52	1.07	4.25	2.98	1.93	1.57
MVEM-P0	0.05	6.18e-02	4.68e-02	1.40e-01	9.47e-02	1.42	1.07	4.31	3.26	1.89	1.60
	0.025	3.10e-02	2.27e-02	7.08e-02	4.56e-02	1.46	1.07	4.31	3.15	1.91	1.59
	0.0125	1.54e-02	1.10e-02	3.53e-02	2.22e-02	1.49	1.07	4.29	3.07	1.92	1.58
	0.00625	7.72e-03	5.44e-03	1.77e-02	1.09e-02	1.51	1.06	4.28	3.02	1.92	1.57
MPFA	0.05	5.91e-02	4.41e-02	1.34e-01	8.93e-02	1.46	1.09	4.12	3.07	1.89	1.59
	0.025	3.03e-02	2.19e-02	6.92e-02	4.41e-02	1.49	1.08	4.20	3.04	1.91	1.58
	0.0125	1.52e-02	1.08e-02	3.49e-02	2.18e-02	1.50	1.07	4.23	3.01	1.92	1.57
	0.00625	7.66e-03	5.38e-03	1.76e-02	1.08e-02	1.52	1.07	4.25	2.99	1.93	1.57
TPFA	0.05	6.67e-02	5.17e-02	1.50e-01	1.04e-01	1.54	1.19	3.09	2.39	1.84	1.58
	0.025	3.74e-02	2.90e-02	8.35e-02	5.83e-02	1.68	1.31	2.36	1.83	1.78	1.52
	0.0125	2.64e-02	2.20e-02	5.73e-02	4.41e-02	1.82	1.52	1.64	1.36	1.63	1.44
	0.00625	1.37e-02	1.15e-02	2.98e-02	2.30e-02	1.64	1.37	1.82	1.52	1.64	1.44

The results for $\mathcal{M}^{\oplus}_{\mathfrak{u};\text{NC}}$ and $\mathcal{M}^{\oplus}_{\mathfrak{u};\text{LC}}$ are omitted since they are equal to $\mathcal{M}^{\oplus}_{\mathfrak{p};\text{NC}}$ and $\mathcal{M}^{\oplus}_{\mathfrak{p};\text{LC}}$.

pair, for arbitrary approximations $\mathfrak{q} \in H^1_0(\Omega) + \mathfrak{g}$ *and* $\mathfrak{v} \in H(\text{div}; \Omega, \Gamma; U_\alpha)$*, are*

$$I_{\mathfrak{p};\alpha}(\mathfrak{q}) := \frac{\mathcal{M}^{\oplus}_{\mathfrak{p};\alpha}}{|||\mathfrak{p} - \mathfrak{q}|||}, \qquad I_{\mathfrak{u};\alpha}(\mathfrak{v}) := \frac{\mathcal{M}^{\oplus}_{\mathfrak{u};\alpha}}{|||\mathfrak{u} - \mathfrak{v}|||_*}, \qquad I_{\mathfrak{p},\mathfrak{u};\alpha}(\mathfrak{q}, \mathfrak{v}) := \frac{\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\alpha}}{|||\mathfrak{p} - \mathfrak{q}, \mathfrak{u} - \mathfrak{v}|||}. \tag{75}$$

Remark 16. *Optimal efficiency indices (equal to 1) are obtained when the approximations match the exact solutions. Moreover, in general the efficiency indices satisfy the bounds:*

$$1 \leq I_{\mathfrak{p};\alpha}(\mathfrak{q}), \qquad 1 \leq I_{\mathfrak{u};\alpha}(\mathfrak{v}), \qquad 1 \leq I_{\mathfrak{p},\mathfrak{u};\alpha}(\mathfrak{q}, \mathfrak{v}) \leq \frac{\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\alpha}}{\mathcal{M}^{\ominus}_{\mathfrak{p},\mathfrak{u};\alpha}} = 2 + \frac{\eta_{\text{R};\alpha}}{\mathcal{M}_\alpha} \tag{76}$$

For the final term, we note that since $\eta_{\text{R};\alpha} \leq \mathcal{M}_\alpha$*, then for* $\alpha = \text{NC}, \text{SC}$ *the total efficiency index satisfies* $I_{\mathfrak{p},\mathfrak{u};\alpha} \leq 3$*, while for local conservation* $I_{\mathfrak{p},\mathfrak{u};\text{LC}} \leq 2 + O(h^2)$ *and finally for exact conservation* $I_{\mathfrak{p},\mathfrak{u};\text{EC}} \leq 2$*.*

7.2 Two-dimensional validation

For our first validation, we consider the 1d/2d case as shown in the left Figure 5. This validation has two purposes: (1) compare the majorants and efficiency indices obtained using global (no mass-conservation) and local (local mass-conservation) Poincaré-Friedrichs constants, and (2) show the different errors associated with subdomains and interfaces.

To this aim, we consider four levels of successively refined combinations of mesh sizes, characterized by $h_{\text{coup}} = h_{\partial_1 \Omega_2} = h_{\Gamma_1} = h_{\Omega_1} = h_{\Gamma_2} = h_{\partial_2 \Omega_2}$. The global Poincaré constant is obtained numerically by solving the associated eigenvalue problem (see e.g., [66]), giving a value of $C_{\Omega,\Gamma} \approx 0.2251$.

Table 2: Two-dimensional validation: Subdomain and interface errors.

	h_{coup}	$\varepsilon_{\text{DF},\Omega_2}$	$\varepsilon_{\text{R},\Omega_2;\text{NC}}$	$\varepsilon_{\text{R},\Omega_2;\text{LC}}$	$\varepsilon_{\text{DF},\Omega_1}$	$\varepsilon_{\text{R},\Omega_1;\text{NC}}$	$\varepsilon_{\text{R},\Omega_1;\text{LC}}$	$\varepsilon_{\text{DF},\Gamma_1}$	$\varepsilon_{\text{DF},\Gamma_2}$
RT0-P0	0.05	4.24e-02	1.41e-02	1.00e-03	2.26e-03	7.99e-03	5.65e-04	1.89e-04	1.89e-04
	0.025	2.14e-02	7.73e-03	3.02e-04	1.14e-03	4.01e-03	1.42e-04	9.03e-05	9.15e-05
	0.0125	1.07e-02	4.00e-03	7.28e-05	5.70e-04	2.01e-03	3.55e-05	4.41e-05	4.41e-05
	0.00625	5.34e-03	2.07e-03	1.91e-05	2.85e-04	1.00e-03	8.87e-06	2.20e-05	2.20e-05
MVEM-P0	0.05	4.55e-02	1.41e-02	1.00e-03	3.25e-03	7.99e-03	5.65e-04	2.52e-04	2.52e-04
	0.025	2.23e-02	7.73e-03	3.02e-04	1.32e-03	4.01e-03	1.42e-04	1.00e-04	1.03e-04
	0.0125	1.09e-02	4.00e-03	7.28e-05	5.98e-04	2.01e-03	3.55e-05	4.50e-05	4.50e-05
	0.00625	5.41e-03	2.07e-03	1.91e-05	2.89e-04	1.00e-03	8.87e-06	2.21e-05	2.21e-05
MPFA	0.05	4.29e-02	1.41e-02	1.00e-03	2.52e-03	7.99e-03	5.65e-04	2.05e-04	2.05e-04
	0.025	2.15e-02	7.73e-03	3.02e-04	1.18e-03	4.01e-03	1.42e-04	9.24e-05	9.35e-05
	0.0125	1.07e-02	4.00e-03	7.28e-05	5.77e-04	2.01e-03	3.55e-05	4.44e-05	4.44e-05
	0.00625	5.36e-03	2.07e-03	1.91e-05	2.86e-04	1.00e-03	8.87e-06	2.20e-05	2.20e-05
TPFA	0.05	5.04e-02	1.41e-02	1.00e-03	2.52e-03	7.99e-03	5.65e-04	1.87e-04	1.89e-04
	0.025	2.86e-02	7.73e-03	3.02e-04	1.18e-03	4.01e-03	1.42e-04	9.42e-05	9.23e-05
	0.0125	2.19e-02	4.00e-03	7.28e-05	5.77e-04	2.01e-03	3.55e-05	4.47e-05	4.46e-05
	0.00625	1.14e-02	2.07e-03	1.91e-05	2.86e-04	1.00e-03	8.87e-06	2.20e-05	2.21e-05

Majorants for the primal, dual, and primal-dual variables are shown in Table 1. We can see that all majorants reflect the convergence tendency of the numerical methods, and in particular (as is well-known), we identify that the TPFA approximation performs relatively poorly on this problem. As expected, the majorants obtained exploiting the local conservation properties of the methods are sharper than the ones obtained using global weights, both in absolute value and in terms of efficiency index.

Further inspection shows that efficiency indices lie within the expected bounds discussed in Remark 16. In particular, efficiency indices for the primal variable using local weights are very accurate, and only a $\sim 7\%$ deviation with respect to the actual error (for the finest grid) is observed in the case of RT0-P0, MVEM-P0, and MPFA. For TPFA, the efficiency index is worse, as a consequence of the flux approximation being worse. Efficiency indices for the dual variable are in general larger than the ones obtained for the primal variable; this is to be expected for mixed-dual approximations with the relatively simple pressure reconstruction, where the approximated fluxes have relatively good accuracy as compared to the reconstructed pressures. Finally, efficiency indices for the primal-dual variable are less than 2 for all methods in consideration.

Considering now the local error indicators, shown in Table 2, we note that diffusive errors decrease linearly for the matrix, fracture, and interfaces. Likewise, residual errors for the matrix and fracture decrease linearly when the global Poincaré-Friedrichs constant is used. When the local Poincaré-Freidrich constants are used, the residual estimators for the matrix and the fracture decrease quadratically, which goes in agreement with the super-convergent properties discussed in Remark 15.

Table 3: Three-dimensional validation: Majorants and efficiency indices.

	h_{coup}	$\mathcal{M}^{\oplus}_{\mathfrak{p};\text{NC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p};\text{LC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\text{NC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\text{LC}}$	$I_{\mathfrak{p};\text{NC}}$	$I_{\mathfrak{p};\text{LC}}$	$I_{\mathfrak{u};\text{NC}}$	$I_{\mathfrak{u};\text{LC}}$	$I_{\mathfrak{p},\mathfrak{u};\text{NC}}$	$I_{\mathfrak{p},\mathfrak{u};\text{LC}}$
RT0-P0	0.2625	2.85e-01	2.36e-01	6.21e-01	4.73e-01	1.25	1.03	4.14	3.43	1.72	1.43
	0.1720	1.94e-01	1.62e-01	4.20e-01	3.24e-01	1.23	1.03	4.22	3.53	1.73	1.50
	0.0827	1.07e-01	8.69e-02	2.33e-01	1.74e-01	1.28	1.04	3.61	2.94	1.70	1.48
	0.0418	5.62e-02	4.58e-02	1.23e-01	9.16e-02	1.25	1.02	3.16	2.58	1.63	1.43
MVEM-P0	0.2625	2.89e-01	2.40e-01	6.28e-01	4.80e-01	1.24	1.03	4.19	3.48	1.72	1.44
	0.1720	1.96e-01	1.64e-01	4.24e-01	3.28e-01	1.23	1.03	4.26	3.57	1.73	1.50
	0.0827	1.08e-01	8.80e-02	2.35e-01	1.76e-01	1.27	1.04	3.65	2.98	1.70	1.48
	0.0418	5.66e-02	4.62e-02	1.24e-01	9.23e-02	1.25	1.02	3.18	2.60	1.63	1.44
MPFA	0.2625	2.90e-01	2.40e-01	6.29e-01	4.82e-01	1.25	1.03	4.08	3.38	1.72	1.43
	0.1720	1.98e-01	1.66e-01	4.28e-01	3.32e-01	1.23	1.03	4.22	3.54	1.73	1.50
	0.0827	1.09e-01	8.90e-02	2.37e-01	1.78e-01	1.27	1.04	3.64	2.98	1.70	1.49
	0.0418	5.69e-02	4.65e-02	1.24e-01	9.30e-02	1.25	1.02	3.18	2.60	1.63	1.44
TPFA	0.2625	3.84e-01	3.35e-01	8.17e-01	6.70e-01	1.24	1.08	2.13	1.86	1.48	1.28
	0.1720	2.95e-01	2.63e-01	6.23e-01	5.27e-01	1.38	1.23	1.66	1.48	1.44	1.30
	0.0827	2.22e-01	2.02e-01	4.63e-01	4.04e-01	1.62	1.48	1.40	1.28	1.45	1.35
	0.0418	2.08e-01	1.97e-01	4.26e-01	3.95e-01	1.76	1.67	1.29	1.23	1.46	1.41

The results for $\mathcal{M}^{\oplus}_{\mathfrak{u};\text{NC}}$ and $\mathcal{M}^{\oplus}_{\mathfrak{u};\text{LC}}$ are omitted since they are equal to $\mathcal{M}^{\oplus}_{\mathfrak{p};\text{NC}}$ and $\mathcal{M}^{\oplus}_{\mathfrak{p};\text{LC}}$.

7.3 Three-dimensional validation

For our next numerical validation, we employ the 2d/3d configuration from the right Figure 5. We repeat the same analysis from the previous section, and investigate four refinement levels. The mixed-dimensional Poincaré constant for this configuration corresponds to a value of $C_{\Omega,\Gamma} \approx 0.1838$. The results are shown in Table 3 and Table 4. As in the previous validation, we can see that the majorants capture the local and global convergence tendency of all numerical methods. Again, RT0-P0, MVEM-P0, and MPFA give quite similar results, whereas TPFA showcase larger errors. As expected, efficiency indices again lie within the stated bounds from Remark 16.

8 Numerical applications

In this section, we apply our estimators to numerical approximations of challenging problems solving the equations of incompressible flow in fractured porous media. Importantly, since source terms are zero in both applications, by applying matching grids the residual errors are zero, and we are in the setting of having an exact conservation property from the numerical approximation. From Remark 16, we then know that the efficiency index for the primal-dual error will be less than 2; even if the exact solution and error are both unknown.

8.1 Two-dimensional application

In this numerical experiment, we consider the benchmark case 3b from [13]. As shown in the left panel of Figure 1, the domain consists of ten (partially intersecting) fractures embedded in a unit square matrix. The exact fracture

Table 4: Three-dimensional validation: Subdomain and interface errors.

	h_{coup}	$\varepsilon_{\text{DF},\Omega_2}$	$\varepsilon_{\text{R},\Omega_2;\text{NC}}$	$\varepsilon_{\text{R},\Omega_2;\text{LC}}$	$\varepsilon_{\text{DF},\Omega_1}$	$\varepsilon_{\text{R},\Omega_1;\text{NC}}$	$\varepsilon_{\text{R},\Omega_1;\text{LC}}$	$\varepsilon_{\text{DF},\Gamma_1}$	$\varepsilon_{\text{DF},\Gamma_2}$
RT0-P0	0.2625	2.35e-01	4.73e-02	2.55e-02	4.67e-03	1.56e-02	6.70e-03	5.05e-03	5.00e-03
	0.1720	1.62e-01	3.08e-02	1.10e-02	4.53e-03	9.01e-03	2.04e-03	1.37e-03	1.37e-03
	0.0827	8.69e-02	1.91e-02	3.91e-03	2.72e-03	5.17e-03	6.70e-04	4.07e-04	4.09e-04
	0.0418	4.58e-02	1.01e-02	1.06e-03	1.40e-03	2.64e-03	1.70e-04	1.08e-04	1.09e-04
MVEM-P0	0.2625	2.39e-01	4.73e-02	2.55e-02	5.78e-03	1.56e-02	6.70e-03	5.54e-03	5.49e-03
	0.1720	1.64e-01	3.08e-02	1.10e-02	5.56e-03	9.01e-03	2.04e-03	1.50e-03	1.50e-03
	0.0827	8.79e-02	1.91e-02	3.91e-03	3.05e-03	5.17e-03	6.70e-04	4.40e-04	4.41e-04
	0.0418	4.61e-02	1.01e-02	1.06e-03	1.46e-03	2.64e-03	1.70e-04	1.13e-04	1.14e-04
MPFA	0.2625	2.40e-01	4.73e-02	2.55e-02	5.04e-03	1.56e-02	6.70e-03	5.93e-03	5.86e-03
	0.1720	1.66e-01	3.08e-02	1.10e-02	4.86e-03	9.01e-03	2.04e-03	1.56e-03	1.55e-03
	0.0827	8.89e-02	1.91e-02	3.91e-03	2.82e-03	5.17e-03	6.70e-04	4.61e-04	4.62e-04
	0.0418	4.65e-02	1.01e-02	1.06e-03	1.41e-03	2.64e-03	1.70e-04	1.14e-04	1.16e-04
TPFA	0.2625	3.34e-01	4.73e-02	2.55e-02	4.88e-03	1.56e-02	6.70e-03	6.04e-03	5.11e-03
	0.1720	2.63e-01	3.08e-02	1.10e-02	4.86e-03	9.01e-03	2.04e-03	1.35e-03	1.29e-03
	0.0827	2.02e-01	1.91e-02	3.91e-03	2.85e-03	5.17e-03	6.70e-04	4.50e-04	4.39e-04
	0.0418	1.97e-01	1.01e-02	1.06e-03	1.46e-03	2.64e-03	1.70e-04	1.02e-04	1.02e-04

Table 5: Error estimates for the two-dimensional application.

	Mesh	$\varepsilon_{\Omega^2;\text{EC}}$	$\varepsilon_{\Omega^1;\text{EC,C}}$	$\varepsilon_{\Omega^1;\text{EC,B}}$	$\varepsilon_{\Gamma^1,\text{C}}$	$\varepsilon_{\Gamma^1,\text{B}}$	ε_{Γ^0}	$\mathcal{M}^{\oplus}_{\mathfrak{p};\text{EC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\text{EC}}$
RT0-P0	Coarse	7.39e-01	2.93e-01	2.98e-04	3.13e+03	1.52e-01	2.24e+01	9.94e+02	1.99e+03
	Intermediate	5.95e-01	1.90e-01	2.77e-04	1.95e+03	1.00e-01	1.79e+01	6.20e+02	1.24e+03
	Fine	4.30e-01	1.07e-01	2.78e-04	9.79e+02	5.15e-02	1.22e+01	3.15e+02	6.30e+02
MVEM-P0	Coarse	7.29e-01	3.51e-01	1.44e-04	3.10e+03	1.46e-01	4.41e+01	9.84e+02	1.97e+03
	Intermediate	5.91e-01	2.23e-01	1.27e-04	1.94e+03	9.43e-02	3.14e+01	6.17e+02	1.23e+03
	Fine	4.28e-01	1.24e-01	1.18e-04	9.78e+02	4.80e-02	2.02e+01	3.15e+02	6.29e+02
MPFA	Coarse	7.39e-01	3.13e-01	1.72e-04	3.03e+03	1.43e-01	3.39e+01	9.63e+02	1.93e+03
	Intermediate	5.98e-01	2.01e-01	1.54e-04	1.89e+03	9.18e-02	2.55e+01	6.00e+02	1.20e+03
	Fine	4.33e-01	1.12e-01	1.46e-04	9.49e+02	4.71e-02	1.68e+01	3.05e+02	6.10e+02
TPFA	Coarse	7.52e-01	3.05e-01	1.76e-04	3.19e+03	1.48e-01	3.67e+01	1.01e03	2.02e+03
	Intermediate	6.08e-01	1.96e-01	1.51e-04	1.95e+03	9.41e-02	2.61e+01	6.12e+02	1.22e+03
	Fine	4.45e-01	1.09e-01	1.60e-04	1.00e+03	4.84e-02	1.86e+01	3.23e+02	6.46e+02

The results for $\mathcal{M}^{\oplus}_{\mathfrak{u};\text{EC}}$ are omitted since they are equal to $\mathcal{M}^{\oplus}_{\mathfrak{p};\text{EC}}$.

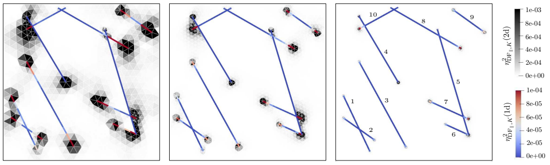


Figure 6: Two-dimensional benchmark problem and the errors associated with the matrix and fractures for the coarse (left), intermediate (center), and fine (right) grid resolutions. Fractures 4 and 5 are blocking, whereas the others are conductive. The local bounds were obtained using MPFA. The results suggest that subdomain diffusive errors are concentrated around fracture tips and fracture intersections.

coordinates can be found in Appendix C of [13]. Fractures 4 and 5 represent blocking fractures ($\mathcal{K} = 10^{-4}$ and $\kappa = 1$) whereas the others represent conductive fractures ($\mathcal{K} = 10^{4}$ and $\kappa = 10^{8}$). The matrix permeability is set to one. A linear pressure drop is imposed from left ($p = 4$) to right ($p = 1$), whereas no flux is prescribed at the top and bottom of the domain.

The benchmark establishes three refinement levels; coarse, intermediate, and fine, with approximately 1500, 4200, and 16000 two-dimensional cells. The structure of the local contributions to the majorant (confer e.g. equation (59)) are shown in Figure 6, based on the approximate solution obtained by the MPFA discretization.

In Table 5, we show the errors bounds for the three refinement levels. To avoid numbering domains and interfaces, we refer to the matrix error as $\varepsilon_{\Omega^2,\mathrm{EC}}$, and group the fracture and interface errors by conductive and blocking. For example, $\varepsilon_{\Omega^1,\mathrm{C},\mathrm{EC}}$ refers to the sum of the errors of 1d conductive fractures.

An important observation is that the persistent reduction of the majorant $\mathcal{M}^{\oplus}_{\mathbf{p},\mathbf{u};\mathrm{EC}}$, together with the known upper and lower bounds on the efficiency indexes established in Remark 16, provides a *post factum* verification of the convergence of all the numerical methods considered.

The error estimates suggest that the contribution to the overall error bounds are concentrated, primarily, on highly conductive interfaces (see the column corresponding to $\varepsilon_{\Gamma^1,\mathrm{C}}$). On a more qualitative note, Figure 6 suggests that subdomain diffusive errors are concentrated at the fracture tips and fracture intersections, which is where singularities may typically be encountered [12].

8.2 Three-dimensional application

Our last numerical application is based on a modified version of the three-dimensional benchmark 2.1 from [14]. The domain consists of nine intersecting fractures embedded in a unit cube, as shown in the middle panel of Figure 1.

Table 6: Error estimates for the three-dimensional application.

	Mesh	$\varepsilon_{\Omega^3;\mathrm{EC}}$	$\varepsilon_{\Omega^2;\mathrm{EC}}$	$\varepsilon_{\Omega^1;\mathrm{EC}}$	ε_{Γ^2}	ε_{Γ^1}	ε_{Γ^0}	$\mathcal{M}^{\oplus}_{\mathfrak{p};\mathrm{EC}}$	$\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\mathrm{EC}}$
RT0-P0	Coarse	6.17e-01	5.81e-04	3.16e-04	9.87e+02	3.63e-02	3.31e-02	5.03e+02	1.01e+03
	Intermediate	4.55e-01	4.61e-04	1.58e-04	7.75e+01	8.86e-03	8.35e-04	3.40e+01	6.81e+01
	Fine	3.86e-01	2.55e-04	9.60e-05	2.26e+01	4.63e-03	4.34e-04	1.07e+01	2.14e+01
MVEM-P0	Coarse	6.07e-01	6.99e-04	2.77e-04	9.54e+02	7.48e-02	6.38e-02	4.66e+02	9.33e+02
	Intermediate	4.55e-01	4.63e-04	1.65e-04	8.19e+01	9.96e-03	4.59e-03	3.59e+01	7.18e+01
	Fine	3.86e-01	2.46e-04	9.17e-05	2.33e+01	4.00e-03	1.75e-03	1.11e+01	2.22e+01
MPFA	Coarse	6.07e-01	7.00e-04	3.15e-04	1.05e+03	4.61e-02	1.69e-02	5.24e+02	1.05e+03
	Intermediate	4.46e-01	4.88e-04	1.61e-04	8.42e+01	7.72e-03	2.31e-03	3.71e+01	7.42e+01
	Fine	3.77e-01	2.53e-04	9.04e-05	2.37e+01	2.82e-03	9.36e-04	1.12e+01	2.24e+01
TPFA	Coarse	6.32e-01	4.72e-04	2.26e-04	7.92e+02	4.21e-02	1.34e-02	3.76e+02	7.52e+02
	Intermediate	4.48e-01	6.27e-04	1.40e-04	1.47e+02	1.56e-02	2.32e-03	6.82e+01	1.36e+02
	Fine	4.07e-01	5.82e-04	8.72e-05	4.60e+01	7.97e-03	1.05e-03	2.04e+01	4.08e+01

The results for $\mathcal{M}^{\oplus}_{\mathfrak{u};\mathrm{EC}}$ are omitted since they are equal to $\mathcal{M}^{\oplus}_{\mathfrak{p};\mathrm{EC}}$.

This results in an intricate network with 106 subdomains and 270 interfaces of different dimensionality.

The original benchmark imposes an inlet flux (purple lower corner $u = -1$) and an outlet pressure (pink upper corner $p = 1$), and for the rest of the external boundaries null flux. Since we have only detailed our results for zero Neumann boundary conditions, we have replaced the inlet flux by a constant pressure condition ($p = 1$) and modified the value of the outlet pressure ($p = 0$). The benchmark assigns heterogeneous permeability to the matrix subdomain, whereas the fractures are assumed to be highly conductive. For the complete description of the benchmark, we refer to [14], and for an impression on how the contributions to the majorant are distributed, see Figure 7. Here we show the error estimates for the whole fracture network obtained with RT0-P0, where it becomes evident that the subdomain diffusive errors are concentrated at the inlet and outlet boundaries; refinement efforts should therefore focus on these regions.

As in Section 8.1, we collect the local errors of subdomains and interfaces of equal dimensionality. The results are summarized in Table 6. As in the previous cases, we have local and global convergence for all four numerical methods. Again, RT0-P0, MVEM-P0, and MPFA show very similar results, while TPFA show larger errors.

As in the 2d case discussed above, the persistent reduction of the majorant $\mathcal{M}^{\oplus}_{\mathfrak{p},\mathfrak{u};\mathrm{EC}}$, again serves as a verification of the convergence of all four numerical methods.

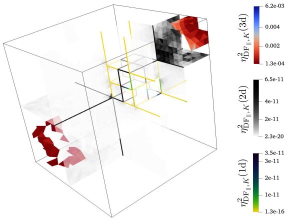

$\eta^2_{\mathrm{DF}_{\|},K}(3\mathrm{d})$
6.2e-03
0.004
0.002
1.3e-04
$\eta^2_{\mathrm{DF}_{\|},K}(2\mathrm{d})$
6.5e-11
4e-11
2e-11
2.3e-20
$\eta^2_{\mathrm{DF}_{\|},K}(1\mathrm{d})$
3.5e-11
3e-11
2e-11
1e-11
1.3e-16

Figure 7: Subdomain diffusive error contributions to the majorant for the fine grid resolution obtained with RT0-P0.

9 Conclusion

In this paper, we obtained *a posteriori* error estimates for mixed-dimensional elliptic equations. Depending upon the level of accuracy at which residual balances can be approximated, we have derived four concrete versions of the majorant; i.e.: for no mass-conservative, subdomain mass-conservative, locally mass-conservative, and point-wise mass-conservative approximations. Furthermore, we have demonstrated both theoretically and numerically that sharper bounds can be obtained (for locally mass-conservative methods) using local Poincaré constants instead of the global ones.

Our bounds have been thoroughly tested with numerical approximations obtained with four locally mass-conservative methods of the lowest-order, namely: RT0-P0, MVEM-P0, MPFA, and TPFA. We performed a detailed efficiency analysis comparing the use of global and local Poincaré-Friedrichs constants in two and three dimensions. In both validations, our upper bounds reflected the optimal convergence rates of the numerical methods. In addition, we applied our bounds to two- and three-dimensional community benchmark problems exhibiting challenging fracture networks. Again, in both cases, the bounds reflected the limitations and the convergence rates of the methods satisfactory.

To the best of our knowledge, the bounds obtained here are the first of their kind to provide a practical tool to measure the error in numerical approximations to the equations modeling the incompressible, single-phase flow in generic fractured porous media.

Funding: *Jhabriel Varela was funded by VISTA – a basic research program in collaboration between The Norwegian Academy of Science and Letters, and Equinor. Additionally, this work was supported in part through the Norwegian Research Council grant 250223. The authors would like to thank W. M. Boon for helpful discussions on this topic.*

References

[1] J. M. Nordbotten. Mixed-dimensional models for real-world applications. *Snapshots of Modern Mathematics from Oberwolfach*, page 11, 2019. doi: 10.14760/SNAP-2019-014-EN.

[2] S. S. Antman. *Nonlinear problems of elasticity*, volume 107 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1995. ISBN 0-387-94199-1. doi: 10.1007/978-1-4757-4147-6.

[3] P. G. Ciarlet. *Mathematical elasticity. Vol. II*, volume 27 of *Studies in Mathematics and its Applications*. North-Holland Publishing Co., Amsterdam, 1997. ISBN 0-444-82570-3. Theory of plates.

[4] W. M. Boon and J. M. Nordbotten. Stable mixed finite elements for linear elasticity with thin inclusions. *Computational Geosciences*, pages 603–620, 2021. doi: 10.1007/s10596-020-10013-2.

[5] C. D'Angelo and A. Quarteroni. On the coupling of 1D and 3D diffusion-reaction equations. Application to tissue perfusion problems. *Math. Models Methods Appl. Sci.*, 18(8):1481–1504, 2008. ISSN 0218-2025. doi: 10.1142/S0218202508003108.

[6] T. Köppl, E. Vidotto, and B. Wohlmuth. A local error estimate for the Poisson equation with a line source term. In *Numerical mathematics and advanced applications—ENUMATH 2015*, volume 112 of *Lect. Notes Comput. Sci. Eng.*, pages 421–429. Springer, Cham, 2016. doi: 10.1007/978-3-319-39929-4_40.

[7] E. Hodneland, E. Hanson, O. Sævareid, G. Nævdal, A. Lundervold, V. Šoltészová, A. Z. Munthe-Kaas, A. Deistung, J. R. Reichenbach, and J. M. Nordbotten. A new framework for assessing subject-specific whole brain circulation and perfusion using MRI-based measurements and a multi-scale continuous flow model. *PLoS computational biology*, 15(6): e1007073, 2019. doi: 10.1371/journal.pcbi.1007073.

[8] T. Koch, K. Heck, N. Schröder, H. Class, and R. Helmig. A new simulation framework for soil–root interaction, evaporation, root growth, and solute transport. *Vadose zone journal*, 17(1):1–21, 2018. doi: https://doi.org/10.2136/vzj2017.12.0210.

[9] C. Alboin, J. Jaffré, J. E. Roberts, and C. Serres. Modeling fractures as interfaces for flow and transport in porous media. In *Fluid flow and transport in porous media: mathematical and numerical treatment (South Hadley, MA, 2001)*, volume 295 of *Contemp. Math.*, pages 13–24. Amer. Math. Soc., Providence, RI, 2002. doi: 10.1090/conm/295/04999.

[10] L. Formaggia, A. Fumagalli, A. Scotti, and P. Ruffo. A reduced model for Darcy's problem in networks of fractures. *ESAIM Math. Model. Numer. Anal.*, 48(4):1089–1116, 2014. ISSN 0764-583X. doi: 10.1051/m2an/2013132.

[11] E. Ahmed, J. Jaffré, and J. E. Roberts. A reduced fracture model for two-phase flow with different rock types. *Math. Comput. Simulation*, 137: 49–70, 2017. ISSN 0378-4754. doi: 10.1016/j.matcom.2016.10.005.

[12] W. M. Boon, J. M. Nordbotten, and I. Yotov. Robust discretization of flow in fractured porous media. *SIAM J. Numer. Anal.*, 56(4):2203–2233, 2018. ISSN 0036-1429. doi: 10.1137/17M1139102.

[13] B. Flemisch, I. Berre, W. Boon, A. Fumagalli, N. Schwenck, A. Scotti, I. Stefansson, and A. Tatomir. Benchmarks for single-phase flow in fractured porous media. *Advances in Water Resources*, 111:239–258, 2018. doi: 10.1016/j.advwatres.2017.10.036.

[14] I. Berre, W. M. Boon, B. Flemisch, A. Fumagalli, D. Gläser, E. Keilegavlen, A. Scotti, I. Stefansson, A. Tatomir, K. Brenner, S. Burbulla, P. Devloo, O. Duran, M. Favino, J. Hennicker, I-H. Lee, K. Lipnikov, R. Masson, K. Mosthaf, M. G. C. Nestola, C-F. Ni, K. Nikitin, P. Schädle, D. Svyatskiy, R. Yanbarisov, and P. Zulian. Verification benchmarks for single-phase flow in three-dimensional fractured porous media. *Advances in Water Resources*, 147:103759, 2021. ISSN 0309-1708. doi: 10.1016/j.advwatres.2020.103759.

[15] W. M. Boon, J. M Nordbotten, and J. E. Vatne. Functional analysis and exterior calculus on mixed-dimensional geometries. *Annali di Matematica Pura ed Applicata (1923-)*, page 757–789, 2021. doi: 10.1007/s10231-020-01013-1.

[16] S. I. Repin. A posteriori error estimation for variational problems with uniformly convex functionals. *Math. Comp.*, 69(230):481–500, 2000. ISSN 0025-5718. doi: 10.1090/S0025-5718-99-01190-4.

[17] S. I. Repin. Two-sided estimates of deviation from exact solutions of uniformly elliptic equations. In *Proceedings of the St. Petersburg Mathematical Society, Vol. IX*, volume 209 of *Amer. Math. Soc. Transl. Ser. 2*, pages 143–171, Providence, RI, 2003. Amer. Math. Soc. doi: 10.1090/trans2/209/06.

[18] P. Neittaanmäki and S. I. Repin. *Reliable methods for computer simulation*, volume 33 of *Studies in Mathematics and its Applications.* Elsevier Science B.V, Amsterdam, 2004. ISBN 0-444-51376-0. Error control and a posteriori estimates.

[19] S. I. Repin, S. Sauter, and A. Smolianski. Two-sided a posteriori error estimates for mixed formulations of elliptic problems. *SIAM J. Numer. Anal.*, 45(3):928–945, 2007. ISSN 0036-1429. doi: 10.1137/050641533.

[20] S. I. Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin, 2008. ISBN 978-3-11-019153-0. doi: 10.1515/9783110203042.

[21] D. Pauly. Solution theory, variational formulations, and functional a posteriori error estimates for general first order systems with applications to electro-magneto-statics and more. *Numer. Funct. Anal. Optim.*, 41(1):16–112, 2020. ISSN 0163-0563. doi: 10.1080/01630563.2018.1490756.

[22] S. Kurz, D. Pauly, D. Praetorius, S. I. Repin, and D. Sebastian. Functional a posteriori error estimates for boundary element methods. *Numer. Math.*, 147(4):937–966, 2021. ISSN 0029-599X. doi: 10.1007/s00211-021-01188-6.

[23] R. Verfürth. A review of a posteriori error estimation techniques for elasticity problems. *Comput. Methods Appl. Mech. Engrg.*, 176(1-4):419–440, 1999. ISSN 0045-7825. doi: 10.1016/S0045-7825(98)00347-8. New advances in computational methods (Cachan, 1997).

[24] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Numer. Methods Engrg.*, 24(2):337–357, 1987. ISSN 0029-5981. doi: 10.1002/nme.1620240206.

[25] J. T. Oden and S. Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Comput. Math. Appl.*, 41(5-6):735–756, 2001. ISSN 0898-1221. doi: 10.1016/S0898-1221(00)00317-5.

[26] M. Vohralík. Unified primal formulation-based a priori and a posteriori error analysis of mixed finite element methods. *Math. Comp.*, 79(272): 2001–2032, 2010. ISSN 0025-5718. doi: 10.1090/S0025-5718-2010-02375-0.

[27] M. Ainsworth. A posteriori error estimation for lowest order Raviart-Thomas mixed finite elements. *SIAM J. Sci. Comput.*, 30(1):189–204, 2007/08. ISSN 1064-8275. doi: 10.1137/06067331X.

[28] C. Pechstein and R. Scheichl. Weighted Poincaré inequalities. *IMA J. Numer. Anal.*, 33(2):652–686, 2013. ISSN 0272-4979. doi: 10.1093/imanum/drs017.

[29] M. Rathmair. On how Poincaré inequalities imply weighted ones. *Monatsh. Math.*, 188(4):753–763, 2019. ISSN 0026-9255. doi: 10.1007/s00605-019-01266-w.

[30] J. M. Nordbotten, W. M. Boon, A. Fumagalli, and E. Keilegavlen. Unified approach to discretization of flow in fractured porous media. *Comput. Geosci.*, 23(2):225–237, 2019. ISSN 1420-0597. doi: 10.1007/s10596-018-9778-9.

[31] B. I. Wohlmuth. Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers. *SIAM J. Numer. Anal.*, 36 (5):1636–1658, 1999. ISSN 0036-1429. doi: 10.1137/S0036142997330512.

[32] Z. Belhachmi. A posteriori error estimates for the 3D stabilized mortar finite element method applied to the Laplace equation. *M2AN Math. Model. Numer. Anal.*, 37(6):991–1011, 2003. ISSN 0764-583X. doi: 10.1051/m2an:2003064.

[33] M. F. Wheeler and I. Yotov. A posteriori error estimates for the mortar mixed finite element method. *SIAM J. Numer. Anal.*, 43(3):1021–1042, 2005. ISSN 0036-1429. doi: 10.1137/S0036142903431687.

[34] G. V. Pencheva, M. Vohralík, M. F. Wheeler, and T. Wildey. Robust a posteriori error control and adaptivity for multiscale, multinumerics, and mortar coupling. *SIAM J. Numer. Anal.*, 51(1):526–554, 2013. ISSN 0036-1429. doi: 10.1137/110839047.

[35] H. Chen and S. Sun. A residual-based a posteriori error estimator for single-phase Darcy flow in fractured porous media. *Numer. Math.*, 136(3): 805–839, 2017. ISSN 0029-599X. doi: 10.1007/s00211-016-0851-9.

[36] Z. Mghazli and I. Naji. Guaranteed *a posteriori* error estimates for a fractured porous medium. *Math. Comput. Simulation*, 164:163–179, 2019. ISSN 0378-4754. doi: 10.1016/j.matcom.2019.02.002.

[37] F. Hecht, Z. Mghazli, I. Naji, and J. E. Roberts. A residual *a posteriori* error estimators for a model for flow in porous media with fractures. *J. Sci. Comput.*, 79(2):935–968, 2019. ISSN 0885-7474. doi: 10.1007/s10915-018-0875-7.

[38] V. Martin, J. Jaffré, and J. E. Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691, 2005. ISSN 1064-8275. doi: 10.1137/S1064827503429363.

[39] E. Keilegavlen, R. Berge, A. Fumagalli, M. Starnoni, I. Stefansson, J. Varela, and I. Berre. Porepy: An open-source software for simulation of multiphysics processes in fractured porous media. *Computational Geosciences*, 25(1):243–265, 2021. doi: 10.1007/s10596-020-10002-5.

[40] R. A. Adams and J. J. F. Fournier. *Sobolev spaces*, volume 140 of *Pure and Applied Mathematics (Amsterdam)*. Elsevier/Academic Press, Amsterdam, second edition, 2003. ISBN 0-12-044143-8.

[41] G. K. Pedersen. *Analysis now*, volume 118 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1989. ISBN 0-387-96788-5. doi: 10.1007/978-1-4612-1007-8.

[42] D. N. Arnold. *Finite element exterior calculus*, volume 93 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2018. ISBN 978-1-611975-53-6. doi: 10.1137/1.9781611975543.ch1.

[43] C. Carstensen and S. A. Funken. Constants in Clément-interpolation error and residual based a posteriori estimates in finite element methods. *East-West J. Numer. Math*, 8(3):153–175, 2000.

[44] S. I. Repin. Computable majorants of constants in the Poincaré and Friedrichs inequalities. *J. Math. Sci. (N.Y.)*, 186(2):307–321, 2012. ISSN 1072-3374. doi: 10.1007/s10958-012-0987-9. Problems in mathematical analysis. No. 66.

[45] M. Vohralík. On the discrete Poincaré-Friedrichs inequalities for nonconforming approximations of the Sobolev space H^1. *Numer. Funct. Anal. Optim.*, 26(7-8):925–952, 2005. ISSN 0163-0563. doi: 10.1080/01630560500444533.

[46] L. E. Payne and H. F. Weinberger. An optimal Poincaré inequality for convex domains. *Arch. Rational Mech. Anal.*, 5:286–292 (1960), 1960. ISSN 0003-9527. doi: 10.1007/BF00252910.

[47] M. Bebendorf. A note on the Poincaré inequality for convex domains. *Z. Anal. Anwendungen*, 22(4):751–756, 2003. ISSN 0232-2064. doi: 10.4171/ZAA/1170.

[48] A. Ern and M. Vohralík. Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations. *SIAM J. Numer. Anal.*, 53(2):1058–1081, 2015. ISSN 0036-1429. doi: 10.1137/130950100.

[49] A. Ern, I. Smears, and M. Vohralík. Guaranteed, locally space-time efficient, and polynomial-degree robust a posteriori error estimates for high-order discretizations of parabolic problems. *SIAM J. Numer. Anal.*, 55(6): 2811–2834, 2017. ISSN 0036-1429. doi: 10.1137/16M1097626.

[50] I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15(4):736–754, 1978. ISSN 0036-1429. doi: 10.1137/0715049.

[51] D. W. Kelly, J. P. de S. R. Gago, O. C. Zienkiewicz, and I. Babuška. A posteriori error analysis and adaptive processes in the finite element method. I. Error analysis. *Internat. J. Numer. Methods Engrg.*, 19(11): 1593–1619, 1983. ISSN 0029-5981. doi: 10.1002/nme.1620191103.

[52] P.-A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975)*, pages 292–315. Lecture Notes in Math., Vol. 606, 1977.

[53] J.-C. Nédélec. Mixed finite elements in $\mathbf{R}^3$. *Numer. Math.*, 35(3):315–341, 1980. ISSN 0029-599X. doi: 10.1007/BF01396415.

[54] W. M. Boon, D. Gläser, R. Helmig, and I. Yotov. Flux-mortar mixed finite element methods on non-matching grids, 2020.

[55] 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(4):1295–1315, 2000. doi: 10.1137/S0036142996308447.

[56] S. Cochez-Dhondt, S. Nicaise, and S. I. Repin. A posteriori error estimates for finite volume approximations. *Math. Model. Nat. Phenom.*, 4(1):106–122, 2009. ISSN 0973-5348. doi: 10.1051/mmnp/20094105.

[57] A. Ern and M. Vohralík. A posteriori error estimation based on potential and flux reconstruction for the heat equation. *SIAM J. Numer. Anal.*, 48 (1):198–223, 2010. ISSN 0036-1429. doi: 10.1137/090759008.

[58] E. Ahmed, F. A. Radu, and J. M. Nordbotten. Adaptive poromechanics computations based on a posteriori error estimates for fully mixed formulations of Biot's consolidation model. *Comput. Methods Appl. Mech. Engrg.*, 347:264–294, 2019. ISSN 0045-7825. doi: 10.1016/j.cma.2018.12.016.

[59] E. Ahmed, J. M. Nordbotten, and F. A. Radu. Adaptive asynchronous time-stepping, stopping criteria, and a posteriori error estimates for fixed-stress iterative schemes for coupled poromechanics problems. *J. Comput. Appl. Math.*, 364:112312, 25, 2020. ISSN 0377-0427. doi: 10.1016/j.cam.2019.06.028.

[60] L. B da Veiga, F. Brezzi, L. D. Marini, and A. Russo. Mixed virtual element methods for general second order elliptic problems on polygonal meshes. *ESAIM Math. Model. Numer. Anal.*, 50(3):727–747, 2016. ISSN 0764-583X. doi: 10.1051/m2an/2015067.

[61] A. Fumagalli and E. Keilegavlen. Dual virtual element methods for discrete fracture matrix models. *Oil & Gas Science and Technology–Revue d'IFP Energies nouvelles*, 74:41, 2019. doi: 10.2516/ogst/2019008.

[62] I. Aavatsmark. An introduction to multipoint flux approximations for quadrilateral grids. *Comput. Geosci.*, 6(3-4):405–432, 2002. ISSN 1420-0597. doi: 10.1023/A:1021291114475. Locally conservative numerical methods for flow in porous media.

[63] J. M. Nordbotten and E. Keilegavlen. An introduction to multi-point flux (MPFA) and stress (MPSA) finite volume methods for thermo-poroelasticity. In *Polyhedral Methods in Geosciences*, pages 119–158. Springer, 2021.

[64] D. Boffi, F. Brezzi, and M. Fortin. *Mixed finite element methods and applications*, volume 44 of *Springer Series in Computational Mathematics*. Springer, Heidelberg, 2013. ISBN 978-3-642-36518-8; 978-3-642-36519-5. doi: 10.1007/978-3-642-36519-5.

[65] J. Varela. jhabriel/mixdim-estimates: v1.4, March 2022. URL https://doi.org/10.5281/zenodo.6383202.

[66] D. Pauly and J. Valdman. Poincaré-Friedrichs type constants for operators involving grad, curl, and div: theory and numerical experiments. *Comput. Math. Appl.*, 79(11):3027–3067, 2020. ISSN 0898-1221. doi: 10.1016/j.camwa.2020.01.004.

A Derivation of variational formulations

Here, we present the derivations for the primal and dual variational formulations for the case of a single fracture immersed in a matrix.

A.1 Derivation of the primal weak form for a single fracture

Substitute (1b) into (1a), multiply each term by $q_2 \in H_0^1(\Omega_2)$, and integrate over Ω_2. Similarly, substitute (1b), (3a), and (3b) into (2a), multiply each term by $q_1 \in H_0^1(\Omega_1)$ and integrate over Ω_1. Add the resulting equations to obtain

$$\begin{aligned}&- \langle \nabla_2 \cdot \mathcal{K}_2 \nabla_2 \, p_2, q_2 \rangle_{\Omega_2} - \langle \nabla_1 \cdot \mathcal{K}_1 \nabla_1 \, p_1, q_1 \rangle_{\Omega_1} + \langle \kappa_1 \, (p_1 - \mathrm{tr}_{\partial_1 \Omega_2} \, p_2), q_1 \rangle_{\Omega_1} \\ &\quad + \langle \kappa_2 \, (p_1 - \mathrm{tr}_{\partial_2 \Omega_2} \, p_2), q_1 \rangle_{\Omega_1} = \langle f_2, q_2 \rangle_{\Omega_2} + \langle f_1, q_1 \rangle_{\Omega_1}. \end{aligned} \tag{77}$$

Using integration by parts, the first term of (77) can be expressed as

$$\begin{aligned}&- \langle \nabla_2 \cdot \mathcal{K}_2 \nabla_2 \, p_2, q_2 \rangle_{\Omega_2} \\ &\quad = \langle \mathcal{K}_2 \nabla_2 \, p_2, \nabla_2 \, q_2 \rangle_{\Omega_2} - \sum_{j=1}^{2} \left\langle \mathrm{tr}_{\partial_j \Omega_2} \, (\mathcal{K}_2 \nabla_2 \, p_2) \cdot n_2, \mathrm{tr}_{\partial_j \Omega_2} \, q_2 \right\rangle_{\partial_j \Omega_2}, \\ &\quad = \langle \mathcal{K}_2 \nabla_2 p_2, \nabla_2 q_2 \rangle_{\Omega_2} - \sum_{j=1}^{2} \left\langle \lambda_j, \mathrm{tr}_{\partial_j \Omega_2} \, q_2 \right\rangle_{\Gamma_j}, \\ &\quad = \langle \mathcal{K}_2 \nabla_2 p_2, \nabla_2 q_2 \rangle_{\Omega_2} + \sum_{j=1}^{2} \left\langle \kappa_j \left(p_1 - \mathrm{tr}_{\partial_j \Omega_2} \, p_2 \right), \mathrm{tr}_{\partial_j \Omega_2} \, q_2 \right\rangle_{\Gamma_j}. \end{aligned} \tag{78}$$

Here, we use the internal boundary conditions (1c) and (1d) and the definition of the mortar fluxes (3a) and (3b). Analogously, integration by parts allows us to write the second term of (77) as

$$- \langle \nabla_1 \cdot \mathcal{K}_1 \nabla_1 \, p_1, q_1 \rangle_{\Omega_1} = \langle \mathcal{K}_1 \nabla_1 \, p_1, \nabla_1 \, q_1 \rangle_{\Omega_1}. \tag{79}$$

Note that the boundary terms vanish due to the choice of boundary conditions.

Finally, we note that the third and fourth terms from (77) can be equivalently written as

$$\left\langle \kappa_j \left(p_1 - \mathrm{tr}_{\partial_j \Omega_2} \, p_2 \right), q_1 \right\rangle_{\Omega_1} = \left\langle \kappa_j \left(p_1 - \mathrm{tr}_{\partial_j \Omega_2} \, p_2 \right), q_1 \right\rangle_{\Gamma_j}, \qquad j \in \{1, 2\}. \tag{80}$$

The proof is completed by substituting (78), (79), and (80) into (77) and grouping common terms.

A.2 Derivation of the dual weak form for a single fracture

Let us start with (13a). Multiply respectively (1b) and (2b) by $v_{0,2} \in V_{0,2}$ and $v_{0,1} \in V_{0,1}$, integrate over the subdomains Ω_2 and Ω_1, use integration by parts to obtain

$$\begin{aligned}\left\langle \mathcal{K}_2^{-1} u_2, v_{0,2}\right\rangle_{\Omega_2} &= \left\langle \mathcal{K}_2^{-1}\left(u_{0,2} + \mathcal{R}_1\lambda_1 + \mathcal{R}_2\lambda_2\right), v_{0,2}\right\rangle_{\Omega_2} = -\langle \nabla_2\, p_2, v_{0,2}\rangle_{\Omega_2} \\ &= \langle p_2, \nabla_2 \cdot v_{0,2}\rangle_{\Omega_2} - \langle g_{D,2}, \operatorname{tr}_{\partial_D \Omega_2} v_{0,2} \cdot n_2\rangle_{\partial_D\Omega_2}. \qquad (81) \\ \left\langle \mathcal{K}_1^{-1} u_1, v_{0,1}\right\rangle_{\Omega_1} &= \left\langle \mathcal{K}_1^{-1} u_{0,1}, v_{0,1}\right\rangle_{\Omega_1} = -\langle \nabla_1\, p_1, v_{0,1}\rangle_{\Omega_1} \\ &= \langle p_1, \nabla_1 \cdot v_{0,1}\rangle_{\Omega_1} - \langle g_{D,1}, \operatorname{tr}_{\partial_D \Omega_1} v_{0,1} \cdot n_1\rangle_{\partial_D\Omega_1}. \qquad (82)\end{aligned}$$

Adding together (81) and (82) gives (13a). We now focus on (13b). First, we use (1b) and multiply by the test functions $\mathcal{R}_j\nu_j$ with $\nu_j \in L^2(\Gamma_j)$ for $j \in \{1, 2\}$, integrate over Ω_2, and apply integration by parts, to obtain:

$$\begin{aligned}\left\langle \mathcal{K}_2^{-1} u_2, \mathcal{R}_j\nu_j\right\rangle_{\Omega_2} &= \left\langle \mathcal{K}_2^{-1}\left(u_{0,2} + \mathcal{R}_1\lambda_1 + \mathcal{R}_2\lambda_2\right), \mathcal{R}_j\nu_j\right\rangle_{\Omega_2} = -\langle \nabla_2\, p_2, \mathcal{R}_j\nu_j\rangle_{\Omega_2} \\ &= \langle p_2, \nabla_2 \cdot (\mathcal{R}_j\nu_j)\rangle_{\Omega_2} - \left\langle \operatorname{tr}_{\partial_j\Omega_2} p_2, \operatorname{tr}_{\partial_j\Omega_2}\left(\mathcal{R}_j\nu_j\right)\cdot n_2\right\rangle_{\partial_j\Omega_2} \\ &= \langle p_2, \nabla_2 \cdot (\mathcal{R}_j\nu_j)\rangle_{\Omega_2} - \left\langle \operatorname{tr}_{\partial_j\Omega_2} p_2, \nu_j\right\rangle_{\partial_j\Omega_2} \\ &= \langle p_2, \nabla_2 \cdot (\mathcal{R}_j\nu_j)\rangle_{\Omega_2} - \left\langle \operatorname{tr}_{\partial_j\Omega_2} p_2, \nu_j\right\rangle_{\Gamma_j}. \qquad (83)\end{aligned}$$

Next, we multiply the interface laws (3a) and (3b) by ν_1 and ν_2, respectively, to get for $j = \{1, 2\}$

$$\left\langle \kappa_1^{-1}\lambda_j, \nu_j\right\rangle_{\Gamma_j} = -\langle p_1, \nu_j\rangle_{\Gamma_j} + \left\langle \operatorname{tr}_{\partial_j\Omega_2} p_2, \nu_j\right\rangle_{\Gamma_j} = -\langle p_1, \nu_j\rangle_{\Omega_1} + \left\langle \operatorname{tr}_{\partial_j\Omega_2} p_2, \nu_j\right\rangle_{\Gamma_j}. \qquad (84)$$

After adding (83) and (84) and canceling common terms, we obtain (13b). Finally, to obtain (13c), we multiply (1a) by $q_2 \in L^2(\Omega_2)$ and (2a) by $q_1 \in L^2(\Omega_1)$, and integrate over their respective subdomains, and add the resulting equations.

B Proof of Theorem 1

Here, we present the proof of the upper bound of the error for the primal variable, for the case of a single fracture immersed in a matrix.

Proof. Start by computing the difference between $p = [p_1, p_2] \in H_0^1(\Omega) + g$ and an arbitrary function $q = [q_1, q_2] \in H_0^1(\Omega) + g$ in the energy norm (14):

$$\begin{aligned}\left|\!\left|\!\left| p - q \right|\!\right|\!\right|^2 &= \langle \mathcal{K}_2\nabla_2(p_2 - q_2), \nabla_2(p_2 - q_2)\rangle_{\Omega_2} + \langle \mathcal{K}_1\nabla_1(p_1 - q_1), \nabla_1(p_1 - q_1)\rangle_{\Omega_1} \\ &\quad + \sum_{j=1}^{2}\left\langle \kappa_j\left[(p_1 - q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2 - q_2)\right], (p_1 - q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2 - q_2)\right\rangle_{\Gamma_j}, \\ &= \langle \mathcal{K}_2\nabla_2 p_2, \nabla_2(p_2 - q_2)\rangle_{\Omega_2} + \langle \mathcal{K}_1\nabla_1 p_1, \nabla_1(p_1 - q_1)\rangle_{\Omega_1} \\ &\quad + \sum_{j=1}^{2}\left\langle \kappa_j\left[(p_1 - q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2 - q_2)\right], (p_1 - q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2 - q_2)\right\rangle_{\Gamma_j} \\ &\quad + \langle -\mathcal{K}_2\nabla_2 q_2, \nabla_2(p_2 - q_2)\rangle_{\Omega_2} + \langle -\mathcal{K}_1\nabla_1 q_1, \nabla_1(p_1 - q_1)\rangle_{\Omega_1} \\ &\quad + \sum_{j=1}^{2}\left\langle -\kappa_j\left(q_1 - \operatorname{tr}_{\partial_j\Omega_2} q_2\right), (p_1 - q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2 - q_2)\right\rangle_{\Gamma_j}. \qquad (85)\end{aligned}$$

By noticing that the first three terms of (85) add up to the right-hand side of (6), and adding the identity

$$
\begin{aligned}
&-\langle v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2, \nabla_2(p_2-q_2)\rangle_{\Omega_2} - \langle v_{0,1}, \nabla_1(p_1-q_1)\rangle_{\Omega_1} \\
&\quad + \langle \nabla_2\cdot(v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2), p_2-q_2\rangle_{\Omega_2} + \langle \nabla_1\cdot v_{0,1}-\nu_1-\nu_2, p_1-q_1\rangle_{\Omega_1} \\
&\quad + \sum_{j=1}^{2} \left\langle \nu_j, (p_1-q_1) - \operatorname{tr}_{\partial_j\Omega_2}(p_2-q_2)\right\rangle_{\Gamma_j} = 0,
\end{aligned}
$$

valid for any $v_0 \in V_0$ and $\nu \in L^2(\Gamma)$ to (85), we obtain

$$
\begin{aligned}
|||p-q|||^2 = &\ \langle -(v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2+\mathcal{K}_2\nabla_2 q_2), \nabla_2(p_2-q_2)\rangle_{\Omega_2} \\
&+ \langle -(v_{0,1}+\mathcal{K}_1\nabla_1 p_1), \nabla_1(p_1-q_1)\rangle_{\Omega_1} \\
&+ \sum_{j=1}^{2} \left\langle -\left[\nu_j+\kappa_j\left(q_1-\operatorname{tr}_{\partial_j\Omega_2} q_2\right)\right], (p_1-q_1)-\operatorname{tr}_{\partial_j\Omega_2}(p_2-q_2)\right\rangle_{\Gamma_j} \\
&+ \langle f_2-\nabla_2\cdot(v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2), p_2-q_2\rangle_{\Omega_2} \\
&+ \langle f_1-\nabla_1\cdot v_{0,1}+\nu_1+\nu_2, p_1-q_1\rangle_{\Omega_1}. \qquad (86)
\end{aligned}
$$

Recognizing that since $\mathcal{K}_2$ is symmetric positive definite, it can be expressed as $\mathcal{K}_2 = \left(\mathcal{K}_2^{1/2}\right)^2$, where $\mathcal{K}_2^{1/2}$ is also symmetric positive definite, and therefore self-adjoint. The square-root of the material coefficients can therefore be moved to the second argument of the three first inner products in (86). After applying the Cauchy-Schwarz inequality to each inner product of (86), one gets

$$
\begin{aligned}
|||p-q|||^2 \le &\left\|\mathcal{K}_2^{-\frac{1}{2}}(v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2+\mathcal{K}_2\nabla_2 q_2)\right\|_{\Omega_2} \left\|\mathcal{K}_2^{\frac{1}{2}}\nabla_2(p_2-q_2)\right\|_{\Omega_2} \\
&+ \left\|\mathcal{K}_1^{-\frac{1}{2}}(v_{0,1}+\mathcal{K}_1\nabla_1 p_1)\right\|_{\Omega_1} \left\|\mathcal{K}_1^{\frac{1}{2}}\nabla_1(p_1-q_1)\right\|_{\Omega_1} \\
&+ \left\|\kappa_1^{-\frac{1}{2}}\left[\nu_1+\kappa_1\left(q_1-\operatorname{tr}_{\partial_1\Omega_2} q_2\right)\right]\right\|_{\Gamma_1} \left\|\kappa_1^{\frac{1}{2}}\left[(p_1-q_1)-\operatorname{tr}_{\partial_1\Omega_2}(p_2-q_2)\right]\right\|_{\Gamma_1} \\
&+ \left\|\kappa_2^{-\frac{1}{2}}\left[\nu_2+\kappa_2\left(q_1-\operatorname{tr}_{\partial_2\Omega_2} q_2\right)\right]\right\|_{\Gamma_2} \left\|\kappa_2^{\frac{1}{2}}\left[(p_1-q_1)-\operatorname{tr}_{\partial_2\Omega_2}(p_2-q_2)\right]\right\|_{\Gamma_2} \\
&+ \|f_2-\nabla_2\cdot(v_{0,2}+\mathcal{R}_1\nu_1+\mathcal{R}_2\nu_2)\|_{\Omega_2} \|p_2-q_2\|_{\Omega_2} \\
&+ \|f_1-\nabla_1\cdot v_{0,1}+\nu_1+\nu_2\|_{\Omega_1} \|p_1-q_1\|_{\Omega_1}
\end{aligned}
$$

Applying the permeability-weighted Poincaré-Friedrichs inequality (40b) to the terms $\|p_1-q_1\|_{\Omega_1}$ and $\|p_2-q_2\|_{\Omega_2}$, the proof of the theorem is completed. □

C Proof of Theorem 2

Here, we present the proof of our main theorem, which deals with the general abstract estimates in a mixed-dimensional setting.

Proof. (1) The proof for the bounds for the mD primal variable follows the one presented in Appendix B, modulo its generalization to the mD setting and the use of weighted norms on the residual terms. Start by computing the difference between any $\mathfrak{q} \in H_0^1(\Omega) + \mathfrak{g}$ and $\mathfrak{p} \in H_0^1(\Omega) + \mathfrak{g}$ using (49), to get

$$\begin{aligned}
\vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert^2 &= \langle \mathfrak{K}\,\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}), \mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} \\
&= \langle \mathfrak{K}\,\mathbb{D}\,\mathfrak{p}, \mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} + \langle -\mathfrak{K}\,\mathbb{D}\,\mathfrak{q}, \mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} \\
&= \langle \mathfrak{f}, \mathfrak{p} - \mathfrak{q} \rangle_{\Omega} + \langle -\mathfrak{K}\,\mathbb{D}\,\mathfrak{q}, \mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} \\
&= \langle \mathfrak{f}, \mathfrak{p} - \mathfrak{q} \rangle_{\Omega} + \left\langle -\mathfrak{K}^{-\frac{1}{2}}\mathbb{D}\,\mathfrak{q}, \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \right\rangle_{\Omega,\Gamma} \\
&= \langle \mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v}, \mathfrak{p} - \mathfrak{q} \rangle_{\Omega} + \left\langle -\mathfrak{K}^{-\frac{1}{2}}(\mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}), \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \right\rangle_{\Omega,\Gamma}.
\end{aligned} \tag{87}$$

Here, we used (49), (55), and added the fact that $\mathfrak{D}\cdot$ and $\mathbb{D}$ are adjoints.

By exploiting the orthogonality property (46) and then introducing the weights to the second and third terms, (87) can be equivalently written as:

$$\begin{aligned}
\vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert^2 &= \langle \mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v}, \pi_W(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega} + \left\langle -\mathfrak{K}^{-\frac{1}{2}}(\mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}), \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,(\mathfrak{p} - \mathfrak{q})) \right\rangle_{\Omega,\Gamma} \\
&= \left\langle \mu^{-1}(\mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v}), \mu\pi_W(\mathfrak{p} - \mathfrak{q}) \right\rangle_{\Omega} + \left\langle -\mathfrak{K}^{-\frac{1}{2}}(\mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}), \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \right\rangle_{\Omega,\Gamma}.
\end{aligned} \tag{88}$$

Finally, applying the Cauchy-Schwarz inequality to the first and second terms of (88), and then the norm definitions (49), (50), and (47), we arrive at the desired bound:

$$\begin{aligned}
\vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert^2 &\leq \vert\vert\vert \mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q} \vert\vert\vert_* \vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert + \left\Vert \mu^{-1}(\mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v}) \right\Vert_{\Omega} \Vert \pi_W(\mathfrak{p} - \mathfrak{q}) \Vert_{W,\mu} \\
&\leq \vert\vert\vert \mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q} \vert\vert\vert_* \vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert + \left\Vert \mu^{-1}(\mathfrak{f} - \mathfrak{D} \cdot \mathfrak{v}) \right\Vert_{\Omega} \vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert \leq \mathcal{M}(\mathfrak{q}, \mathfrak{v}, \mathfrak{f}, \mu) \vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert.
\end{aligned} \tag{89}$$

(2) The proof for the bounds for the dual variable is given next. We remark that an alternative proof based on a generalized abstract estimate (see [26], Theorem 6.1) can be used to obtain equivalent upper bounds after its generalization to the mD setting.

We start by adding the square of the primal and dual error to obtain:

$$\begin{aligned}
&\vert\vert\vert \mathfrak{p} - \mathfrak{q} \vert\vert\vert^2 + \vert\vert\vert \mathfrak{u} - \mathfrak{v} \vert\vert\vert_*^2 = \langle \mathfrak{K}\,\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}), \mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} + \left\langle \mathfrak{K}^{-1}(\mathfrak{u} - \mathfrak{v}), \mathfrak{u} - \mathfrak{v} \right\rangle_{\Omega,\Gamma} \\
&= \left\langle \mathfrak{u} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}, \mathfrak{K}^{-1}\mathfrak{u} + \mathbb{D}\,\mathfrak{q} \right\rangle_{\Omega,\Gamma} + \left\langle \mathfrak{K}^{-1}(\mathfrak{u} - \mathfrak{v}), \mathfrak{u} - \mathfrak{v} \right\rangle_{\Omega,\Gamma} \\
&= \left\langle \mathfrak{u} - \mathfrak{v} + \mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}, \mathfrak{K}^{-1}\mathfrak{u} - \mathfrak{K}^{-1}\mathfrak{v} + \mathbb{D}\,\mathfrak{q} + \mathfrak{K}^{-1}\mathfrak{v} \right\rangle_{\Omega,\Gamma} + \left\langle \mathfrak{K}^{-1}(\mathfrak{u} - \mathfrak{v}), \mathfrak{u} - \mathfrak{v} \right\rangle_{\Omega,\Gamma} \\
&= \left\langle \mathfrak{v} + \mathfrak{K}\,\mathbb{D}\,\mathfrak{q}, \mathfrak{K}^{-1}\mathfrak{v} + \mathbb{D}\,\mathfrak{q} \right\rangle_{\Omega,\Gamma} + 2\langle \mathfrak{u} - \mathfrak{v}, -\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma} \\
&= \left\langle \mathfrak{K}^{-\frac{1}{2}}\mathfrak{v} + \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,\mathfrak{q}, \mathfrak{K}^{-\frac{1}{2}}\mathfrak{v} + \mathfrak{K}^{\frac{1}{2}}\mathbb{D}\,\mathfrak{q} \right\rangle_{\Omega,\Gamma} + 2\langle \mathfrak{u} - \mathfrak{v}, -\mathbb{D}\,(\mathfrak{p} - \mathfrak{q}) \rangle_{\Omega,\Gamma}.
\end{aligned} \tag{90}$$

Here, we used the norm definitions (49) and (50) together with the mD constitutive relationship (29a).

Using partial integration, mass conservation (29b), and the orthogonality property (46), the second term of (90) can be equivalently written as

$$\begin{aligned}&\langle \mathfrak{u}-\mathfrak{v}, -\mathbb{D}\,(\mathfrak{p}-\mathfrak{q})\rangle_{\Omega,\Gamma} = \langle \mathfrak{D}\cdot(\mathfrak{u}-\mathfrak{v}), -(\mathfrak{p}-\mathfrak{q})\rangle_{\Omega} = \langle \mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v}, -(\mathfrak{p}-\mathfrak{q})\rangle_{\Omega}\\ &\quad = \langle \mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v}, -\pi_W(\mathfrak{p}-\mathfrak{q})\rangle_{\Omega} = \left\langle \mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v}), -\mu\pi_W(\mathfrak{p}-\mathfrak{q})\right\rangle_{\Omega}. \end{aligned} \tag{91}$$

Using the Cauchy-Schwarz inequality twice and the definition of the weighted norms (47), (91) can be estimated as

$$\begin{aligned}&\left|\langle \mathfrak{u}-\mathfrak{v}, -\mathbb{D}\,(\mathfrak{p}-\mathfrak{q})\rangle_{\Omega,\Gamma}\right| \leq \left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega}\|\pi_W(\mathfrak{p}-\mathfrak{q})\|_{W,\mu}\\ &\quad = \left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega}|\!|\!|\mathfrak{p}-\mathfrak{q}|\!|\!| \leq \frac{1}{2}\left(\left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega}^{2} + |\!|\!|\mathfrak{p}-\mathfrak{q}|\!|\!|^{2}\right). \end{aligned} \tag{92}$$

Substituting (92) into (90) and applying the Cauchy-Schwarz inequality to the first term, we arrive at,

$$|\!|\!|\mathfrak{u}-\mathfrak{v}|\!|\!|_{*}^{2} \leq |\!|\!|\mathfrak{v}+\mathfrak{K}\mathbb{D}\,\mathfrak{q}|\!|\!|_{*}^{2} + \left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega}^{2},$$

from which we conclude that (56) indeed holds.

(3) To prove the upper bound for the primal-dual pair, we choose an arbitrary pair $(\mathfrak{q},\mathfrak{v}) \in (H_0^1(\Omega)+\mathfrak{g}) \times H(\mathrm{div};\Omega,\Gamma;U)$, and measure its difference with the exact solution $(\mathfrak{p},\mathfrak{u}) \in (H_0^1(\Omega)+\mathfrak{g}) \times H(\mathrm{div};\Omega,\Gamma)$ in the norm (52), to get

$$\begin{aligned}|\!|\!|[(\mathfrak{p}-\mathfrak{q},\mathfrak{u}-\mathfrak{v})]|\!|\!| &= |\!|\!|\mathfrak{p}-\mathfrak{q}|\!|\!| + |\!|\!|\mathfrak{u}-\mathfrak{v}|\!|\!|_{*} + \left\|\mu^{-1}\mathfrak{D}\cdot(\mathfrak{u}-\mathfrak{v})\right\|_{\Omega}\\ &\leq 2\mathcal{M} + \left\|\mu^{-1}\mathfrak{D}\cdot(\mathfrak{u}-\mathfrak{v})\right\|_{\Omega},\end{aligned}$$

where we use the bounds (55) and (56).

For the proof of the lower bound, we start from the definition of the majorant, to get

$$\begin{aligned}\mathcal{M} &= |\!|\!|\mathfrak{v}+\mathfrak{K}\mathbb{D}\,\mathfrak{q}|\!|\!|_{*} + \left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega}\\ &\leq |\!|\!|\mathfrak{u}-\mathfrak{v}|\!|\!|_{*} + |\!|\!|\mathfrak{K}\mathbb{D}\,(\mathfrak{p}-\mathfrak{q})|\!|\!|_{*} + \left\|\mu^{-1}(\mathfrak{f}-\mathfrak{D}\cdot\mathfrak{v})\right\|_{\Omega} = |\!|\!|[(\mathfrak{p}-\mathfrak{q},\mathfrak{u}-\mathfrak{v})]|\!|\!|.\end{aligned}$$

This completes the proof for the two-sided bounds and the abstract theorem. □

D Exact solutions to numerical validations

Herein, we provide the exact expressions for the pressure, velocities, mortar fluxes, and source terms for the numerical validations presented in Section 7.

We will conveniently define the following quantities for notational compactness:

$$\begin{aligned}\alpha(x) &= x_1 - 0.50, & &\\ \beta_1(x) &= x_2 - 0.25, & \beta_2(x) &= x_2 - 0.75,\\ \gamma_1(x) &= x_3 - 0.25, & \gamma_2(x) &= x_3 - 0.75,\end{aligned}$$

where $x = [x_1, x_2, x_3]$.

D.1 Exact solutions for the 1d/2d validation

The matrix subdomain Ω_2 is decomposed into three regions, i.e. $\Omega_2 = \cup_{k=1}^{3} \Omega_2^k$, given by:

$$\begin{aligned}
\Omega_2^1 &= \{x \in \Omega_2 : 0.00 < x_2 < 0.25\} \,, \\
\Omega_2^2 &= \{x \in \Omega_2 : 0.25 \leq x_2 < 0.75\} \,, \\
\Omega_2^3 &= \{x \in \Omega_2 : 0.75 \leq x_2 < 1.00\} \,.
\end{aligned}$$

Let us now define the distance function $d(x)$ from Ω_2 to Ω_1. That is,

$$d(x) = \begin{cases} \left(\alpha(x)^2 + \beta_1(x)^2\right)^{0.5}, & x \in \Omega_2^1 \\ \left(\alpha(x)^2\right)^{0.5}, & x \in \Omega_2^2 \\ \left(\alpha(x)^2 + \beta_2(x)^2\right)^{0.5}, & x \in \Omega_2^3 \end{cases} , \tag{93}$$

and the bubble function $\omega(x)$:

$$\omega(x) = \begin{cases} \beta_1(x)^2 \beta_2(x)^2, & x \in \Omega_2^2 \\ 0, & \text{otherwise} \end{cases} . \tag{94}$$

In Table 7, we include the exact solutions for all the variables of interest. Note that the parameter n controls the regularity of the solution. For this particular validation, a value of $n = 1.5$ was adopted.

D.2 Exact solutions for the 2d/3d validation

Analogously to the previous case, we decompose the three-dimensional matrix Ω_2 into nine subdomains, i.e. $\Omega_2 = \cup_{k=1}^{9} \Omega_2^k$, given by

$$\begin{aligned}
\Omega_2^1 &= \{x \in \Omega_2 : 0.00 < x_2 < 0.25,\ 0.00 < x_3 < 0.25\} \,, \\
\Omega_2^2 &= \{x \in \Omega_2 : 0.00 < x_2 < 0.25,\ 0.25 \leq x_3 < 0.75\} \,, \\
\Omega_2^3 &= \{x \in \Omega_2 : 0.00 < x_2 < 0.25,\ 0.75 \leq x_3 < 1.00\} \,, \\
\Omega_2^4 &= \{x \in \Omega_2 : 0.25 \leq x_2 < 0.75,\ 0.00 < x_3 < 0.25\} \,, \\
\Omega_2^5 &= \{x \in \Omega_2 : 0.25 \leq x_2 < 0.75,\ 0.25 \leq x_3 < 0.75\} \,, \\
\Omega_2^6 &= \{x \in \Omega_2 : 0.25 \leq x_2 < 0.75,\ 0.75 \leq x_3 < 1.00\} \,, \\
\Omega_2^7 &= \{x \in \Omega_2 : 0.75 \leq x_2 < 1.00,\ 0.00 < x_3 < 0.25\} \,, \\
\Omega_2^8 &= \{x \in \Omega_2 : 0.75 \leq x_2 < 1.00,\ 0.25 \leq x_3 < 0.75\} \,, \\
\Omega_2^9 &= \{x \in \Omega_2 : 0.75 \leq x_2 < 1.00,\ 0.75 \leq x_3 < 1.00\} \,.
\end{aligned}$$

Table 7: Exact solutions for the 1d/2d validation.

$p_2 =$	$d^{n+1} + \omega d$	Ω_2^2
	d_2^{n+1}	$\Omega_2 \setminus \Omega_2^2$
$u_2 =$	$-d^{n+1}(n+1)\begin{bmatrix}\alpha & \beta_1\end{bmatrix}$	Ω_2^1
	$-d\begin{bmatrix}\alpha^{-1}\left(\omega + d^n(n+1)\right) & 2\beta_1^2\beta_2 + 2\beta_1\beta_2^2\end{bmatrix}$	Ω_2^2
	$-d^{n+1}(n+1)\begin{bmatrix}\alpha & \beta_2\end{bmatrix}$	Ω_2^3
$f_2 =$	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_1^2 d^{n-1}(n-1)\right)$	Ω_2^1
	$-2d\left(\beta_1(\beta_1 + 2\beta_2) + \beta_2(2\beta_1 + \beta_2)\right) - d^{n-1}n(n+1)$	Ω_2^2
	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_2^2 d^{n-1}(n-1)\right)$	Ω_2^3
$\lambda_1 =$	ω	Γ_1
$\lambda_2 =$	ω	Γ_2
$p_2 =$	0	$\partial_1\Omega_2$
$p_2 =$	0	$\partial_2\Omega_2$
$p_1 =$	$-\omega$	Ω_1
$u_1 =$	$\begin{bmatrix}0 & 2\beta_1^2\beta_2 + 2\beta_1\beta_2^2\end{bmatrix}$	Ω_1
$\sum_{j\in\hat{S}_1}\lambda_j =$	2ω	Ω_1
$f_1 =$	$8\beta_1\beta_2 + 2(\beta_1^2 + \beta_2^2) - 2\omega$	Ω_1

The distance function $d_2(x)$ from Ω_2 to Ω_1 is now given by

$$d_2(x) = \begin{cases} \left(\alpha(x)^2 + \beta_1(x)^2 + \gamma_1(x)^2\right)^{0.5}, & x \in \Omega_2^1, \\ \left(\alpha(x)^2 + \beta_1(x)^2\right)^{0.5}, & x \in \Omega_2^2, \\ \left(\alpha(x)^2 + \beta_1(x)^2 + \gamma_2(x)^2\right)^{0.5}, & x \in \Omega_2^3, \\ \left(\alpha(x)^2 + \gamma_1(x)^2\right)^{0.5}, & x \in \Omega_2^4, \\ \left(\alpha(x)^2\right)^{0.5}, & x \in \Omega_2^5, \\ \left(\alpha(x)^2 + \gamma_2(x)^2\right)^{0.5}, & x \in \Omega_2^6, \\ \left(\alpha(x)^2 + \beta_2(x)^2 + \gamma_1(x)^2\right)^{0.5}, & x \in \Omega_2^7, \\ \left(\alpha(x)^2 + \beta_2(x)^2\right)^{0.5}, & x \in \Omega_2^8, \\ \left(\alpha(x)^2 + \beta_2(x)^2 + \gamma_2(x)^2\right)^{0.5}, & x \in \Omega_2^9, \end{cases} \tag{95}$$

and the bubble function $\omega(x)$:

$$\omega(x) = \begin{cases} \beta_1(x)^2\beta_2(x)^2\gamma_1(x)^2\gamma_2(x)^2, & x \in \Omega_2^5 \\ 0, & \text{otherwise} \end{cases}. \tag{96}$$

In Table 8, we show the exact solutions for all the variables of interest. Once again, a value of $n = 1.5$ is adopted for this validation.

Table 8: Exact solutions for the $2d/3d$ validation.

$p_2 =$	$d^{n+1} + \omega d$	Ω_2^2
	d_2^{n+1}	$\Omega_2 \setminus \Omega_2^2$
$u_2 =$	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_1 & \gamma_1\end{bmatrix}$	Ω_2^1
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_1 & 0\end{bmatrix}$	Ω_2^2
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_1 & \gamma_2\end{bmatrix}$	Ω_2^3
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & 0 & \gamma_1\end{bmatrix}$	Ω_2^4
	$-d\begin{bmatrix}\alpha^{-1}(\omega + d^n(n+1)) & 2\beta_1^2\beta_2\gamma_1^2\gamma_2^2 + 2\beta_1\beta_2^2\gamma_1^2\gamma_2^2 & 2\beta_1^2\beta_2^2\gamma_1^2\gamma_2 + 2\beta_1^2\beta_2^2\gamma_1\gamma_2^2\end{bmatrix}$	Ω_2^5
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & 0 & \gamma_2\end{bmatrix}$	Ω_2^6
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_2 & \gamma_1\end{bmatrix}$	Ω_2^7
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_2 & 0\end{bmatrix}$	Ω_2^8
	$-d^{n-1}(n+1)\begin{bmatrix}\alpha & \beta_2 & \gamma_2\end{bmatrix}$	Ω_2^9
$f_2 =$	$-d^{-2}(n+1)\left(3d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_1^2 d^{n-1}(n-1) + \gamma_1^2 d^{n-1}(n-1)\right)$	Ω_2^1
	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_1^2 d^{n-1}(n-1)\right)$	Ω_2^2
	$-d^{-2}(n+1)\left(3d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_1^2 d^{n-1}(n-1) + \gamma_2^2 d^{n-1}(n-1)\right)$	Ω_2^3
	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \gamma_1^2 d^{n-1}(n-1)\right)$	Ω_2^4
	$-2d\left(\beta_1^2\beta_2^2\left(\gamma_1\left(\gamma_1 + 2\gamma_2\right) + \gamma_2\left(2\gamma_1 + \gamma_2\right)\right) + \gamma_1^2\gamma_2^2\left(\beta_1\left(\beta_1 + 2\beta_2\right) + \beta_2\left(2\beta_1 + \beta_2\right)\right)\right)$ $-\alpha^{-2}\omega d^{n+1}(n+1)^2 - \alpha^{-2}\omega d^{n+1}(n+1)$	Ω_2^5
	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \gamma_2^2 d^{n-1}(n-1)\right)$	Ω_2^6
	$-d^{-2}(n+1)\left(3d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_2^2 d^{n-1}(n-1) + \gamma_1^2 d^{n-1}(n-1)\right)$	Ω_2^7
	$-d^{-2}(n+1)\left(2d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_2^2 d^{n-1}(n-1)\right)$	Ω_2^8
	$-d^{-2}(n+1)\left(3d^{n+1} + \alpha^2 d^{n-1}(n-1) + \beta_2^2 d^{n-1}(n-1) + \gamma_2^2 d^{n-1}(n-1)\right)$	Ω_2^9
$\lambda_1 =$	ω	Γ_1
$\lambda_2 =$	ω	Γ_2
$p_2 =$	0	$\partial_1\Omega_2$
$p_2 =$	0	$\partial_2\Omega_2$
$p_1 =$	$-\omega$	Ω_1
$u_1 =$	$\begin{bmatrix}0 & 2\gamma_1^2\gamma_2^2(\beta_1\beta_2^2 + \beta_1^2\beta_2) & 2\beta_1^2\beta_2^2(\gamma_1\gamma_2^2 + \gamma_1^2\gamma_2)\end{bmatrix}$	Ω_1
$\sum_{j\in\hat{S}_1}\lambda_j =$	2ω	Ω_1
$f_1 =$	$\beta_1^2\gamma_2^2 + 4\beta_1\beta_2\gamma_2^2 + \beta_2^2\gamma_1^2 + 4\beta_2^2\gamma_1\gamma_2 + 2\beta_2^2\gamma_2^2 - 2\omega$	Ω_1

NORDIC ECOLABEL 241 PRINTED MATTER 731

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

UNIVERSITAS BERGENSIS

uib.no

ISBN: 9788230850886 (print)
9788230853320 (PDF)