Preconditioning for Flow in Fractured Porous Media

Ana Budiša

Thesis for the degree of Philosophiae Doctor (PhD)
University of Bergen, Norway
2020

UNIVERSITY OF BERGEN

Preconditioning for Flow in Fractured Porous Media

Ana Budiša

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

Date of defense: 24.01.2020

Year: 2020

Title: Preconditioning for Flow in Fractured Porous Media

Name: Ana Budiša

Print: Skipnes Kommunikasjon / University of Bergen

Preface

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

The PhD project has been financially supported by Norwegian Research Council grant 250223.

Acknowledgments

Gratitude. Putting that word on paper does not come near to how grateful I am to all the people that have supported me throughout my PhD journey. Indeed, it has been a journey, at times quite rocky, windy, wavy and definitely rainy, but the only way out was through. Now, at the end of it, I am glad that the first thing I can share is gratitude. (For knowledge see Chapter 1, 2,...)

I am immensely grateful to my PhD advisors. Eirik, thank you for providing me with knowledge and opportunities to develop my research and myself. Thank you for your patience, support, encouragement and always having plan B when the journey got difficult. Also, thank you Jan and Florin for the inspiring thoughts and discussions that have influenced my research.

Dear Porous Media Group, you are awesome. It has been a great honour to be a part of it, these three years would not have been the same without any of you. A special thanks goes to Elyes and Alessio for our research collaboration, I have learned so much from you two. Wonderful Wietse, thank you for believing in me when I could not, it meant a great deal. Max, thank you for all those one-last-beers, I hope we will have many more. I have to also extend my thanks to Michael, Ivar, Ingeborg, Runar, Jakub, Manuel and David for all the good days and great nights out together.

I am especially grateful to Xiaozhe Hu who has hosted me at Tufts University for three months. Your intuition and knowledge has greatly shaped the research in my thesis. It has been a pleasure collaborating with you and I hope we can continue on this path. Thank you also for hosting me at your home and for all the amazing food you and James introduced me to. I'm still searching for that no. 1 burger though!

Neizmjerno fala mojoj familiji and prijateljima u Hrvatskoj. Fala mojim roditeljima Dariji i Zdravku i mojoj sestri Mariji (i ludoj Luni!) šta me uvik spremno i sretno dočekaju nazad di god da išla. Fala mojim prijateljima s faksa i iz djetinjstva šta su još uvik tu uz mene iako kilometrima daleko.

Thank you!

Abstract

The key to reliable simulations of flow in fractured porous media is the proper design, analysis and implementation of numerical methods. These methods should take into account the specific properties of the underlying model, while at the same time be flexible enough to handle variability of the model's components. The particular features of fractured porous media we concern ourselves with are the complex geometry of the fracture network and the disparity in scales in the model parameters. The model we study is based on interpretation of fractures and the porous rock as a mixed-dimensional geometry, and the resulting system of partial differential equations is highly coupled and parameter-dependent. In this thesis, we build upon the common approaches to discretization of the flow problem and deliver a numerical solution by constructing efficient numerical solvers and preconditioners. The two main topics of our research are the design of preconditioners to finite element discretization of the linear flow model and the development of linearization methods to the non-linear model.

In the first part, we consider the fact that our flow problem reveals the saddle-point structure. This motivates to see how some established approaches to preconditioning saddle-point problems work under the mixed-dimensional complexity. We construct the preconditioners to the classical solving approaches, such as Krylov subspace methods, based on the well-posedness of our saddle-point system. As the goal of any preconditioner is to approximate the inverse of the coefficient operator of the system, the principal idea of our approach is to find that inverse mapping that is equivalent in terms of norms on the given function spaces. In this way, we can ensure that the preconditioned numerical solvers will converge more rapidly, independently of values of the discretization and physical parameters. In our case, we are able to derive two such preconditioners by identifying two different topologies on the given discrete finite-element spaces. In fact, one of the approaches leads to a general framework to preconditioning mixed-dimensional elliptic problems that can be applied to other problems with a similar hierarchical structure as the model of flow in fractured porous media.

Finally, we study a choice of non-linear and time-dependent flow models that appear in cases of enhanced conductivity of the fractures and compressibility of the fluid. The development of the iterative solution methods for our problem considers the natural domain decomposition setting imposed by the fracture network and the standard linearization methods are adapted to the mixed-dimensional setting. By using non-matching grids, we can employ a multiscale method to the interface problem to handle the dominating computational cost

in each iteration of the non-linear solver, namely the repeated solving process on the rock matrix subdomains. The flexibility of the method is showcased by successfully applying it to several non-linear flow models.

Outline

This dissertation consists of two parts. Part I introduces the background theory covered in the scientific results in Part II.

Part I consists of four chapters. Chapter 1 introduces the subject of flow in a porous medium with fractures and how it is encompassed in an broader application of subsurface energy storage. In Chapter 2, we present the governing laws in a mathematical setting, from a continuous model to a discrete system of equations. Then, we provide in Chapter 3 a set of computational methods and preconditioning techniques that serve as linear and non-linear solvers for the systems of equations arising in Chapter 2. Finally, Chapter 4 contains short summaries of scientific papers included in Part II, as well as an outlook on future research.

Part II presents the main results of the dissertation collected in five scientific articles:

Paper A A. Budiša, X. Hu, *Block Preconditioners for Mixed-Dimensional Discretization of Flow in Fractured Porous Media*, Computational Geosciences, in review. arXiv:1910.04704 [math.NA].

Paper B A. Budiša, W. M. Boon, X. Hu, *Mixed-Dimensional Auxiliary Space Preconditioners*, SIAM Journal on Scientific Computing, in review. arXiv:1905.13513 [math.NA].

Paper C E. Ahmed, A. Fumagalli, A. Budiša (2019), *A Multiscale Flux Basis for Mortar Mixed Discretizations of Reduced Darcy-Forchheimer Fracture Models*, Computer Methods in Applied Mechanics and Engineering 354, 16–36. doi: 10.1016/j.cma.2019.05.034.

Paper D E. Ahmed, A. Fumagalli, A. Budiša, E. Keilegavlen, J. M. Nordbotten, A. F. Radu, *Robust Linear Domain Decomposition Schemes for Reduced Nonlinear Fracture Flow Models*, SIAM Journal on Numerical Analysis, in review. arXiv:1906.05831 [math.NA].

Contents

Preface iii

Acknowledgments v

Abstract vii

Outline ix

I Scientific Background 1

1 Introduction 3
1.1 Motivation . . . 4
1.1.1 Subsurface Energy Storage . . . 4
1.1.2 Flow in Fractured Porous Media . . . 5

2 Mathematical Model of Flow in Fractured Porous Media 7
2.1 Mixed-Dimensional Geometry . . . 7
2.2 Linear Flow Model . . . 9
2.2.1 Variational Formulation . . . 10
2.2.2 Conforming Discretization . . . 12
2.2.3 Non-matching Grids . . . 13
2.3 Non-linear and Time-Dependent Flow Models . . . 14
2.3.1 Non-linear Fracture Flow Models . . . 14
2.3.2 Compressible Flow Model . . . 15

3 Solvers and Preconditioners 17
3.1 Direct and Iterative Methods . . . 18
3.1.1 Sparse Direct Solvers . . . 18
3.1.2 Iterative Methods . . . 19
3.2 Block Preconditioners . . . 22
3.2.1 The General Theory . . . 22
3.2.2 Application to the Mixed-Dimensional Problem . . . 24
3.2.3 Preconditioners Based on an Alternative Formulation . . . 26

3.2.4 Auxiliary Space Preconditioners 27
3.3 Domain Decomposition Method 31
3.3.1 General Approach 31
3.3.2 Application to the Fracture Flow Model 33
3.3.3 Multiscale Flux Basis 35
3.4 Linearization Methods 36
3.4.1 Fixed-point method 37
3.4.2 Newton's method 38
3.4.3 L-scheme 39

4 Summary and Outlook **41**
4.1 Summary of the Papers 41
4.2 Outlook 45

Bibliography **47**

II Scientific Results **53**

A Block Preconditioners for Mixed-dimensional Discretization of Flow in Fractured Porous Media

B Mixed-Dimensional Auxiliary Space Preconditioners

C A Multiscale Flux Basis for Mortar Mixed Discretizations of Reduced Darcy-Forchheimer Fracture Models

D Robust Linear Domain Decomposition Schemes for Reduced Non-linear Fracture Flow Models

Part I

Scientific Background

Chapter 1

Introduction

In industry and research, simulations of physical phenomena play an important role in observing, understanding and utilizing the world we live in. It can lead us to answers to fundamental questions, design and implementation of scientific experiments, but also decision-making and realization of projects in practice. Therefore, the simulations need to be reliable and computationally feasible. In order to provide that, we turn to developing computational techniques using a range of mathematical tools to accurately describe physical processes of our interest.

These processes are commonly represented as systems of partial differential equations. As any equations, we aim to determine that the (unique) solution exists, but actually finding the exact solution is rarely possible. Discrete approximations, however, turn the continuous problem into a system of algebraic equations that can be solved by a variety of numerical methods. Regardless of choice of the discretization, these algebraic systems inevitably become large and demand generous computational power and memory. In particular, the higher accuracy of the approximative solution is required, the larger the resulting number of unknowns in the system. Thus, the numerical methods we use have to be scalable while preserving the fidelity of the numerical solution.

The typical approach to finding the numerical solution is to construct new or modify the existing iterative methods. In many situations they are often a flexible and computationally cheap choice, but still the major cost of these methods is the number of iterations needed to converge. The iteration count is closely related to the number of unknowns and disparity in magnitudes between possibly many parameters in the system. A remedy to scalability issues of the iterative methods is *preconditioning*. The idea of imposing preconditioners to iteratively solving large systems of equations is to achieve convergence with hopefully a handful of iterations needed, independently of the size of the system and given parameters. Hence, the preconditioners are usually dependent on the problem in hand, although it is desirable to provide general design techniques. In fact, they can directly follow from the properties of system of equations we seek to solve.

1.1 Motivation

In this thesis, we argue that we can derive iterative solution methods and preconditioners with mentioned features in applications to fluid flow in porous media with fractures or, more generally, thin inclusions. We describe the main characteristics of this physical process, its relevance in most common real-life applications, and the main developments of our work related to its simulations.

1.1.1 Subsurface Energy Storage

The simulations of fluid flow processes in fractured porous media are without doubt a crucial part of various applications in biomedicine, geology and engineering. Many of them related to subsurface fractured reservoirs, such as CO_2 and energy storage, nuclear waste disposal and geothermal energy production, depend on accurate predictions of flow patterns within to enhance the storage capacity or to avoid potential leakage hazard. The application that served as the main motivation to work in this thesis is the thermo-mechanical subsurface energy storage (TheMSES).

TheMSES concerns storage and production of thermal and mechanical energy by injecting and extracting fluids under high temperature and pressure into the subsurface reservoirs. This is especially attractive in storing the excess energy from sources that are subject to great production variability in time. For example, the renewable energy sources, such as wind and solar power, depend on favorable weather conditions for energy production and that sometimes does not match the energy demand in the society. For that, the natural subsurface permeable layers can provide a large storage capacity to balance between different time scales of energy consumption and supply. In addition, by increasing the use of energy from sustainable resources, it decreases the need for the use of fossil fuels. This can in turn lower our carbon footprint and contribute to mitigating climate change effects. The suitable storage sites can be found globally in different geological formations in shallow and deeper subsurface, for example depleted oil and gas reservoirs and salt caverns. The implementation of the TheMSES requires structured planning, economical assessment and above all scientific analysis of energy storage options. Although similar technologies has been already been studied to a certain degree [10, 63], there are still many open challenges in research and engineering regarding the applications of TheMSES.

There are many features of TheMSES which must be studied in developing accurate scientific models, including hydraulic, mechanical, thermal, chemical and microbial effects. To start building up such complex models, we need to first consider the dominating process in our system – fluid flow. The geological formations that we are interested in are constituted of porous materials that transmit the fluid through the subsurface reservoir. Often these porous materials also contain large connected pathways, called fractures, that are distinguishable by their dominating influence on the flow behavior. Therefore, this thesis focuses on the aspects of flow within the fractured rock, more specifically the computational side of the fracture fluid flow modeling.

1.1.2 Flow in Fractured Porous Media

In the context of porous media, the particular characteristic of fractures is their large-aspect-ratio in relation to the surrounding material due to their long but very thin appearance. They are often intertwined forming complex networks and may be empty or filled with material, meaning can be considered as a porous medium in itself. Moreover, they are typically more conductive than of the surrounding rock so we expect that the overall fluid flow behavior will be mostly stimulated by the fracture flow. On the other hand, they can act as natural barriers and drastically divert or stop the flow pathways. As such, their impact depends on their size and conductivity properties which can scale individually over several orders of magnitude. In order to perform simulations of flow in fractured porous media, we rely on mathematical models and numerical representations that exploit all the aspects of this physical phenomenon, ranging from geometry, material properties to governing flow processes. An overview of common modeling approaches can be found in [12].

One of the main components of any model of flow in fractured porous media is the distinctive geometry of the problem. Since fractures have a dominating effect on the flow patterns in the system, it motivates to consider them as an explicit part of the model. Those models, often referred to as Discrete Fracture Model (DFM), represent the fractures separately from the rock domain. However, it is not always feasible in practice to incorporate all fractures, especially very small ones. The common approach is to include a small number of them, based on some selection criteria often related to length or connectivity properties, and upscale the rest as a part of the rock domain. In this way, the conceptual model of fractured porous media can be applied to cases when the rock is considered practically impermeable.

Furthermore, fractures act as thin inclusions within the medium, with their width size, called aperture, significantly smaller than the other lateral extensions. Therefore, it is only natural to represent them as separate lower-dimensional geometrical entities. This kind of a fracture model have been introduced in [3, 4] and have gained a lot of interest in research recently. The benefits of this approach are numerous, especially regarding separate meshing and flexibility of incorporating different material properties and governing laws on each fracture. The significant research efforts have gone into developing adapting flow models, both single-phase and multiphase, and developing new discretization methods to the reduced geometry. Some contributions can be found in [16, 20, 27, 31, 56] and in the benchmark studies [11, 25].

Due to the vast amount of literature on the topic, it is undeniable that much has been invested into proper numerical methods for flow problems in fractured porous media. Still, the common challenge faced in the majority of them is the computational efficiency. Our goal is to apply those numerical methods in large-scale simulations of subsurface processes, where even today the limitations of technology are present in terms of memory and CPU power consumption. As mentioned, the algebraic systems of equations resulting from the discretization approaches to DFM often contain large number of unknowns to solve for and proper computational methods need to be developed. This refers to both iterative methods and preconditioning techniques. The most common approach is still to directly use the standard solving methods, such as domain decomposition and multigrid methods, but they are

not adapted to all the specific features of the models of fractured porous media that in many cases greatly diminish their efficiency. For example, the heterogeneity in the physical parameters often influences the stability of the methods, meaning that they may have difficulty to deliver an accurate enough solution. The fractured porous media models suffer from this problem so we need to adapt the methods to our problem or develop new ones. Although the importance of it has lately been recognized [5, 8], little has been done so far in that direction. Therefore, the work in this thesis aim to bridge that gap by providing the analytical and computational tools for efficient simulations of flow processes in a porous medium with fractures.

The main contributions of this dissertation are the following:

1. **Constructing parameter-independent preconditioners for mixed-dimensional linear flow models.** We propose a set of block preconditioners for Krylov subspace methods for solving the linear system of equations arising from the mixed finite element discretization of our linear flow model. They are based on the chosen weighted norms in which our discrete system is well-posed, independently of discretization and physical parameters. Beside being theoretically robust, the preconditioners can also be implemented straightforwardly and effectively by taking advantage of the block structure of the problem.

2. **Developing preconditioners for finite-element discretizations of general mixed-dimensional elliptic problems.** Using the established auxiliary space theory and the mixed-dimensional finite element exterior calculus, we provide a general approach to preconditioning mixed-dimensional elliptic partial differential equations. For the purpose, we extend the stable regular decomposition to the mixed-dimensional geometries in continuous and discrete sense. With that, we are able to construct parameter-independent preconditioners that can be used as a component to block preconditioners derived for mixed-dimensional flow problems. Although only implemented on the fractured porous media model, the preconditioners can directly be applied to more general elliptic problems.

3. **An analysis and comparison of numerical schemes for non-linear flow models.** We provide several approaches to linearization of a selection of non-linear fracture flow models. Based on the non-overlapping domain decomposition framework, we analyze and implement linearization schemes on the non-linear fracture interface problem that at the same time handle the inter-dimensional coupling between fractures and the rock matrix. We are also able to extend the schemes to the model that incorporates fluid compressibility with a discretization scheme in time.

Chapter 2

Mathematical Model of Flow in Fractured Porous Media

This chapter starts with a brief description of mathematical aspects of the governing laws of flow in fractured porous media. The systems of equations arising from the mathematical model are the main study points of this thesis and their properties are what leads us to designing fast and robust solvers for simulating the underlying physical processes.

We first present how complex fractured domains can be modeled as composite mixed-dimensional geometrical structures. After that has been set, we continue with introducing the linear single-phase flow model, both in strong and weak formulation, and complete it with choices of discretization and the resulting algebraic block form. Then, the linear model is extended to non-linear and time-dependent cases with their respective discrete formulations.

2.1 Mixed-Dimensional Geometry

The geometrical interpretation of the fractured rock forms the basis of our model. Unlike methods that deal with fractures at discretization step [20, 26, 57], we incorporate the fracture network into a geometrical feature that handles in whole the complex structure of the rock, fractures and their intersections before introducing the flow model. In this way we gain on the flexibility when meshing, but also it allows us to impose different governing laws and material properties on each part of the geometry.

The fractures can be interpreted as thin inclusions in a porous medium due to their large aspect ratios. A common way to accurately represent fractures is the so-called reduced model approach as developed in [3, 4, 27] and expanded more recently in [16, 47]. Therefore, we model them as lower-dimensional manifolds embedded in a porous medium domain. For example, within a three-dimensional permeable rock, the fractures are seen as two-dimensional features that can decompose the rock domain. This structure is inherited when fractures intersect, making it a one-dimensional manifold, and repeated until we reach a point manifold of dimension zero.

Formally, consider $\Omega \subset \mathbb{R}^n$ to be a domain of the fractured rock of dimension $n = 2$

or $n = 3$. For $d \leq n$ and i in an index set I^d, denote Ω_i^d as a d-dimensional manifold or a subdomain contained in Y. Intersections of several d-dimensional manifold create separate $(d-1)$-dimensional manifolds, which in turn makes all Ω_i^d disjoint. Union over the subscript set I^d represents all d-dimensional subdomains $\Omega^d = \bigcup_{i \in I^d} \Omega_i^d$. To account for the coupling between codimension one subdomains, we introduce the interfaces Γ_{ij}^d between Ω_i^d and adjacent Ω^{d+1}, for j in index set J_i^d of all neighboring higher-dimensional subdomains. The interfaces Γ_{ij}^d coincide with the subdomain Ω_i^d, but they note the side on which the interface with Ω^{d+1} is taking place, see Figure 2.1. We collect all interfaces of Ω_i^d with a union set $\Gamma_i^d = \bigcup_{j \in J_i^d} \Gamma_{ij}^d$, while the union $\Gamma^d = \bigcup_{i \in I^d} \Gamma_i^d$ represents all d-dimensional interfaces. Moreover, let $\boldsymbol{\nu}$ be the outward unit normal vector to Ω^d on the boundary $\partial\Omega^d$. Specifically, it means that on Γ_i^d the normal vector points from Ω^{d+1} toward the lower-dimensional subdomain Ω_i^d. Summing up the decomposition, the fractured porous medium domain Ω with the boundary $\partial\Omega$ and the interface Γ is given as

$$\Omega = \bigcup_{d=0}^{n} \Omega^d, \qquad \partial\Omega = \bigcup_{d=1}^{n} \partial\Omega^d \backslash \Gamma^{d-1}, \qquad \Gamma = \bigcup_{d=0}^{n-1} \Gamma^d. \tag{2.1}$$

Lastly, assume that the boundary of Ω can be partitioned to $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ such that $\partial\Omega_D \cap \partial\Omega_N = \emptyset$. We adopt the notation for each subdomain, that is, $\partial\Omega_{iD}^d = \partial\Omega_i^d \cap \partial\Omega_D$ and $\partial\Omega_{iN}^d = \partial\Omega_i^d \cap \partial\Omega_N$ for $i \in I^d$, $0 < d \leq n$. This partition of the external boundary will later be important in the flow model to set pressure boundary conditions on $\partial\Omega_D$ and flux boundary conditions on $\partial\Omega_N$.

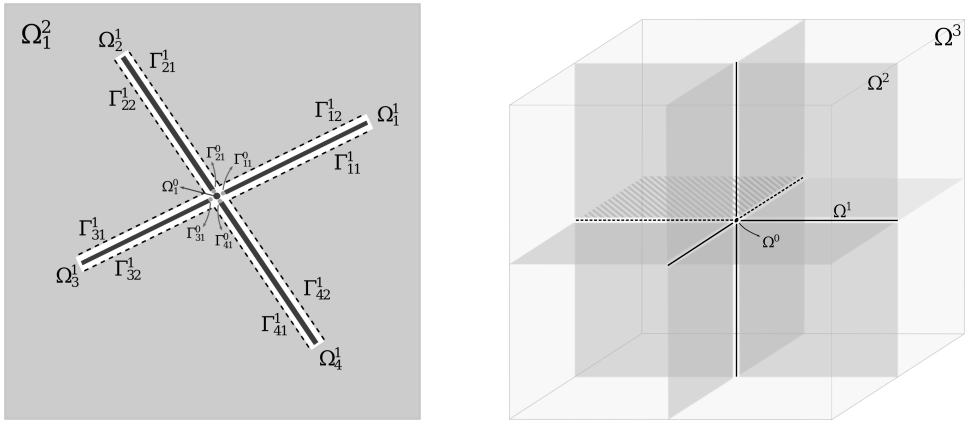


Figure 2.1: We illustrate the hierarchical structure of the mixed-dimensional decomposition of a fractured porous medium. On the left, the two-dimensional setting shows a fracture as a one-dimensional feature and interfaces that it shares with the porous medium domain. On the right, a sketch of a fracture network in three dimensions shows how the manifolds decompose the porous medium domain, as well as how they are split by intersection lines and a intersection point.

Throughout the thesis, omitting subdomain subscripts and dimension superscripts notes a parameter or a variable defined on the union of geometric features over those indices. We bring back the notation with indices in cases when clarification is necessary.

2.2 Linear Flow Model

Now that we have set up the dimensional decomposition framework, we introduce the governing laws in the subdomains and fractures. First, we start with a linear flow model.

We assume that the material permeability $\boldsymbol{K}^d$ and normal permeability $\boldsymbol{K}_{\boldsymbol{\nu}}^d$ tensors are bounded both above and below, symmetric and positive definite. Furthermore, denote with γ_{ij}^d the distance from Γ_{ij}^d to Ω_i^d, which for $d = n - 1$ represents the fracture aperture. In theory, the physical parameters $\boldsymbol{K}^d$, $\boldsymbol{K}_{\boldsymbol{\nu}}^d$ and γ_{ij}^d can vary spatially, however, we consider them to take constant values on each subdomain in Ω as it is usually the case in the papers in Part II.

In each subdomain Ω^d, we introduce the governing laws of the single-phase incompressible flow model – Darcy's law and mass conservation [49, 65]. It states to find fluid velocity or flux $\boldsymbol{u}^d$ and pressure p^d that satisfy

$$\boldsymbol{u}^d = -\boldsymbol{K}^d \nabla p^d, \qquad \text{in } \Omega^d, \quad 0 < d \leq n, \tag{2.2a}$$

$$\nabla \cdot \boldsymbol{u}^d + [\![\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu}]\!] = f^d, \qquad \text{in } \Omega^d, \quad 0 \leq d \leq n, \tag{2.2b}$$

with the jump term $[\![\cdot]\!]$ defined as

$$[\![\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu}]\!]|_{\Omega_i^d} = -\sum_{j \in J_i^d} (\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu})|_{\Gamma_{ij}^d}, \qquad i \in I^d, \quad 0 \leq d \leq n-1. \tag{2.3}$$

that represents the in-flow source from the adjacent $(d+1)$-dimensional subdomains. Bear in mind that although there is no flow in the point Ω^0, to complete the model we set $\boldsymbol{u}^0$ to zero and then (2.2b) reads $[\![\boldsymbol{u}^1 \cdot \boldsymbol{\nu}]\!] = f^0$, which is actually balancing the inflow and outflow sources in Ω^0. In the same manner, since there is no notion of interface Γ^n, the (2.2b) reduces only to $\nabla \cdot \boldsymbol{u}^n = f^n$.

Furthermore, we introduce an additional interface conditions on Γ_{ij}^d to describe the notion of normal flux due to difference in pressure from Ω_i^d to Ω^{d+1},

$$(\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu})|_{\Gamma_{ij}^d} = -\boldsymbol{K}_{\boldsymbol{\nu}}^d \frac{p_i^d - p^{d+1}|_{\Gamma_{ij}^d}}{\gamma_{ij}^d}, \quad \text{on } \Gamma_{ij}^d, \qquad j \in J_i^d,\ i \in I^d,\ 0 \leq d \leq n-1, \tag{2.4}$$

where $p^{d+1}|_{\Gamma_{ij}^d}$ is the trace of pressure p^{d+1} on the interface Γ_{ij}^d, assuming sufficient regularity to take such trace. To complete the system, an example of boundary conditions is given with

$$p^d = g^d, \qquad \text{on } \partial\Omega_D^d, \quad 0 \leq d \leq n, \tag{2.5a}$$

$$\boldsymbol{u}^d \cdot \boldsymbol{\nu} = 0, \qquad \text{on } \partial\Omega_N^d, \quad 0 < d \leq n. \tag{2.5b}$$

Here, we have used $\boldsymbol{u}^d$ as the flux integrated over the direction normal to Ω^d and p^d as averaged pressure in each Ω^d, $0 \leq d \leq n$. Therefore, the reduced equations on each Ω^d for $d \neq n$ require scaling of the permeability tensors $\boldsymbol{K}^d$ and $\boldsymbol{K}_{\boldsymbol{\nu}}^d$ with the cross-sectional area of order $\mathcal{O}(\gamma^{n-d})$. A more detailed discussion on the scaling is given in [16].

There are several ways to set weak formulations of the above flow model that are mutually equivalent, that is, have the same unique solution. Each of them motivates a different kind of discretization scheme that can be used to approximate the solution of the continuous problem, that are not necessarily equivalent. Hence, the resulting algebraic formulation may have different properties, such as definiteness and eigenvalue spectrum, that in turn require different linear and non-linear solvers. In the following, we will give a few possible choices of the weak and discrete formulations of the problem (2.2)–(2.5) that are considered in the papers in Part II.

2.2.1 Variational Formulation

We begin with defining the necessary function spaces. For any open bounded set $\omega \in \mathbb{R}^n$, let $L^2(\omega)$ be the space of square-integrable functions on ω, which is a Hilbert space with the inner product $(f, g)_\omega = \int_\omega fg\, dx$ and induced norm $\|f\|_\omega^2 = (f, f)_\omega^2$. Furthermore, we require a function space $H(\mathrm{div}, \omega)$, a subspace of $(L^2(\omega))^n$ of functions with square-integrable divergence, that is, for $\boldsymbol{f} \in (L^2(\omega))^n$ we have $\nabla \cdot \boldsymbol{f} \in L^2(\omega)$. The function space $H(\mathrm{div}, \omega)$ is also a Hilbert space with the inner product $(f, g)_{H(\mathrm{div},\omega)} = (f, g)_\omega + (\nabla \cdot f, \nabla \cdot g)_\omega$.

To find a weak and, subsequently, a discrete solution to (2.2)–(2.5), we extend this concept to the mixed-dimensional framework. On the mixed-dimensional geometry Ω with the interface Γ, the L^2 function spaces are composed as

$$L^2(\Omega) = \prod_{d=0}^{n} L^2(\Omega^d), \qquad L^2(\Gamma) = \prod_{d=0}^{n-1} L^2(\Gamma^d), \tag{2.6}$$

with the inner products

$$(\cdot, \cdot)_\Omega = \sum_{d=0}^{n} (\cdot, \cdot)_{\Omega^d}, \qquad (\cdot, \cdot)_\Gamma = \sum_{d=0}^{n} (\cdot, \cdot)_{\Gamma^d}, \tag{2.7}$$

and the induced norms

$$\|\cdot\|_\Omega^2 = \sum_{d=0}^{n} \|\cdot\|_{\Omega^d}^2, \qquad \|\cdot\|_\Gamma^2 = \sum_{d=0}^{n} \|\cdot\|_{\Gamma^d}^2. \tag{2.8}$$

We define also the mixed-dimensional divergence

$$\mathbf{D} \cdot \boldsymbol{f} = \nabla \cdot \boldsymbol{f} + [\![\boldsymbol{f} \cdot \boldsymbol{\nu}]\!], \tag{2.9}$$

that operates as a combination of a standard divergence ∇ tangentially on each Ω^d and a jump of normal traces of functions on Ω^{d+1} across the common part of the interface Γ^d.

Therefore, the functions that we can take mixed-dimensional divergence of are contained in the function space

$$H(\mathbf{D}\cdot,\Omega)=\prod_{d=1}^{n}\{\boldsymbol{f}^d\in H(\mathrm{div},\Omega^d):(\boldsymbol{f}^d\cdot\boldsymbol{\nu})|_{\Gamma^{d-1}}\in L^2(\Gamma^{d-1})\}. \tag{2.10}$$

The relevant function spaces for the variational formulation now are

$$\begin{aligned}\boldsymbol{V}&=\{\boldsymbol{v}\in H(\mathbf{D}\cdot,\Omega):(\boldsymbol{v}\cdot\boldsymbol{\nu})|_{\partial\Omega_N}=0\},\\ Q&=L^2(\Omega),\end{aligned}$$

for flux $\boldsymbol{u}$ and pressure p, respectively.

The variational formulation of (2.2)–(2.5) states: Find $\boldsymbol{u}\in\boldsymbol{V}$ and $p\in Q$ such that

$$(\boldsymbol{K}^{-1}\boldsymbol{u},\boldsymbol{v})_\Omega+(\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\boldsymbol{u}\cdot\boldsymbol{\nu},\boldsymbol{v}\cdot\boldsymbol{\nu})_\Gamma-(p,\mathbf{D}\cdot\boldsymbol{v})_\Omega=-(g,\boldsymbol{v}\cdot\boldsymbol{\nu})_{\partial\Omega_D}\qquad\forall\boldsymbol{v}\in\boldsymbol{V}, \tag{2.11a}$$

$$(\mathbf{D}\cdot\boldsymbol{u},q)_\Omega=(f,q)_\Omega\qquad\forall q\in Q, \tag{2.11b}$$

with $f\in L^2(\Omega)$ and $g\in H^{\frac{1}{2}}(\partial\Omega_D)$. To show the uniqueness of the solution, we rewrite the system (2.11) and make use of the theory of saddle point problems.

Define bilinear forms $a(\cdot,\cdot):\boldsymbol{V}\times\boldsymbol{V}\to\mathbb{R}$ and $b(\cdot,\cdot):\boldsymbol{V}\times Q\to\mathbb{R}$ as

$$a(\boldsymbol{u},\boldsymbol{v})=(\boldsymbol{K}^{-1}\boldsymbol{u},\boldsymbol{v})_\Omega+(\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\boldsymbol{u}\cdot\boldsymbol{\nu},\boldsymbol{v}\cdot\boldsymbol{\nu})_\Gamma, \tag{2.12a}$$

$$b(\boldsymbol{v},p)=-(p,\mathbf{D}\cdot\boldsymbol{v})_\Omega. \tag{2.12b}$$

Then the saddle point form of system (2.11) reads: Find $(\boldsymbol{u},p)\in\boldsymbol{V}\times Q$ such that

$$a(\boldsymbol{u},\boldsymbol{v})+b(\boldsymbol{v},p)=-(g,\boldsymbol{v}\cdot\boldsymbol{\nu})_{\partial\Omega_D},\qquad\forall\,\boldsymbol{v}\in\boldsymbol{V}, \tag{2.13a}$$

$$b(\boldsymbol{u},q)=-(f,q)_\Omega,\qquad\forall\,q\in Q. \tag{2.13b}$$

In order to prove that the above problem admits a unique solution to the , two conditions on the bilinears forms a and b are sufficient:

1. (Coercivity): There exists a constant $C_a>0$ such that it holds

$$a(\boldsymbol{v},\boldsymbol{v})\geq C_a\|\boldsymbol{v}\|_{\boldsymbol{V}}^2, \tag{2.14}$$

   for $\boldsymbol{v}\in\boldsymbol{V}$ such that $b(\boldsymbol{v},q)=0$ for any $q\in Q$.

2. (Inf-sup): There exists a constant $C_b>0$ such that it holds

$$\inf_{q\in Q}\sup_{\boldsymbol{v}\in\boldsymbol{V}}\frac{b(\boldsymbol{v},q)}{\|\boldsymbol{v}\|_{\boldsymbol{V}}\|q\|_Q}\geq C_b. \tag{2.15}$$

In [16] it has been shown that the these conditions are indeed satisfied and, following the classical Brezzi theory [13, 18], we conclude that the saddle point system (2.13) is well-posed. That is, the unique solution of (2.13) exists, which is by equivalence the unique solution of (2.11).

2.2.2 Conforming Discretization

The dual variational formulation (2.13) describes the solution as two variables, which motivates the same structure of the solution in the discrete sense. Therefore, the discretization schemes we mainly consider are mixed finite element pairs that satisfy discrete versions of the conditions (2.14) and (2.15).

Let $\mathcal{T}_\Omega^d$ be a d-dimensional shape-regular simplicial tesselation of Ω^d and $\mathcal{T}_\Omega = \bigcup_{d=0}^{n} \mathcal{T}_\Omega^d$ the complete mixed-dimensional grid. The grids are constructed so that they are matching along the interface. Let $h = \max_{0\le d\le n} h^d$ be the characteristic mesh size parameter the subscript h describe the discrete entities. Consider $\boldsymbol{V}_h \subset \boldsymbol{V}$ and $Q_h \subset Q$ to be the lowest-order stable mixed finite element approximations on $\mathcal{T}_\Omega$: $\boldsymbol{V}_h = \mathbb{RT}_0(\mathcal{T}_\Omega)$ as the lowest-order Raviart-Thomas-Nédélec spaces [46, 54] and $Q_h = \mathbb{P}_0(\mathcal{T}_\Omega)$ as the space of piecewise constant polynomials.

The finite element approximation of the system (2.11) is formulated as follows: Find $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h$ such that,

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h) = -(g, \boldsymbol{v}_h \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{2.16a}$$

$$b(\boldsymbol{u}_h, q_h) = -(f, q_h)_\Omega, \qquad \forall\, q_h \in Q_h. \tag{2.16b}$$

Due to our choice of the finite element spaces, the conditions on the bilinear forms are preserved directly. For further details, we refer the reader to [13, 15, 16, 18].

Finally, we give a block formulation of the discrete saddle point system (2.16). Denote $\boldsymbol{V}_h'$ and Q_h' to be the dual spaces of $\boldsymbol{V}_h$ and Q_h, respectively, and $\langle\cdot,\cdot\rangle$ the duality pairing. We define the linear operators $A_{\boldsymbol{u}} : \boldsymbol{V}_h \to \boldsymbol{V}_h'$ and $B : \boldsymbol{V}_h \to Q_h'$ such that

$$\langle A_{\boldsymbol{u}} \boldsymbol{u}_h, \boldsymbol{v}_h\rangle = a(\boldsymbol{u}_h, \boldsymbol{v}_h) \qquad \boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{2.17a}$$

$$\langle B\boldsymbol{v}_h, p_h\rangle = b(\boldsymbol{v}_h, p_h) \qquad p_h \in Q_h. \tag{2.17b}$$

Then, (2.16) takes the following equivalent form

$$\mathcal{A}\begin{pmatrix}\boldsymbol{u}_h\\ p_h\end{pmatrix} = \begin{pmatrix}G\\ F\end{pmatrix} \quad \text{with } \mathcal{A} = \begin{pmatrix}A_{\boldsymbol{u}} & B^T\\ -B & 0\end{pmatrix}, \tag{2.18}$$

where the right hand side is defined as $G(\boldsymbol{v}_h) = -(g, \boldsymbol{v}_h \cdot \boldsymbol{\nu})_{\partial\Omega_D}$ and $F(q_h) = (f, q_h)_\Omega$.

The well-posedness conditions (2.14)–(2.15) ensure that $\mathcal{A}$ is an isomorphism from $\boldsymbol{V}_h \times Q_h$ to $\boldsymbol{V}_h' \times Q_h'$, therefore the unique solution to (2.16) is also the unique solution to (2.18).

Although we have concluded that the unique solution of the linear system (2.18) exists, the question of finding it is still open. Looking back at the construction of this linear system, we see that it is highly coupled over interfaces of subdomains, its underlying fracture network geometry can be complex and simultaneously contain very long and very small fractures, and it incorporates physical parameters that can vary spatially on several orders of magnitude. This makes the linear system (2.18) very difficult to solve. Thus, to attain the solution efficiently, we have to address all of these features in the numerical methods that we use. Such methods are provided mainly in Paper A and B and shortly described in Chapter 3, more specifically in Section 3.2.

2.2.3 Non-matching Grids

The advantage of having separate manifolds for each subdomains is that we can construct separate grids that do not necessarily match on the interface. This setting can be desirable if certain parts of the mixed-dimensional domain Ω require finer meshing while the rest should remain unchanged and even allows coarsening. For example, we can have a more heterogeneous and non-linear flow patterns within the fractures, while the rock matrix flow still exhibits a slower and stable profile. Here, we give a concise presentation on how the flow problem (2.2)–(2.5) can be modified to incorporate non-matching grids, while a more detailed description and analysis can be found in e.g. [16, 37, 48].

To couple the non-matching grids, we introduce an auxiliary mortar variable λ on Γ. Specifically, λ_{ij}^d represents the flux across Γ_{ij}^d between the two subdomains Ω_i^d and Ω_j^{d+1}. Therefore, we can define the mortar variable as

$$\lambda_{ij}^d = (\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu})|_{\Gamma_{ij}^d}, \qquad j \in J_i^d,\ i \in I^d,\ 0 \le d \le n-1. \tag{2.19}$$

Based on the choice of function space for $\boldsymbol{u}$, we have that λ lies in the space $\Lambda = L^2(\Gamma)$. Taking $\boldsymbol{V}_0 = \{\boldsymbol{v} \in \boldsymbol{V} : (\boldsymbol{v} \cdot \boldsymbol{\nu})|_\Gamma = 0\}$, the variational formulation (2.11) can equivalently be written as: Find $\boldsymbol{u} = (\boldsymbol{u}_0, \lambda) \in \boldsymbol{V}_0 \times \Lambda$ and $p \in Q$ such that

$$(\boldsymbol{K}^{-1}(\boldsymbol{u}_0 + \mathcal{R}\lambda), \boldsymbol{v}_0 + \mathcal{R}\mu)_\Omega + (\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\lambda, \mu)_\Gamma - (p, \mathbf{D} \cdot \boldsymbol{v})_\Omega = -(g, \boldsymbol{v}_0 \cdot \boldsymbol{\nu})_{\partial\Omega_D}$$
$$\forall \boldsymbol{v} \in \boldsymbol{V}_0 \times \Lambda,\ \boldsymbol{v} = (\boldsymbol{v}_0, \mu), \tag{2.20a}$$
$$(\mathbf{D} \cdot \boldsymbol{u}, q)_\Omega = (f, q)_\Omega \quad \forall q \in Q, \tag{2.20b}$$

where $\mathcal{R}$ is an extension operator from Λ to $\boldsymbol{V} \cdot \boldsymbol{\nu}$, the space of normal traces of functions in $\boldsymbol{V}$. Additionally, the saddle point structure in (2.13) stays the same considering the bilinear forms

$$a(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{K}^{-1}(\boldsymbol{u}_0 + \mathcal{R}\lambda), \boldsymbol{v}_0 + \mathcal{R}\mu)_\Omega + (\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\lambda, \mu)_\Gamma, \tag{2.21a}$$
$$b(\boldsymbol{v}, p) = -(p, \mathbf{D} \cdot \boldsymbol{v})_\Omega = -(p, \nabla \cdot \boldsymbol{v}_0 + [\![\mu]\!])_\Omega. \tag{2.21b}$$

Now, let $\mathcal{T}_\Gamma = \bigcup_{d=0}^{n-1} \mathcal{T}_\Gamma^d$ represent the additional grids on Γ. Notice that the normal traces of the flux variable $\boldsymbol{v}_h \in V_h$ are piecewise constant, so we can take the discrete approximation of the function space Λ as $\Lambda_h = \mathbb{P}_0(\mathcal{T}_\Gamma)$. Let $\boldsymbol{V}_{0h}$ represent the functions in $\boldsymbol{V}_h$ with zero normal trace on Γ. The discrete formulation is then equivalent to (2.16) considering the modified bilinear forms (2.21) and the solution in spaces $\boldsymbol{u}_h = (\boldsymbol{u}_{0h}, \lambda_h) \in \boldsymbol{V}_{0h} \times \Lambda_h$ and $p_h \in Q_h$. This formulation is considered in Paper A.

On the other hand, it is also possible to use the pressure in the lower-dimensional subdomains as the coupling mortar variable. That is, in the interface equation (2.4) we can instead take

$$\lambda_i^d = p_i^d, \qquad i \in I^d,\ 0 \le d \le n-1. \tag{2.22}$$

Then, the mortar function space Λ is a subspace of the pressure space Q, or more precisely

$$Q = Q^n \times \Lambda = L^2(\Omega^n) \times \prod_{d=0}^{n} L^2(\Omega^d). \tag{2.23}$$

Notice that it is not necessary to extend λ to Γ since in the variational formulation we use the jump operator $[\![\cdot]\!]$ to map the normal flux $\boldsymbol{u} \cdot \boldsymbol{\nu}$ to the lower-dimensional subdomains. Specifically, the bilinear form b is modified to

$$b(\boldsymbol{v}, p) = -(p^n, \nabla \cdot \boldsymbol{v}^n)_{\Omega^n} - \sum_{0<d<n} (\lambda^d, \nabla \cdot \boldsymbol{v}^d)_{\Omega^d} - \sum_{0 \leq d<n} (\lambda^d, [\![\boldsymbol{v}^{d+1} \cdot \boldsymbol{\nu}]\!])_{\Omega^d}, \tag{2.24}$$

for $\boldsymbol{v} \in \boldsymbol{V}$ and $p = (p^n, \lambda) \in Q$ as in (2.23).

Although not immediately following from the variational problem in Section 2.2.1, this approach is more common in modeling fractures as interfaces [3, 27]. In fact, it leads to the traditional domain decomposition setting based on imposing the continuity of pressure over the interface, and it allows to reduce the system (2.16) to an interface problem. We describe this setting in Section 3.3 and use it in Papers C and D.

2.3 Non-linear and Time-Dependent Flow Models

The linear Darcy's law (2.2a) is most commonly used as the constitutive law for flow in (fractured) porous media. However, it does not account for all flow patterns, such as the ones influenced by high-velocity and non-linear flow or the fluid compressibility. Here, we give an overview of several other flow models that are considered in this thesis, namely in Papers C and D.

2.3.1 Non-linear Fracture Flow Models

The fractures often exhibit a higher hydraulic conductivity than in the surrounding medium and consequently the fracture flow may dominate the total flow process. However, due to a higher flow velocity, the fracture flow patterns may deviate from the standard Darcy's law, which indicates deriving different flow models in the fractures. In particular, a non-linear flow behavior can be observed in case of high-velocity flows. The simplest proposed models concern correcting the permeability term in (2.2a) to

$$\boldsymbol{k}(\boldsymbol{u}) = \boldsymbol{K}^{-1} + \xi(\boldsymbol{u}). \tag{2.25}$$

This influence the form $a(\cdot, \cdot)$ in (2.12a) that now becomes

$$a(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{k}(\boldsymbol{u})\boldsymbol{u}, \boldsymbol{v})_\Omega + (\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\boldsymbol{u} \cdot \boldsymbol{\nu}, \boldsymbol{v} \cdot \boldsymbol{\nu})_\Gamma. \tag{2.26}$$

Depending on the choice of the correction term $\xi(\boldsymbol{u})$, we derive different flow models. One of the common corrections is the Forchheimer term

$$\xi(\boldsymbol{u}) = \zeta |\boldsymbol{u}| \boldsymbol{I}, \tag{2.27}$$

where $\zeta > 0$ is a scalar Forchheimer coefficient and $|\cdot|$ denotes the Euclidean norm. Combining (2.27) in fracture subdomains Ω_i^d for $d < n$ with the standard Darcy's law given in (2.2a) in the rock matrix subdomains $d = n$, the new flow model can be identified as the

Darcy-Forchheimer flow model for fractured porous media [28, 38]. Moreover, a more general law that incorporates changes in fluid viscosity considers a Cross' law [24] correction term

$$\xi(\boldsymbol{u}) = \frac{\upsilon_0 - \upsilon_\infty}{1 + \boldsymbol{K}|\boldsymbol{u}|^{2-r}}, \tag{2.28}$$

with $\upsilon_0, \upsilon_\infty$ and r are fluid-type dependent physical parameters. Many other flow models are possible, including multiphase and unsaturated flow [2, 9, 31, 42], however they are outside the scope of this thesis.

2.3.2 Compressible Flow Model

Another modification to the flow model is to consider the changes in fluid density or the rock porosity in time [49, Section 2.2]. In short, it comprises an additional compressibility term for pressure in the mass conservation equation (2.2b),

$$s_0 \frac{\partial p^d}{\partial t} + \nabla \cdot \boldsymbol{u}^d + [\![\boldsymbol{u}^{d+1} \cdot \boldsymbol{\nu}]\!] = f^d, \qquad \text{in } I \times \Omega^d, \quad 0 \le d \le n. \tag{2.29}$$

where $I = (0, T]$ is given time interval and s_0 is the specific storativity constant. To complete the model, we need to set an initial condition on the pressure, i.e.

$$p^d(0, \cdot) = p_0^d, \qquad \text{in } \Omega^d, \quad 0 \le d \le n. \tag{2.30}$$

For the variational formulation, we need to extend the spatial function spaces to include the time component. The time derivative only concerns the pressure variable, therefore we search for solution $(\boldsymbol{u}, p)$ in the space $L^2(0, T; V) \times H^1(0, T; Q)$. With defining the bilinear form $c : Q \times Q \to \mathbb{R}$ as

$$c(p, q) = (s_0 p, q)_\Omega \qquad p, q \in Q, \tag{2.31}$$

the updated saddle-point formulation (2.13) takes the following form

$$a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) = -(g, \boldsymbol{v} \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{v} \in \boldsymbol{V}, \tag{2.32a}$$

$$c(\partial_t p, q) + b(\boldsymbol{u}, q) = -(f, q)_\Omega, \qquad \forall\, q \in Q. \tag{2.32b}$$

with $p_0 \in L^2(\Omega)$, $f \in L^2(0, T; L^2(\Omega))$ and $g \in L^2(0, T; H^{\frac{1}{2}}(\partial\Omega_D))$.

Similarly, we can extend the mixed finite element spatial discretization to approximate the time derivative of the pressure with the backward Euler method. For an integer M, let $(\tau^m)_{0 \le m \le M}$ be the discrete time steps such that $T = \sum_{m=1}^{M}$. The discrete times are set as $t^0 = 0$ and $t^m = \sum_{j=1}^{m} \tau^j$ with discrete time intervals $I^m = (t^{m-1}, t^m]$. The fully discrete formulation of (2.32) states that assuming $p_h^{(m-1)}$ is given, at each time step iteration $m \ge 1$ find $(\boldsymbol{u}_h^{(m)}, p_h^{(m)}) \in \boldsymbol{V}_h \times Q_h$ such that

$$a(\boldsymbol{u}_h^{(m)}, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h^{(m)}) = -(g, \boldsymbol{v}_h \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{2.33a}$$

$$c(p_h^{(m)} - p_h^{(m-1)}, q_h) + \tau^m b(\boldsymbol{u}_h^{(m)}, q_h) = -\tau^m (f, q_h)_\Omega, \qquad \forall\, q_h \in Q_h. \tag{2.33b}$$

Still, the system (2.33) is non-linear because of (2.26) and, to find the solution at each time step, we first need to linearize it. In Papers C and D we provide several linearization methods designed for solving (2.33), which we also describe in short in Section 3.4.

Chapter 3

Solvers and Preconditioners

To simulate the mathematical models described in the previous chapter, we rely on linear and non-linear *solvers*, that is the numerical methods used to compute the solution of linear or non-linear systems of equations. Based on the choice of discretization and basis functions, the systems of partial differential equations turn into large sparse algebraic systems that are often difficult to solve using classical solving approaches. In our case, the geometry of the fractured rock is often complex and the models can contain heterogeneous physical parameters, which largely affect the structure and the properties of the resulting algebraic system.

To overcome this issue, a design of problem-specific solvers is required. In this chapter, we propose several *direct* and *iterative* approaches to solving the linear system (2.18), but also the *linearization* schemes to solving (2.33). In particular, we give focus on improving the performance of the standard iterative methods by applying *preconditioners* in each iteration of the chosen method.

In Section 3.1, we first introduce what the direct and iterative methods are, as well as how preconditioners come into play, and present several standard approaches often used in general applications. The main contribution of Section 3.2 is a class of preconditioners specifically designed for the discretizations of the model given in Section 2.2. The saddle-point structure of the mixed finite element problem (2.18) gives rise to parameter-independent, namely *robust*, preconditioners based on the well-posedness of the problem. Finally, after introducing the concept of domain decomposition, in Section 3.4 we present the non-linear solvers for non-linear models that were given in Section 2.3.

As for the implementation, the solvers proposed in this thesis are mainly implemented using two softwares developed in two programming languages. One of them is PorePy [37] that is based in Python and provides the mixed-dimensional geometry and discretization of the models presented in the previous chapter. Also, we use Python to implement the non-linear solvers. On the other hand, the iterative methods and preconditioners given later in Section 3.2 are developed using HAZMATH [1] solver library that is built in C.

3.1 Direct and Iterative Methods

In this section, we give a short overview of some classical numerical methods to find the solution $x \in \mathbb{R}^n$ to the algebraic linear system

$$\mathsf{A}\mathsf{x} = \mathsf{b}, \tag{3.1}$$

where $\mathsf{b} \in \mathbb{R}^n$ given and $\mathsf{A} \in \mathbb{R}^{n \times n}$ is an invertible *sparse* matrix. The sparse structure is common in many applications when numerically solving partial differential equations, for example when using finite element method where basis functions have small support. In large-scale systems, this is beneficial since sparse matrices require less storage space, but it is often necessary to use specialized algorithms to take advantage of the sparsity patterns in the system.

Depending on the properties of the matrix A, many solving techniques have been developed over the years to approximate the solution $\mathsf{x} = \mathsf{A}^{-1}\mathsf{b}$ that can be classified into *direct* and *iterative* methods [32, 33].

3.1.1 Sparse Direct Solvers

The direct methods for solving (3.1) concern computing the solution with a finite number of operations, which is exact up to rounding errors. The most common direct computation of x is to use the sparse *LU decomposition* of the matrix A. Although the standard LU decomposition has $\approx \frac{2}{3}n^3$ computational cost, the sparse version of the algorithm can significantly reduce the required number of floating-point operations and memory usage. It can be found in many linear algebra libraries, such as SuperLU [41], SuiteSparse [21] and Pardiso [40]. Specifically, the software PorePy [37], as one of the implementation tools in our work, uses the Python library SciPy [36] that has an interface with the SuperLU library. Hence, in the following we give a simply description of the algorithm.

Using sparse Gaussian eliminations, the algorithm is given in two parts:

1. Compute the factorization $\mathsf{P}_r\mathsf{D}_r\mathsf{A}\mathsf{D}_c\mathsf{P}_c = \mathsf{L}\mathsf{U}$, where we have premultiplied A with row and column permutation matrices P_r and P_c and scaled rows and columns with diagonal matrices D_r and D_c. These matrices are specially constructed to improve numerical stability and parallelism of the factorization. The factorization itself produces a lower triangular matrix L, with ones on the diagonal, and an upper triangular matrix U.

2. Compute the solution as

$$\mathsf{x} = (\mathsf{D}_r^{-1}\mathsf{P}_r^{-1}\mathsf{L}\mathsf{U}\mathsf{P}_c^{-1}\mathsf{D}_c^{-1})^{-1}b = (\mathsf{D}_c(\mathsf{P}_c(\mathsf{U}^{-1}(\mathsf{L}^{-1}(\mathsf{P}_r(\mathsf{D}_r\mathsf{b})))))).$$

The sparse direct methods show a solid performance in common applications. However, the major drawback of direct methods is still the excessive computational cost that occurs in large-scale simulations in comparison to iterative methods that exhibit $\approx n^2$ complexity or better.

3.1.2 Iterative Methods

To find an iterative solution of the system (3.1) means to generate a sequence of approximations $\{\mathbf{x}^{(k)}\}$ for $k = 1, 2, \ldots$ from the initial guess $\mathbf{x}^{(0)}$, where each $x^{(k)}$ is closer to the exact solution $\mathbf{x}$. We say that the iterative solution has converged if it approximates the exact solution up to a prescribed tolerance, usually measured in the magnitude of the residual $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$.

The iterative methods exploit the fast matrix-vector product computation. Efficient iterative methods usually require only a very few products with the stiffness matrix $\mathbf{A}$, while also keeping the number of iterations needed to reach convergence minimal. This is typically done by transforming the original system (3.1) to

$$\mathbf{B}\mathbf{A}\mathbf{x} = \mathbf{B}\mathbf{b}, \tag{3.2a}$$

or

$$\mathbf{A}\mathbf{B}\mathbf{y} = \mathbf{b}, \quad \mathbf{x} = \mathbf{B}\mathbf{y}. \tag{3.2b}$$

which may be more suitable for iterative computations. The matrix $\mathbf{B}$ is called a *preconditioner* matrix and should in some way approximate the inverse of $\mathbf{A}$. The form (3.2a) gives a *left preconditioner* and (3.2b) a *right preconditioner*. In Section 3.2 we also present the preconditioners in a more general setting regarding linear operators defined on Hilbert spaces.

Considering this, the basic iterative methods then solve the system

$$\mathbf{x}^{(k)} = \mathbf{x}^{(k-1)} + \mathbf{B}(\mathbf{b} - \mathbf{A}\mathbf{x}^{(k-1)}), \quad m \geq 0. \tag{3.3}$$

If we take $\mathbf{B} = \mathbf{A}^{-1}$, we see that the method will converge to the solution $\mathbf{x}$ in one iteration. Therefore, a requirement that the application of the preconditioner should closely resemble to an application of the inverse of the system matrix is highly beneficial. In the following, we explain how this is achieved in some classical iterative techniques that have also served as a tool in the work of this thesis.

Stationary iterative methods. To find $\mathbf{B}$, these methods typically consider a certain splitting of the matrix $\mathbf{A}$. For example, let $\mathbf{A}$ allow the following splitting

$$\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}, \tag{3.4}$$

where $\mathbf{D}$ is the diagonal of $\mathbf{A}$, and $\mathbf{L}$ and $\mathbf{U}$ are the strictly lower and upper triangular part of $\mathbf{A}$, respectively. Depending on the choice of $\mathbf{B}$, we can get different iterative methods, such as *Jacobi method* for $\mathbf{B} = \omega\mathbf{D}^{-1}$ and *Gauss-Seidel method* for $\mathbf{B} = (\omega^{-1}\mathbf{D} + \mathbf{L})^{-1}$, for given $\omega > 0$. The weight constant ω is chosen so that the iterative methods are guaranteed to converge, although it is usually limited to a small class of matrices. See, for example, [33] for more details on the convergence of the stationary iterative methods. A more efficient and often better approximation $\mathbf{B}$ can be derived by constructing a hierarchy of grids.

Algebraic multigrid method (AMG). The AMG method was developed to achieve an automatic coarsening process based on the underlying matrix of the given system of equations. Instead of operating on a hierarchy of geometrical grids, as the case in geometric multigrid methods, AMG can directly be applied to problems with complex unstructured meshes and jumping coefficients. Although applications are many, it has been mostly developed to efficiently solve systems of equations that are symmetric positive definite. Therefore, we present in short the main idea of AMG on (sparse) symmetric positive definite problems and refer to the relevant literature [61, 67] for more information on the topic.

We focus on two-level methods in an operator setting and assume that the extension to a multi-level method (e.g. V-cycle or W-cycle algorithms) can be done by recursively applying the two-level algorithm. Let V a finite-dimensional vector space (usually $V = \mathbb{R}^n$ in matrix setting) and V' its dual space (usually $V = V'$). Let $\mathsf{x} \in V$ be the solution of the linear system

$$\mathsf{A}\mathsf{x} = \mathsf{b}, \tag{3.5}$$

with $\mathsf{b} \in V'$ and $\mathsf{A} : V \to V'$. The main ingredients of a two-level AMG method to solve the above linear system are:

- A smoother $\mathsf{R} : V' \to V$, as a relaxation operator;
- A coarse space V_c, not necessarily a subspace of V;
- A prolongation operator $\mathsf{P} : V_c \to V$;
- A coarse space solver $\mathsf{B}_c : V_c' \to V_c$, an approximation of the inverse of $\mathsf{A}_c = \mathsf{P}^T\mathsf{A}\mathsf{P}$.

A two-level AMG algorithm can be characterized as a application of an operator $\mathsf{B} : V' \to V$ that should approximate A^{-1}. Given $\mathsf{b} \in V'$, the action $\mathsf{B}\mathsf{b}$ is given with

1. Compute a coarse grid correction: $\mathsf{w} = \mathsf{P}\mathsf{B}_c\mathsf{P}^T\mathsf{b}$;
2. Smoothen the residual: $\mathsf{B}\mathsf{b} = \mathsf{w} + \mathsf{R}(\mathsf{b} - \mathsf{A}\mathsf{w})$.

If $V_c \subset V$, the prolongation operator P can be chosen as the inclusion $\iota_c : V_c \to V$. Then, the error propagation operator of the two-level AMG is

$$\mathsf{E} = \mathsf{I} - \mathsf{B}\mathsf{A} = (\mathsf{I} - \mathsf{R}\mathsf{A})(\mathsf{I} - \Pi_c), \tag{3.6}$$

where $\Pi_c = \iota_c\mathsf{A}_c^{-1}\iota_c^T\mathsf{A}$. Therefore, to design a fast converging AMG method, we need to minimize $\|\mathsf{E}\|_{\mathsf{A}}$. Without going into a detailed convergence analysis, see e.g. [67, Section 5.2], it can already be observed from (3.6) that the key to an efficient AMG method is to balance the interplay between the smoother R and the coarse space V_c. A common choice is to fix the smoother, such as Jacobi or Gauss-Seidel method, and then choose a suitable coarse space that minimizes $\|\mathsf{E}\|_{\mathsf{A}}$.

However, it is often simpler and more efficient to accelerate the multigrid methods by using them as preconditioners in other iterative methods, such as the *Krylov subspace methods* which we introduce directly after. This way, we can save on optimizing the interplay between multigrid components but rather improve the convergence of the chosen iterative method. We consider this approach in some of the work in this thesis, for example in Paper A and B.

Krylov subspace methods. The principal idea of these methods is to produce a sequence $\{\mathbf{x}^{(k)}\}$ of approximative solutions where we look for each $\mathbf{x}^{(k)}$ in the Krylov subspace of order k

$$\mathcal{K}_k(\mathbf{A}, \mathbf{b}) = \{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{k-1}\mathbf{b}\}, \tag{3.7}$$

by minimizing the residual over the subspace. Particularly, if $\mathbf{A}$ is symmetric positive definite, the unique solution to (3.1) can be characterized as a minimization problem

$$\mathbf{x} = \arg\min_{\mathbf{y} \in \mathbb{R}^n} \mathsf{r}(\mathbf{y}), \tag{3.8}$$

where $\mathsf{r}(\mathbf{y}) = \mathbf{y}^T\mathbf{A}\mathbf{y} - 2\mathbf{y}^T\mathbf{b}$. The k-th approximation is then a solution $\mathbf{x}^{(k)} \in \mathcal{K}_k$ to (3.8) where we have minimized over all $\mathbf{y} \in \mathcal{K}_k$. In fact, it means that each $\mathbf{x}^{(k)}$ is the best approximation of $\mathbf{x}$ in $\mathcal{K}_k$.

The rate of the convergence of Krylov subspace methods is usually characterized by the condition number $\kappa(\mathbf{A}) = |||\mathbf{A}|||\,|||\mathbf{A}^{-1}|||$ for some consistent matrix norm $|||\cdot|||$. For example, a method resulting from (3.8) called the *conjugate gradient method* (CG) [34] generates a sequence $\{\mathbf{x}^{(k)}\}$ such that

$$\|\mathbf{x} - \mathbf{x}^{(k)}\|_{\mathbf{A}} \leq 2\left(\frac{\sqrt{\kappa(\mathbf{A})} - 1}{\sqrt{\kappa(\mathbf{A})} + 1}\right)^k \|\mathbf{x} - \mathbf{x}^{(0)}\|_{\mathbf{A}} \tag{3.9}$$

for any initial guess $\mathbf{x}^{(0)}$ and $\|\mathbf{y}\|_{\mathbf{A}}^2 = \mathbf{x}^T\mathbf{A}\mathbf{y}$.

Similar methods exist for matrices that are not symmetric positive definite. For indefinite matrices, one can use *minimal residual method* (MINRES) [50]. If they are also not symmetric, *generalized minimal residual* (GMRES) [55] is the most common choice. In those cases, it is possible to derive a similar estimate on the convergence rate as in (3.9), that depends only on $\kappa(\mathbf{A})$. In particular, the systems that we consider fit in these categories, therefore we discuss the implementation of MINRES and GMRES to our problems in Section 3.2.

One can significantly accelerate the convergence of the Krylov subspace methods by using preconditioners. If N is the dimension of the solution space, then $\mathcal{K}_N$ builds a basis for the whole space, which consequently means that the Krylov subspace methods are guaranteed to converge in at most N iterations. On the other hand, as also representing the number of unknowns in the system, N can get very large in general applications. Thus, the aim of the preconditioners is to affect the rate of convergence, as seen in (3.9), by directly reducing the condition number of the system (hence, "*pre-conditioning*").

These preconditioners are usually problem-specific, however their construction often incorporates (a combination of) standard solving techniques, such as the ones mentioned earlier in this section. The following section, though, provides a general approach to designing the preconditioners to Krylov subspace methods for saddle point problems, and how it can be applied to our system (2.18).

3.2 Block Preconditioners

The block formulation (2.18) naturally motivates a preconditioning framework that treats each block of the coefficient operator of the linear system. We first introduce a general setting of this framework as preconditioners for Krylov subspace methods that are based on the well-posedness of the underlying discrete problem. We show how our linear system fits within the framework and present two approaches to block preconditioners that can be applied to the system. Most importantly, we present methods to implement the preconditioners in a computationally feasible and efficient way.

3.2.1 The General Theory

To design the block preconditioners, we require some analytical tools in a more general saddle point problem setting given in [45]. Assume that we have a real separable Hilbert space $\boldsymbol{X}$ with an inner product $(\cdot,\cdot)_{\boldsymbol{X}}$ that induces the norm $\|\cdot\|_{\boldsymbol{X}}$. Let $\boldsymbol{X}'$ be its dual space and let $\langle\cdot,\cdot\rangle$ the duality pairing between them. Also, assume that there is a symmetric bilinear form $\mathcal{L}(\cdot,\cdot)$ on $\boldsymbol{X}$ that satisfies the well-posedness conditions similar to (2.14)–(2.15),

$$|\mathcal{L}(\boldsymbol{x},\boldsymbol{y})| \leq \alpha\|\boldsymbol{x}\|_{\boldsymbol{X}}\|\boldsymbol{y}\|_{\boldsymbol{X}}, \quad \text{and} \quad \inf_{\boldsymbol{x}\in\boldsymbol{X}}\sup_{\boldsymbol{y}\in\boldsymbol{X}}\frac{\mathcal{L}(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|_{\boldsymbol{X}}\|\boldsymbol{y}\|_{\boldsymbol{X}}} \geq \beta, \quad \forall \boldsymbol{x},\boldsymbol{y}\in\boldsymbol{X}, \tag{3.10}$$

for $\alpha,\beta>0$. Denote a right hand side $\boldsymbol{b}\in\boldsymbol{X}'$ and a linear operator $\mathcal{A}:\boldsymbol{X}\to\boldsymbol{X}'$ induced by the bilinear form $\mathcal{L}(\cdot,\cdot)$ such that $\langle\mathcal{A}\boldsymbol{x},\boldsymbol{y}\rangle=\mathcal{L}(\boldsymbol{x},\boldsymbol{y})$. We aim to iteratively solve the linear system

$$\mathcal{A}\boldsymbol{x}=\boldsymbol{b}. \tag{3.11}$$

As mentioned before, efficient methods for such systems are the Krylov subspace methods that often require preconditioners to obtain faster convergence. In this case, the conditions (3.10) ensure that $\mathcal{A}$ is a bounded, symmetric and isomorphic linear operator and the system (3.11) has a unique solution. However, $\mathcal{A}$ maps the functions in $\boldsymbol{X}$ to the dual space $\boldsymbol{X}'$, which is often much bigger than the original space, so the convergence of the Krylov subspace methods is not well defined. Thus, the preconditioner applied to (3.11) should be an isomorphic mapping from $\boldsymbol{X}'$ back to $\boldsymbol{X}$.

Consider a symmetric positive definite linear operator $\mathcal{B}:\boldsymbol{X}'\to\boldsymbol{X}$ which induces an inner product $(\boldsymbol{x},\boldsymbol{y})_{\mathcal{B}^{-1}}=\langle\mathcal{B}^{-1}\boldsymbol{x},\boldsymbol{y}\rangle$ on $\boldsymbol{X}$ and corresponding norm $\|\boldsymbol{x}\|^2_{\mathcal{B}^{-1}}=(\boldsymbol{x},\boldsymbol{x})_{\mathcal{B}^{-1}}$. In turn, $\mathcal{B}\mathcal{A}:\boldsymbol{X}\to\boldsymbol{X}$ is a symmetric isomorphism with respect to $(\cdot,\cdot)_{\mathcal{B}^{-1}}$, and the preconditioned system

$$\mathcal{B}\mathcal{A}\boldsymbol{x}=\mathcal{B}\boldsymbol{b} \tag{3.12}$$

can be solved with the MINRES method. Consequently, if $\mathcal{B}$ also satisfies

$$c_1\|\boldsymbol{x}\|^2_{\boldsymbol{X}} \leq \|\boldsymbol{x}\|^2_{\mathcal{B}^{-1}} \leq c_2\|\boldsymbol{x}\|^2_{\boldsymbol{X}}, \tag{3.13}$$

then $\mathcal{A}$ and $\mathcal{B}$ are referred to as *norm equivalent* and, with (3.10), we can get a bound on the condition number of the system as

$$\kappa(\mathcal{B}\mathcal{A}) \leq \|\mathcal{B}\mathcal{A}\|_{\mathscr{L}(\boldsymbol{X},\boldsymbol{X})}\|(\mathcal{B}\mathcal{A})^{-1}\|_{\mathscr{L}(\boldsymbol{X},\boldsymbol{X})} \leq \frac{c_2\alpha}{c_1\beta}. \tag{3.14}$$

Here, $\mathscr{L}(\boldsymbol{X}, \boldsymbol{X})$ denotes the set of all bounded linear operators on $\boldsymbol{X}$ and, for any $\tilde{\mathcal{A}} \in \mathscr{L}(\boldsymbol{X}, \boldsymbol{X})$, the corresponding operator norm is defined as $\|\tilde{\mathcal{A}}\|_{\mathscr{L}(\boldsymbol{X},\boldsymbol{X})} = \sup_{\mathbf{0}\neq \boldsymbol{x}\in\boldsymbol{X}} \dfrac{\|\tilde{\mathcal{A}}\boldsymbol{x}\|_{\boldsymbol{X}}}{\|\boldsymbol{x}\|_{\boldsymbol{X}}}$.

The natural choice of $\mathcal{B}$ is the *Riesz operator* $\mathcal{B}_D$ corresponding to the inner product $(\cdot,\cdot)_{\boldsymbol{X}}$,

$$(\mathcal{B}_D \boldsymbol{f}, \boldsymbol{x})_{\boldsymbol{X}} = \langle \boldsymbol{f}, \boldsymbol{x}\rangle, \qquad \forall \boldsymbol{f} \in \boldsymbol{X}',\ \boldsymbol{x} \in \boldsymbol{X}. \tag{3.15}$$

Then, it follows from the definition of the operator norm and the inner product $(\cdot,\cdot)_{\mathcal{B}^{-1}}$ that (3.13) holds with constants $c_1 = c_2 = 1$, that is $\kappa(\mathcal{B}_D\mathcal{A}) \leq \dfrac{\alpha}{\beta}$.

It is often the case in discretizations of saddle point systems that they may depend on a number of parameters, for example physical and discretization parameters. Let us assume that applies to the linear system (3.11). The direct consequence of (3.14) is that, if the well-posedness constants α,β and the norm-equivalence constants c_1, c_2 are independent of these parameters, the convergence of the MINRES method is uniform and $\mathcal{B}$ provides a *robust* preconditioner. We will show that our preconditioners for (2.18) presented later in Section 3.2.3 and Section 3.2.4 fit into that category.

In addition to the norm equivalent kind, another class of *field-of-values equivalent* (FoV-equivalent) preconditioners [22, 23, 44] can be applied in (3.12). These preconditioners usually arise from the norm equivalent ones and have similar properties, but they are not necessarily symmetric.

Consider a bounded linear operator $\mathcal{B}_L : \boldsymbol{X}' \to \boldsymbol{X}$ in the preconditioned system (3.12). If for any $\boldsymbol{x} \in \boldsymbol{X}$ the operator $\mathcal{B}_L$ satisfies

$$\xi_1 \leq \frac{(\mathcal{B}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{x})_{\mathcal{B}^{-1}}}{(\boldsymbol{x}, \boldsymbol{x})_{\mathcal{B}^{-1}}}, \quad \text{and} \quad \frac{\|\mathcal{B}_L\mathcal{A}\boldsymbol{x}\|_{\mathcal{B}^{-1}}}{\|\boldsymbol{x}\|_{\mathcal{B}^{-1}}} \leq \xi_2, \tag{3.16}$$

for constants $\xi_1, \xi_2 > 0$, then we say that $\mathcal{B}_L$ and $\mathcal{A}$ are FoV-equivalent. From [22, 23] we have that $\mathcal{B}_L$ is a left preconditioner for the GMRES method with the convergence rate

$$\|\mathcal{B}_L\mathcal{A}(\boldsymbol{x} - \boldsymbol{x}^{(k)})\|^2_{\mathcal{B}^{-1}} \leq \left(1 - \frac{\xi_1^2}{\xi_2^2}\right)^k \|\mathcal{B}_L\mathcal{A}(\boldsymbol{x} - \boldsymbol{x}^{(0)})\|^2_{\mathcal{B}^{-1}}. \tag{3.17}$$

Similarly, consider $\mathcal{B}_U : \boldsymbol{X}' \to \boldsymbol{X}$ in the preconditioned system

$$\mathcal{A}\mathcal{B}_U\boldsymbol{y} = \boldsymbol{b}, \qquad \boldsymbol{x} = \mathcal{B}_U\boldsymbol{y}, \tag{3.18}$$

and define an inner product on $\boldsymbol{X}'$ as $(\boldsymbol{x}', \boldsymbol{y}')_{\mathcal{B}} := \langle \boldsymbol{x}', \mathcal{B}\boldsymbol{y}'\rangle$, for $\boldsymbol{x}', \boldsymbol{y}' \in \boldsymbol{X}'$. Then, $\mathcal{B}_U$ and $\mathcal{A}$ are FoV-equivalent if for any $\boldsymbol{x}' \in \boldsymbol{X}'$

$$\xi_1 \leq \frac{(\mathcal{A}\mathcal{B}_U\boldsymbol{x}', \boldsymbol{x}')_{\mathcal{B}}}{(\boldsymbol{x}', \boldsymbol{x}')_{\mathcal{B}}}, \quad \text{and} \quad \frac{\|\mathcal{A}\mathcal{B}_U\boldsymbol{x}'\|_{\mathcal{B}}}{\|\boldsymbol{x}'\|_{\mathcal{B}}} \leq \xi_2, \tag{3.19}$$

and $\mathcal{B}_U$ can be used as a uniform right preconditioner for the GMRES method.

3.2.2 Application to the Mixed-Dimensional Problem

After preparing the grounds in the previous section, we continue now with applying the general theory to design the block preconditioners for our mixed-dimensional problem (2.18). Take $\boldsymbol{X} = \boldsymbol{V}_h \times Q_h$ and define an energy norm as

$$\|\boldsymbol{x}\|_{\boldsymbol{X}}^2 = \|(\boldsymbol{u}_h, p_h)\|_{\boldsymbol{X}}^2 = \|\boldsymbol{u}_h\|_{\boldsymbol{V}}^2 + \|p_h\|_Q^2, \qquad \boldsymbol{x} \in \boldsymbol{X}. \tag{3.20}$$

It follows that the coefficient operator $\mathcal{A} : \boldsymbol{X} \to \boldsymbol{X}'$ in (2.18) is induced by the bilinear form

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = \mathcal{L}((\boldsymbol{u}_h, p_h), (\boldsymbol{v}_h, q_h)) = a(\boldsymbol{u}_h, \boldsymbol{v}_h) + b(\boldsymbol{v}_h, p_h) - b(\boldsymbol{u}_h, q_h) \qquad \boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{X}. \tag{3.21}$$

Furthermore, the conditions (2.14)–(2.15) imply that $\mathcal{A}$ satisfies the well-posedness conditions (3.10), with the corresponding constants independent of physical and discretization parameters.

This formulation induces two different ways of deriving preconditioners for (2.18). Roughly speaking, one strategy is to give more weight to the matrix block concerning the flux variable $\boldsymbol{u}_h$, while the other looks more closely at the pressure block.

"Flux-based" block preconditioner. Applying directly the theory from Section 3.2.1, the Riesz operator corresponding to the norm (3.20) takes the form

$$\mathcal{B}_D = \begin{pmatrix} A_{\boldsymbol{u}} + B^T B & 0 \\ 0 & A_p \end{pmatrix}^{-1}, \tag{3.22}$$

where A_p is the identity operator with regards to the norm on Q, i.e. $\langle A_p q_h, q_h \rangle = \|q_h\|_Q^2$ for $q_h \in Q_h$. In the piecewise constant basis of the discrete space Q_h, the matrix representation of this operator is a diagonal matrix which is easy to invert. Therefore, the main challenge in implementation of this preconditioner is how to apply the inverse of the block $A_{\boldsymbol{u}} + B^T B$ that corresponds to a mixed-dimensional $H(\mathrm{div})$ problem. For small enough systems it is possible to use any direct method, but the typical applications result in a large number of degrees of freedom and the direct methods may not be a feasible solution. Another way is to approximate the inverse with another iterative method. However, to conserve the robustness of the preconditioner, we require that the approximation is spectrally equivalent.

To that end, we consider the following preconditioner

$$\mathcal{M}_D = \begin{pmatrix} M_{\boldsymbol{u}} & 0 \\ 0 & M_p \end{pmatrix}, \tag{3.23}$$

where the diagonal blocks $M_{\boldsymbol{u}}$ and M_p are symmetric positive definite and spectrally equivalent to diagonal blocks in $A + B^T B$ and A_p, respectively. Specifically, it means that

$$c_{1,\boldsymbol{u}} \langle M_{\boldsymbol{u}} \boldsymbol{u}_h, \boldsymbol{u}_h \rangle \leq \langle (A_{\boldsymbol{u}} + B^T B)^{-1} \boldsymbol{u}_h, \boldsymbol{u}_h \rangle \leq c_{2,\boldsymbol{u}} \langle M_{\boldsymbol{u}} \boldsymbol{u}_h, \boldsymbol{u}_h \rangle, \tag{3.24a}$$

$$c_{1,p} \langle M_p q_h, q_h \rangle \leq \langle A_p^{-1} q_h, q_h \rangle \leq c_{2,p} \langle M_p q_h, q_h \rangle, \tag{3.24b}$$

for some positive constants $c_{1,\boldsymbol{u}}$, $c_{1,p}$, $c_{2,\boldsymbol{u}}$, and $c_{2,p}$. The applications of the diagonal blocks $M_{\boldsymbol{u}}$ and M_p are often implemented as another level of iterative methods within

the outer Krylov subspace method of choice. Notice that $\mathcal{M}_D$ is also a norm-equivalent preconditioner. In fact, we get that the condition number $\kappa(\mathcal{M}_D\mathcal{A})$ satisfies (3.14) with $c_1 = \min\{c_{1,\boldsymbol{u}}, c_{1,p}\}$ and $c_2 = \max\{c_{2,\boldsymbol{u}}, c_{2,p}\}$. As before, we require the constants to be independent of discretization and physical parameters to obtain a robust preconditioner. One way to design $\mathcal{M}_D$ as a spectrally equivalent approximation of (3.22) is to use auxiliary space theory [35, 39], which is the topic of Paper B and further discussed in Section 3.2.4.

Remark 3.2.1 *Since our discrete system* (2.16) *is a parameter-dependent problem, depending on the physical parameters* $\boldsymbol{K}$ *and* $\boldsymbol{K}_{\boldsymbol{\nu}}$*, this reflects in the norms used to design a norm equivalent preconditioner* $\mathcal{B}_D$*. For that, a more suitable preconditioner is given with*

$$\mathcal{B}_D = \begin{pmatrix} A_{\boldsymbol{u}} + \varepsilon B^T B & 0 \\ 0 & \varepsilon^{-1} A_p \end{pmatrix}^{-1}, \tag{3.25}$$

where $\varepsilon > 0$ *is a constant depending on the eigenvalues of* $\boldsymbol{K}$ *and* $\boldsymbol{K}_{\boldsymbol{\nu}}$*. However, we continue this section with deriving a spectrally equivalent analogue to* (3.22) *and refer to Paper B for the discussion on the choice of* ε*.*

Once we have $\mathcal{M}_D$, we can easily derive the FoV-equivalent preconditioners. Taking (3.23) into account, the left preconditioner $\mathcal{M}_L$ equivalent to $\mathcal{B}_L$ and the right preconditioner $\mathcal{M}_U$ equivalent to $\mathcal{B}_U$ are given with

$$\mathcal{M}_L = \begin{pmatrix} M_{\boldsymbol{u}}^{-1} & 0 \\ -B & M_p^{-1} \end{pmatrix}^{-1} \quad \text{and} \quad \mathcal{M}_U = \begin{pmatrix} M_{\boldsymbol{u}}^{-1} & B^T \\ 0 & M_p^{-1} \end{pmatrix}^{-1}. \tag{3.26}$$

"Pressure-based" block preconditioner. On the other hand, we can reformulate our discrete problem to obtain a simpler, but still a efficient and robust preconditioner. Consider a block preconditioner

$$\mathcal{B}_D = \begin{pmatrix} A_{\boldsymbol{u}} & 0 \\ 0 & BA_{\boldsymbol{u}}^{-1}B^T \end{pmatrix}^{-1}. \tag{3.27}$$

Notice that the diagonal block $BA_{\boldsymbol{u}}^{-1}B^T$ is symmetric positive definite and can be seen as a mixed-dimensional equivalent to a Laplacian operator $\Delta = \operatorname{div} \operatorname{grad}$. This may be beneficial since the standard Krylov subspace methods work well with second-order elliptic operators. Though, we still need to show that the above preconditioner fits within the theory in Section 3.2.1.

The general block preconditioner is based on the choices of norms on the discrete spaces where we look for our solution. In the case of (3.27), it is assumed that choice of function space X_h allows for a primal formulation for pressure variable, that is, that we look for p_h in a discrete approximation of the space $H^1(\Omega) = \bigoplus_{d=0}^{n} H^1(\Omega^d)$. If that is true, $\mathcal{B}_D$ is a Riesz mapping corresponding to the norm

$$\begin{aligned} \|\boldsymbol{x}\|_{\boldsymbol{X}}^2 &= \|\boldsymbol{u}_h\|_{L^2(\Omega)}^2 + \|p_h\|_{H^1(\Omega)}^2 \\ &= \|\boldsymbol{u}_h\|_{L^2(\Omega)}^2 + \|p_h\|_{L^2(\Omega)}^2 + \|\boldsymbol{D}p_h\|_{L^2(\Omega)}^2. \end{aligned} \tag{3.28}$$

However, this is not obvious since Q_h, as a space of piecewise constants, is not a subspace of $H^1(\Omega)$. Still, it is possible to introduce a mixed-dimensional discrete gradient operator $\boldsymbol{D}_h$ such that $Q_h \in H(\boldsymbol{D}_h, \Omega)$ and the bound on the condition number (3.14) is preserved. Paper A provides a detailed analysis and implementation of this preconditioner, though we shortly discuss the idea in the following Section 3.2.3.

Lastly, it is also possible and preferable in this case to derive a spectrally equivalent preconditioner $\mathcal{M}_D$ and FoV-equivalent preconditioners $\mathcal{M}_L$ and $\mathcal{M}_U$. They take a similar form to (3.23) and (3.26), but the diagonal blocks $M_{\boldsymbol{u}}$ and M_p should now be equivalent to the diagonal blocks in (3.27). We refer the reader to Papers A and B for a more thorough presentation and provide a derivation of only the block diagonal preconditioners in the following sections.

3.2.3 Preconditioners Based on an Alternative Formulation

We first take a closer look at the "pressure-based" block preconditioners to solving (2.18). We begin with defining a bilinear form

$$a^{\boldsymbol{D}}(\boldsymbol{u}_h, \boldsymbol{v}_h) \coloneqq (\boldsymbol{K}^{-1}\boldsymbol{u}_h, \boldsymbol{v}_h)_{\boldsymbol{D},\Omega} + (\boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\boldsymbol{u}_h \cdot \boldsymbol{\nu}, \boldsymbol{v}_h \cdot \boldsymbol{\nu})_\Gamma, \tag{3.29}$$

where

$$(K^{-1}\boldsymbol{u}_h, \boldsymbol{v}_h)_{\boldsymbol{D},\Omega} \coloneqq \sum_{d=0}^{n} \left\{ \sum_{T^d \in \mathcal{T}_\Omega^d} \left[\sum_{\mathrm{f}^d \in \partial T^d} (\boldsymbol{u}_h \cdot \boldsymbol{\nu}_{\mathrm{f}^d})(\boldsymbol{v}_h \cdot \boldsymbol{\nu}_{\mathrm{f}^d})(\boldsymbol{K}^{-1}\boldsymbol{\phi}_{\mathrm{f}^d}, \boldsymbol{\phi}_{\mathrm{f}^d})_{T^d} \right] \right\}. \tag{3.30}$$

With $T^d \in \mathcal{T}_\Omega^d$ we denote a d-dimensional simplicial element of the grid, with $\mathrm{f}^d \in \partial T$ a face of the element T^d, with $\boldsymbol{\nu}_{f^d}$ the outer unit normal of face f^d, and with $\boldsymbol{\phi}_{\mathrm{f}^d} \in \mathbb{RT}_0(T^d)$ the basis function on face f^d.

Then, we can introduce the discrete gradient operator $\boldsymbol{D}_h : Q_h \to \boldsymbol{V}_h$ such that for any $\boldsymbol{v}_h \in \boldsymbol{V}_h$,

$$a^{\boldsymbol{D}}(\boldsymbol{D}_h p_h, \boldsymbol{v}_h) \coloneqq b(\boldsymbol{v}_h, p_h) = -(p_h, \boldsymbol{D} \cdot \boldsymbol{v}_h)_\Omega . \tag{3.31}$$

Using the discrete gradient operator, an *alternative formulation* to (2.16) is given as follows: Find $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times Q_h$ such that

$$a(\boldsymbol{u}_h, \boldsymbol{v}_h) + a^{\boldsymbol{D}}(\boldsymbol{D}_h p_h, \boldsymbol{v}_h) = -(g, \boldsymbol{v}_h \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{3.32a}$$

$$a^{\boldsymbol{D}}(\boldsymbol{D}_h q_h, \boldsymbol{u}_h) = -(f, q_h)_\Omega, \qquad \forall\, q_h \in Q_h. \tag{3.32b}$$

Notice that the above alternative formulation is by definition equivalent to the original formulation (2.16) and, therefore, well-posed with respect to the norm (3.20). Consequently, the norm equivalent preconditioner in this case is still the Riesz mapping (3.22). However, we want to derive a preconditioner similar to (3.27) that should work well with the Krylov subspace methods. For that purpose, we consider an new energy norm different to (3.20) and show the well-posedness with the new norm.

Consider an alternative energy norm

$$|||\boldsymbol{x}|||^2 = |||(\boldsymbol{u}_h, p_h)|||^2 = \|\boldsymbol{u}_h\|_a^2 + \|\boldsymbol{D}_h p_h\|_{a^{\boldsymbol{D}}}^2, \qquad \boldsymbol{x} \in \boldsymbol{X}, \tag{3.33}$$

where $\|\boldsymbol{u}_h\|_a^2 = a(\boldsymbol{u}_h, \boldsymbol{u}_h)$ and $\|\boldsymbol{u}_h\|_{a^{\boldsymbol{D}}}^2 = a^{\boldsymbol{D}}(\boldsymbol{u}_h, \boldsymbol{u}_h)$. The reason why this norm is of interest is that it can be shown that the forms $a(\cdot,\cdot)$ and $a^{\boldsymbol{D}}(\cdot,\cdot)$ are spectrally equivalent. That is,

$$c_{1,\boldsymbol{D}}\|\boldsymbol{v}_h\|_{a^{\boldsymbol{D}}} \leq \|\boldsymbol{v}_h\|_a \leq c_{2,\boldsymbol{D}}\|\boldsymbol{v}_h\|_{a^{\boldsymbol{D}}}, \qquad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{3.34}$$

holds for constants $c_{1,\boldsymbol{D}}, c_{2,\boldsymbol{D}} > 0$, depending only on the shape regularity of the mesh $\mathcal{T}_\Omega$. Based on this, in Paper A we show that another version of the inf-sup condition (2.15) regarding the discrete gradient $\boldsymbol{D}_h$ holds. It states that there exists a constant $\beta_{\boldsymbol{D}} > 0$ independent of the discretization and physical parameters, but depending on the shape regularity of the mesh, such that

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{v}_h \in \boldsymbol{V}_h} \frac{a^{\boldsymbol{D}}(\boldsymbol{D}_h q_h, \boldsymbol{v}_h)}{\|\boldsymbol{v}_h\|_a \|\boldsymbol{D}_h q_h\|_{a^{\boldsymbol{D}}}} \geq \beta_{\boldsymbol{D}}. \tag{3.35}$$

In turn, the bilinear form

$$\mathcal{L}(\boldsymbol{u}_h, p_h; \boldsymbol{v}_h, q_h) := a(\boldsymbol{u}_h, \boldsymbol{v}_h) + a^{\boldsymbol{D}}(\boldsymbol{D}_h p_h, \boldsymbol{v}_h) + a^{\boldsymbol{D}}(\boldsymbol{D}_h q_h, \boldsymbol{u}_h), \tag{3.36}$$

corresponding to the alternative formulation (3.32) satisfies the well-posedness conditions (3.10), induced by (3.35), in the new energy norm $|||\cdot|||$.

Finally, we derive the preconditioner. Let $D_{A_{\boldsymbol{u}}} : \boldsymbol{V}_h \to \boldsymbol{V}_h'$ be defined as

$$\langle D_{A_{\boldsymbol{u}}} \boldsymbol{u}_h, \boldsymbol{v}_h \rangle = a^{\boldsymbol{D}}(\boldsymbol{u}_h, \boldsymbol{v}_h), \qquad \boldsymbol{u}_h, \boldsymbol{v}_h \in \boldsymbol{V}_h. \tag{3.37}$$

In fact, by definitions of $a(\cdot,\cdot)$ and $a^{\boldsymbol{D}}(\cdot,\cdot)$, it follows that the matrix representation of linear operator $D_{A_{\boldsymbol{u}}}$ is exactly the diagonal of the matrix representation of linear operator $A_{\boldsymbol{u}}$. Therefore, by (3.31) we have for any $q_h \in Q_h$

$$\|\boldsymbol{D}_h q_h\|_{a^{\boldsymbol{D}}}^2 = \langle D_{A_{\boldsymbol{u}}} \boldsymbol{D}_h q_h, \boldsymbol{D}_h q_h \rangle = \langle B^T q_h, D_{A_{\boldsymbol{u}}}^{-1} B^T q_h \rangle = \langle B D_{A_{\boldsymbol{u}}}^{-1} B^T q_h, q_h \rangle, \tag{3.38}$$

and the norm (3.33) yields the Riesz mapping

$$\mathcal{B}_D = \begin{pmatrix} A_{\boldsymbol{u}} & 0 \\ 0 & B D_{A_{\boldsymbol{u}}}^{-1} B^T \end{pmatrix}^{-1}. \tag{3.39}$$

As discussed in Section 3.2.1, $\mathcal{B}_D$ is a norm equivalent preconditioner for solving the system (2.18) and the condition number $\kappa(\mathcal{B}_D \mathcal{A})$ is uniformly bounded, although the bound requires a shape regular mesh on Ω. Still, the numerical results in Paper A show that the requirement on the mesh regularity may be strict, since the preconditioner (3.39) performs well even when the mesh contains a small percentage of bad elements.

3.2.4 Auxiliary Space Preconditioners

Let us now go back to $\mathcal{M}_D$ in (3.23), the spectrally equivalent approximation of the Riesz mapping (3.22) corresponding to the norm $\|\cdot\|_{\boldsymbol{X}}$ in (3.20). Since the inverse of the operator A_p is trivial to implement, we can take $M_p = A_p^{-1}$. However, as discussed before, finding a good approximation of the operator $A_{\boldsymbol{u}} + B^T B$ requires looking into what it means to solve

a problem on the function space $H(\mathbf{D}\,\cdot, \Omega)$, the mixed-dimensional analogue to the $H(\mathrm{div})$ space.

The classical solution techniques are designed to work well with problems with a second-order differential operator that has strong ellipticity properties, such as the Laplacian operator Δ on the standard $H(\mathrm{grad})$ space. In contrast to that, in the $H(\mathrm{div})$ space, the main difficulty in developing solvers and preconditioners is that the kernel of the div operator is a much larger space containing all rotations. Hence, it requires a development of special solving methods that treat this large kernel separately. One of the ways it can be resolved is by observing that functions in $H(\mathrm{div})$ allow for a stable decomposition into components of higher regularity and a vector potential, hence called *regular decomposition*. By stable we mean that the decomposition is bounded in terms of norms. Based on that, suitable preconditioners would contain components spectrally equivalent to a second order differential operator where then the standard solution techniques can be applied. Still, using this approach in a discrete setting is not immediate and this is where the *auxiliary space methods* [66] turn out to be especially useful. To shorten the presentation, we refer the reader to [7, 35, 39, 62] for more details on the analysis and implementation of these methods and restrict to the application in the mixed-dimensional case.

Auxiliary Space Methods. We first present the general framework of the auxiliary space preconditioning. Assume V is a separable Hilbert space with an inner product $a(\cdot,\cdot)$ that induces a norm $\|\cdot\|_A$ and a symmetric positive definite operator $A : V \mapsto V'$ as $\langle Au, v\rangle = a(u, v)$. We aim to find $u \in V$ that for a given right hand side $f \in V'$ solves

$$Au = f. \tag{3.40}$$

Consider another inner product $s(\cdot,\cdot)$ on V different from $a(\cdot,\cdot)$. For $l = 1, 2, \ldots, L$, let W_l be additional Hilbert spaces with an inner product $a_l(\cdot,\cdot)$ that induces the linear operator $A_l : W_l \mapsto W_j'$. Also, assume that there exist transfer operators $\Pi_l : W_l \mapsto V$. Define the auxiliary product space $\bar{V} = V \times W_1 \times W_2 \times \cdots \times W_L$. Then, we can represent the inner product on $\bar{V}$ as

$$\bar{a}(\bar{v}, \bar{v}) = s(v, v) + \sum_{l=1}^{L} a_l(w_l, w_l) \qquad \bar{v} = (v, w_1, \ldots, w_L) \in \bar{V}. \tag{3.41}$$

Using $\bar{V}$, the auxiliary space preconditioner $B : V' \mapsto V$ for the linear problem (3.40) takes the following form

$$B = S^{-1} + \sum_{l=1}^{L} \Pi_l A_l^{-1} \Pi_l^*, \tag{3.42}$$

where Π_l^* is the adjoint of Π_l and the $S : V \mapsto V'$ is called the *smoother* operator and is induced by the inner product $s(\cdot,\cdot)$. If the following conditions are satisfied

(i) There exist $c_l > 0$, $l = 1, \ldots, L$ such that $\|\Pi_l w_l\|_A \leq c_l \|w_l\|_{A_l}, \quad \forall w_l \in W_l$;

(ii) There exist $c_s > 0$ such that $\|v\|_A \leq c_s \|v\|_S, \quad \forall v \in V$;

(iii) For every $v \in V$, there exists a decomposition $v = v_0 + \sum_{l=1}^{L} w_l$, $v_0 \in V$, $w_l \in W_l$ and $c_0 > 0$ such that

$$\|v_0\|_S^2 + \sum_{l=1}^{L} \|w_l\|_{A_l}^2 \leq c_0 \|v\|_A^2;$$

then we get a uniform bound on the condition number $\kappa(BA) \leq c_0^2(c_s^2 + c_1^2 + \cdots + c_L^2)$ [66]. Note that we can replace each A_l^{-1} by spectrally equivalent preconditioners B_l by replacing the inner product $a_l(\cdot,\cdot)$ on W_l with an spectrally equivalent inner product $b_l(\cdot,\cdot)$.

We apply this method in our mixed-dimensional norm equivalent preconditioner (3.23). More specifically, we use it as the operator $M_{\boldsymbol{u}}$, which is the spectrally equivalent approximation of the first block in the Riesz mapping (3.22) corresponding to the flux variable $\boldsymbol{u}_h$. The main ingredient of the implementation of the auxiliary space method in our case is the regular decomposition of functions in the finite element approximation of the space $H(\mathbf{D}\cdot,\Omega)$ defined in (2.10).

Regular Decomposition. Similar to the fixed-dimensional case, the regular decomposition of functions in the mixed-dimensional space $H(\mathbf{D}\cdot,\Omega)$ is based on the properties of a cochain complex known as de Rham complex [15, 59]. In particular, in the three-dimensional setting, the complex can be represented as

$$H(\mathbf{D},\Omega) \xrightarrow{\mathbf{D}} H(\mathbf{D}\times,\Omega) \xrightarrow{\mathbf{D}\times} H(\mathbf{D}\cdot,\Omega) \xrightarrow{\mathbf{D}\cdot} L^2(\Omega). \tag{3.43}$$

In particular, this means that $\mathbf{D}\cdot(\mathbf{D}\times\boldsymbol{f}) = 0$ for any $\boldsymbol{f} \in H(\mathbf{D}\times,\Omega)$, and for each function $\boldsymbol{u} \in H(\mathbf{D}\cdot,\Omega)$ with $\mathbf{D}\cdot\boldsymbol{u} = 0$ there exist a function $\boldsymbol{f}_{\boldsymbol{u}} \in H(\mathbf{D}\times,\Omega)$ such that $\mathbf{D}\times\boldsymbol{f}_{\boldsymbol{u}} = \boldsymbol{u}$. Notice that the last statement implies that any function $\boldsymbol{u} \in H(\mathbf{D}\cdot,\Omega)$ can be decomposed into two orthogonal parts: $\boldsymbol{a} \in H(\mathbf{D}\cdot,\Omega)$ such that $\mathbf{D}\cdot\boldsymbol{a} = \mathbf{D}\cdot\boldsymbol{u}$, and a $(\mathbf{D}\cdot)$-free part $\mathbf{D}\times\boldsymbol{c}$ for some $\boldsymbol{c} \in H(\mathbf{D}\times,\Omega)$.

Although not necessarily orthogonal, the general idea of the regular decomposition is to ensure a similar partition, but with the components that have a slightly higher regularity. The higher regularity here, in terms of $H(\mathrm{grad})$ functions, helps with implementing auxiliary space methods in the preconditioner to $H(\mathrm{div})$ problems. The reason is that the application of the operators A_l on the auxiliary product space will closely resemble the application of the Laplacian or the vector-Laplacian operator. We refer to [35] and the preliminaries of Paper B to see how this works in the fixed-dimensional setting, while we concentrate on the mixed-dimensional case. We remind that our goal is to combine this new mixed-dimensional regular decomposition and the auxiliary space method into $M_{\boldsymbol{u}}$, that should give an efficient way to approximate the first (flux) block in the Riesz mapping preconditioner (3.22).

Let $H^1(\mathbf{D}\cdot,\Omega) \subset H(\mathbf{D}\cdot,\Omega)$ and $H^1(\mathbf{D}\times,\Omega) \subset H(\mathbf{D}\times,\Omega)$ be the subspaces of functions with H^1 regularity locally on each subdomain Ω_i^d. Then, in the continuous setting, we have shown that for any function $\boldsymbol{u} \in H(\mathbf{D}\cdot,\Omega)$, there exist $\boldsymbol{a} \in H^1(\mathbf{D}\cdot,\Omega)$ and $\boldsymbol{c} \in H^1(\mathbf{D}\times,\Omega)$ such that

$$\boldsymbol{u} = \boldsymbol{a} + \mathbf{D}\times\boldsymbol{c}, \qquad \text{and} \qquad \|\boldsymbol{a}\|_{H^1(\mathbf{D}\cdot,\Omega)} + \|\boldsymbol{c}\|_{H^1(\mathbf{D}\times,\Omega)} \leq C\|\boldsymbol{u}\|_{H(\mathbf{D}\cdot,\Omega)}, \tag{3.44}$$

for some constant $C > 0$.

In the discrete setting, we look at the mixed-dimensional finite element exterior calculus, namely the discrete mixed-dimensional de Rham complex [15]. Following that structure, we have that the finite element approximation $H_h(\mathbf{D}\cdot, \Omega) \subset H(\mathbf{D}\cdot, \Omega)$ consists of lowest-order face-based Nedelec elements for dimension $d = 3$, lowest-order Raviart-Thomas elements for $d = 2$ and linear Lagrange elements for $d = 1$. To approximate the subspaces of increased regularity, we define another finite element approximations $H_h^1(\mathbf{D}\cdot, \Omega) \subset H^1(\mathbf{D}\cdot, \Omega)$ and $H_h^1(\mathbf{D}\times, \Omega) \subset H^1(\mathbf{D}\times, \Omega)$, consisting of tuples of nodal linear Lagrange elements on each subdomain Ω_i^d. Then, the discrete analogue of the decomposition (3.44) states that for any $\boldsymbol{u}_h \in H_h(\mathbf{D}\cdot, \Omega)$, there exist functions $\boldsymbol{a}_h \in H_h^1(\mathbf{D}\cdot, \Omega)$ and $\boldsymbol{c}_h \in H_h^1(\mathbf{D}\times, \Omega)$ and *high-frequency* functions $\boldsymbol{b}_h \in H_h(\mathbf{D}\cdot, \Omega)$ and $\boldsymbol{d}_h \in H_h(\mathbf{D}\times, \Omega)$ such that

$$\boldsymbol{u}_h = \Pi_h^{\mathbf{D}\cdot} \boldsymbol{a}_h + \boldsymbol{b}_h + \mathbf{D}\times(\Pi_h^{\mathbf{D}\times} \boldsymbol{c}_h + \boldsymbol{d}_h), \qquad \text{and} \tag{3.45a}$$

$$\begin{aligned} \|\boldsymbol{a}_h\|_{H^1(\mathbf{D}\cdot,\Omega)} + \|h^{-1}\boldsymbol{b}_h\|_{L^2(\Omega)} + \|\boldsymbol{c}_h\|_{H^1(\mathbf{D}\times,\Omega)} + \|h^{-1}\boldsymbol{d}_h\|_{L^2(\Omega)} \\ \leq C_h \|\boldsymbol{u}_h\|_{H(\mathbf{D}\cdot,\Omega)}, \end{aligned} \tag{3.45b}$$

for some constant $C_h > 0$ independent of the discretization parameter h. Here, we have denoted with $\Pi_h^{\mathbf{D}\cdot} : H(\mathbf{D}\cdot, \Omega) \to H_h(\mathbf{D}\cdot, \Omega)$ and $\Pi_h^{\mathbf{D}\times} : H(\mathbf{D}\times, \Omega) \to H_h(\mathbf{D}\times, \Omega)$ the stable projection operators from the continuous to the corresponding discrete spaces.

We observe that the above regular decomposition satisfies the conditions (i)–(iii) of the general framework of the auxiliary space methods. More specifically, take $V = H_h(\mathbf{D}\cdot, \Omega)$, $W_1 = H_h^1(\mathbf{D}\cdot, \Omega)$, $W_2 = H_h(\mathbf{D}\times, \Omega)$, $W_3 = H_h^1(\mathbf{D}\cdot, \Omega)$, and take the transfer operators $\Pi_1 = \Pi_h^{\mathbf{D}\cdot}$ restricted to $H_h^1(\mathbf{D}\cdot, \Omega)$, $\Pi_2 = \mathbf{D}\times$ and $\Pi_3 = \mathbf{D}\times\Pi_h^{\mathbf{D}\times}$ restricted to $H_h^1(\mathbf{D}\times, \Omega)$. Let $A_{\mathbf{D}\cdot}$ and $A_{\mathbf{D}\times}$ represent the inner products on the spaces $H_h^1(\mathbf{D}\cdot, \Omega)$ and $H^1(\mathbf{D}\times, \Omega)$, respectively, and choose smoother operators $S_{\mathbf{D}\cdot}$ and $S_{\mathbf{D}\times}$. Then, our preconditioner $M_{\boldsymbol{u}}$ as part of the Riesz mapping (3.23) is given with

$$M_{\boldsymbol{u}} = \Pi_h^{\mathbf{D}\cdot} A_{\mathbf{D}\cdot}^{-1} (\Pi_h^{\mathbf{D}\cdot})^* + S_{\mathbf{D}\cdot}^{-1} + ((\mathbf{D}\times)\Pi_h^{\mathbf{D}\times}) A_{\mathbf{D}\times}^{-1} ((\mathbf{D}\times)\Pi_h^{\mathbf{D}\times})^* + (\mathbf{D}\times) S_{\mathbf{D}\times}^{-1} (\mathbf{D}\times)^*. \tag{3.46}$$

The conditions (i)–(iii) directly imply that $M_{\boldsymbol{u}}$ is spectrally equivalent to the upper block $A_{\boldsymbol{u}} + B^T B$ in the Riesz mapping (3.22). Therefore, $\mathcal{M}_D$ in (3.23) is a norm equivalent preconditioner for the system (2.18). Moreover, the bound in (3.45) ensures that the condition number $\kappa(\mathcal{M}_D \mathcal{A})$ is uniform and $\mathcal{M}_D$ is a robust preconditioner.

Implementing (3.46) often requires to replace the inverses with spectrally equivalent operators. Usually, we substitute $A_{\mathbf{D}\cdot}^{-1}$ and $A_{\mathbf{D}\times}^{-1}$ each with the AMG method, since both matrices act as second-order elliptic operators and AMG methods perform well in those cases, as discussed in Section 3.1.2. However, we note that the performance of AMG in the mixed-dimensional setting has not yet been investigated fully. We reserve this as a topic of future research, as outlined in Section 4.2.

Common implementation of smoothers $S_{\mathbf{D}\cdot}$ and $S_{\mathbf{D}\times}$ are either one Jacobi or Gauss-Seidel iteration. Paper B presents in detail the analysis and implementation of the mixed-dimensional auxiliary space preconditioners in a more general setting.

Remark 3.2.2 *In Paper B, we develop the mixed-dimensional regular decomposition and the auxiliary space preconditioners for any systems of mixed-dimensional elliptic partial differential equations. In particular, these preconditioners can also be applied to $H(\mathbf{D}\times,\Omega)$ problems or even coupled mixed-dimensional problems. The implementation is, however, a possible prospect in future research, see Section 4.2.*

3.3 Domain Decomposition Method

In contrast to the previous sections, here we aim to exploit more the distinctive geometrical structure of fractured porous media in the numerical solvers. The approach we present is straightforwardly imposed by the fracture network as decomposing the rock domain into separate parts that exchange information over the network. Therefore, the domain decomposition methods are shown to be the perfect candidate. In the following, we introduce the idea of general domain decomposition methods, demonstrate how they work on an example of porous media flow without fractures, and finally apply the framework to our problem with fracture flow.

3.3.1 General Approach

Domain decomposition methods have been developed for solving systems of equations that are too large to be solved with traditional techniques. The principal idea is to benefit from the multiprocessor architecture of modern computers by decomposing a large problem into several smaller subproblems that can be solved in parallel. These subproblems are then coupled through the unknowns at the shared interfaces. Moreover, in this way, each subproblem can represent different kinds of processes in the system, making domain decomposition methods applicable to a wide range of physical phenomena, such as our flow model.

In the simplest setting of a domain decomposition method, we assume that a bounded open domain Ω can be partitioned into two subdomains Ω_1 and Ω_2 that share a common interface $\Sigma = \bar{\Omega}_1 \cap \bar{\Omega}_2$, see Figure 3.1. We aim to reformulate a general problem

$$Au = f \qquad \text{in } \Omega, \tag{3.47}$$

for u the unknown solution, f a given right hand side and A represents a differential operator, by splitting it into two problems on the two subdomains. Let u_i and f_i be the restriction of the solution u and right hand side function f on each Ω_i, $i = 1, 2$, respectively. Then (3.47) is equivalent to the multi-domain problem

$$Au_1 = f_1 \qquad \text{in } \Omega_1, \tag{3.48a}$$
$$Au_2 = f_2 \qquad \text{in } \Omega_2, \tag{3.48b}$$

with interface conditions

$$\Phi(u_1) = \Phi(u_2) \qquad \text{on } \Sigma, \tag{3.48c}$$
$$\Psi(u_1) = \Psi(u_2) \qquad \text{on } \Sigma, \tag{3.48d}$$

where the functions Φ and Ψ depend on the nature of the problem in hand. The conditions (3.48c)–(3.48d) are called the transmission conditions and they usually express a certain continuity of the solution over the interface between the subdomains. With that, these domain decomposition methods solve the subdomain problems explicitly or iteratively by coupling the subdomain solutions in a special interface problem.

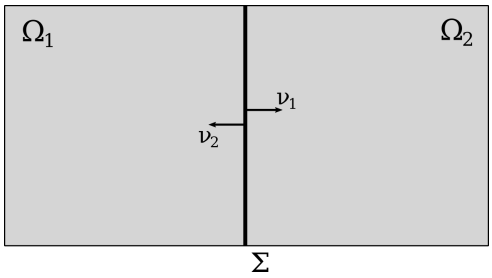


Figure 3.1: A simple domain decomposition of a bounded set Ω.

For easier implementation of the method to our problem, let us first demonstrate how the concept of domain decomposition works on a standard flow in porous media model without fractures. This method was introduced in [64]. The equations (3.48a)–(3.48b) are replaced by

$$\boldsymbol{u}_i = -\boldsymbol{K}\nabla p_i \qquad \text{in } \Omega_i, \tag{3.49a}$$

$$\nabla \cdot \boldsymbol{u}_i = f_i \qquad \text{in } \Omega_i, \tag{3.49b}$$

for $i = 1, 2$, and the transmission conditions are given as the continuity of pressure and normal flux over Σ,

$$p_1 = p_2 \qquad \text{on } \Sigma, \tag{3.49c}$$

$$\boldsymbol{u}_1 \cdot \boldsymbol{\nu}_1 = -\boldsymbol{u}_2 \cdot \boldsymbol{\nu}_2 \qquad \text{on } \Sigma. \tag{3.49d}$$

For simplicity, assume that $p_i = 0$ and $\boldsymbol{u}_i \cdot \boldsymbol{\nu} = 0$ on rest of the boundary of Ω_i.

Let λ indicate the unknown pressure on the interface Σ in (3.49c). Also, take $\boldsymbol{V}_i = H(\mathrm{div}, \Omega_i)$, $Q_i = L^2(\Omega_i)$ and $\Lambda = H^{\frac{1}{2}}(\Sigma)$. Similar as in Section 2.2.1, the variational formulation of the above problem states to find $(\boldsymbol{u}_i, p_i) \in \boldsymbol{V}_i \times Q_i$ and the interface pressure $\lambda \in \Lambda$ such that for $i = 1, 2$,

$$(\boldsymbol{K}^{-1}\boldsymbol{u}_i, \boldsymbol{v}_i)_{\Omega_i} - (p_i, \nabla \cdot \boldsymbol{v}_i)_{\Omega_i} + (\lambda, \boldsymbol{u}_i \cdot \boldsymbol{\nu}_i)_{\Sigma} = 0 \qquad \forall \boldsymbol{v}_i \in \boldsymbol{V}_i, \tag{3.50a}$$

$$(\nabla \cdot \boldsymbol{u}_i, q_i)_{\Omega_i} = (f_i, q_i)_{\Omega_i} \qquad \forall q_i \in Q_i. \tag{3.50b}$$

$$\sum_{i=1}^{2} \langle \boldsymbol{u}_i \cdot \boldsymbol{\nu}_i, \mu \rangle = 0 \qquad \forall \mu \in \Lambda. \tag{3.50c}$$

The equation (3.50c) balances the contributions from the subdomains by imposing the weak continuity of the normal flux. However, we still cannot solve separately the subdomain problems since they are coupled over (3.50c). Therefore, we reformulate the system (3.50) by

reducing it to an interface problem. Define a linear operator $S : \Lambda \to \Lambda'$ such that for given $\lambda \in \Lambda$

$$\langle S\lambda, \mu\rangle = -\sum_{i=1}^{2} \langle \boldsymbol{u}_i^* \cdot \boldsymbol{\nu}_i, \mu\rangle \qquad \mu \in \Lambda, \tag{3.51}$$

where $(\boldsymbol{u}_i^*, p_i^*) \in \boldsymbol{V}_i \times Q_i$ solve

$$(\boldsymbol{K}^{-1}\boldsymbol{u}_i^*, \boldsymbol{v}_i)_{\Omega_i} - (p_i^*, \nabla \cdot \boldsymbol{v}_i)_{\Omega_i} + (\lambda, \boldsymbol{u}_i^* \cdot \boldsymbol{\nu}_i)_{\Sigma} = 0 \qquad \forall \boldsymbol{v}_i \in \boldsymbol{V}_i, \tag{3.52a}$$

$$(\nabla \cdot \boldsymbol{u}_i^*, q_i)_{\Omega_i} = 0 \qquad \forall q_i \in Q_i. \tag{3.52b}$$

Also, let $(\boldsymbol{u}_i^0, p_i^0) \in \boldsymbol{V}_i \times Q_i$ solve

$$(\boldsymbol{K}^{-1}\boldsymbol{u}_i^0, \boldsymbol{v}_i)_{\Omega_i} - (p_i^0, \nabla \cdot \boldsymbol{v}_i)_{\Omega_i} = 0 \qquad \forall \boldsymbol{v}_i \in \boldsymbol{V}_i, \tag{3.53a}$$

$$(\nabla \cdot \boldsymbol{u}_i^0, q_i)_{\Omega_i} = (f_i, q_i)_{\Omega_i} \qquad \forall q_i \in Q_i. \tag{3.53b}$$

Take $\chi \in \Lambda'$ such that $\langle \chi, \mu\rangle = \sum_{i=1}^{2} \langle \boldsymbol{u}_i^0 \cdot \boldsymbol{\nu}_i, \mu\rangle$. Then (3.50) is equivalent to finding $\lambda \in \Lambda$ in

$$\langle S\lambda, \mu\rangle = \langle \chi, \mu\rangle \qquad \forall \mu \in \Lambda, \tag{3.54}$$

and the subdomain solutions are given with $\boldsymbol{u}_i = \boldsymbol{u}_i^0 + \boldsymbol{u}_i^*$ and $p_i = p_i^0 + p_i^*$.
The operator S is called the *Steklov-Poincaré* operator. An evaluation of S includes:

(i) Use the given interface pressure λ as a pressure boundary condition on Σ for the subdomain problems;

(ii) Solve the two subdomain problems to get $(\boldsymbol{u}_i^*, p_i^*)$ for $i = 1, 2$;

(iii) Map back the normal trace of the flux $-\boldsymbol{u}_i^* \cdot \boldsymbol{\nu}_i$ to the interface.

The equation (3.54) makes sure that the normal fluxes from (3.52) and (3.53) are balanced in a weak sense. A similar procedure can be retrieved in the discrete setting, giving a system of equations that can be reduced to a smaller system for the interface degrees of freedom.

The evaluation of the discrete Steklov-Poincaré operator can be achieved with a direct or an iterative approach. The direct approach includes explicitly computing the application of the operator S to each degree of freedom on the interface but, in the case when that number of degrees of freedom is large, we rely on iterative methods to solve (3.54). See [6, 51, 53] for more examples of domain decomposition methods in applications to flow in porous media.

3.3.2 Application to the Fracture Flow Model

In the case where the interface Σ contains a fracture, we have to also account for the flow contributions tangentially along the interface in (3.54). In fact, it is exactly what happens in (2.2b) for $d = n - 1$. Following the guidelines in the previous section, we derive the reduced interface problem for our mixed-dimensional model (2.2). For simplicity, we restrict the setting to a two-dimensional problem with one fracture dividing the rock matrix domain into two parts, as in Figure 3.2. Also, we jump straight to the discrete formulation of our

problem that can be related to the non-matching form in Section 2.2.3. Note that we slightly change the notation and remove the dimensional superscripts, but rather denote the fracture subdomain with an subscript f. This will be clearer in Section 3.4 where we are required to introduce extra indices in the new iterative procedures.

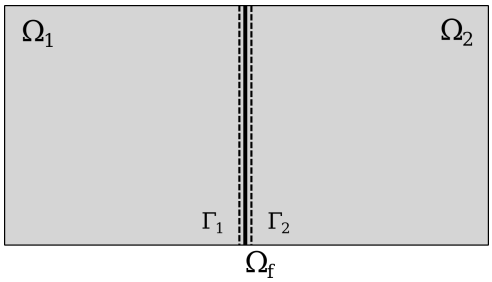


Figure 3.2: A simple domain decomposition of a bounded set Ω containing a fracture Ω_f.

Let Ω_1, Ω_2 be the two rock matrix subdomains that share an interface with a fracture Ω_f, see the illustration in Figure 3.2. If we write down each equation in (2.16), the mixed-dimensional discrete fracture flow problem states to find $(\boldsymbol{u}_{h,i}, p_{h,i}) \in \boldsymbol{V}_{h,i} \times Q_{h,i}$ for $i = 1, 2$ and $(\boldsymbol{u}_{h,f}, p_{h,f}) \in \boldsymbol{V}_{h,f} \times Q_{h,f}$ such that

$$
\begin{aligned}
&(\boldsymbol{K}^{-1}\boldsymbol{u}_{h,i}, \boldsymbol{v}_{h,i})_{\Omega_i} + \boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\langle \boldsymbol{u}_{h,i}\cdot\boldsymbol{\nu}_i, \boldsymbol{v}_{h,i}\cdot\boldsymbol{\nu}_i\rangle_{\Gamma_i} \\
&\qquad -(p_{h,i}, \nabla\cdot\boldsymbol{v}_{h,i})_{\Omega_i} - \langle p_{h,f}, \boldsymbol{v}_{h,i}\cdot\boldsymbol{\nu}_i\rangle_{\Gamma_i} = 0 \qquad \forall \boldsymbol{v}_{h,i} \in \boldsymbol{V}_{h,i}, \qquad (3.55a)\\
&\qquad (\nabla\cdot\boldsymbol{u}_{h,i}, q_{h,i})_{\Omega_i} = (f, q_{h,i})_{\Omega_i} \qquad \forall q_{h,i} \in Q_{h,i}, \qquad (3.55b)\\
&(\boldsymbol{K}^{-1}\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f} - (p_{h,f}, \nabla\cdot\boldsymbol{v}_{h,f})_{\Omega_f} = 0 \qquad \forall \boldsymbol{v}_{h,f} \in \boldsymbol{V}_{h,f}, \qquad (3.55c)\\
&(\nabla\cdot\boldsymbol{u}_{h,f}, q_{h,f})_{\Omega_f} + (\llbracket \boldsymbol{u}_{h,i}\cdot\boldsymbol{\nu}_i \rrbracket, q_{h,f})_{\Omega_f} = (f, q_{h,f})_{\Omega_f} \qquad \forall q_{h,f} \in Q_{h,f}. \qquad (3.55d)
\end{aligned}
$$

Here, we have assumed homogeneous pressure boundary conditions in (2.5a) to simplify the formulation. The equations (3.55a)–(3.55b) represent the local problems on the subdomains Ω_1 and Ω_2, while (3.55c)–(3.55d) contain the fracture flow problems and the balancing of contributions from the subdomains. Similar as in (3.51), take $\lambda_h = p_{h,f}$ and $\Lambda_h = Q_{h,f}$, and define the Steklov-Poincaré operator $S : \Lambda_h \to \Lambda_h'$

$$
\langle S\lambda_h, \mu_h\rangle = -\sum_{i=1}^{2}\langle \boldsymbol{u}_{h,i}^*\cdot\boldsymbol{\nu}_i, \mu_h\rangle \qquad \mu_h \in \Lambda_h, \qquad (3.56)
$$

where $(\boldsymbol{u}_{h,i}^*, p_{h,i}^*) \in \boldsymbol{V}_{h,i} \times Q_{h,i}$ solve

$$
\begin{aligned}
&(\boldsymbol{K}^{-1}\boldsymbol{u}_{h,i}^*, \boldsymbol{v}_{h,i})_{\Omega_i} + \boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\langle \boldsymbol{u}_{h,i}^*\cdot\boldsymbol{\nu}_i, \boldsymbol{v}_{h,i}\cdot\boldsymbol{\nu}_i\rangle_{\Gamma_i} \\
&\qquad -(p_{h,i}^*, \nabla\cdot\boldsymbol{v}_{h,i})_{\Omega_i} - \langle \lambda_h, \boldsymbol{v}_{h,i}\cdot\boldsymbol{\nu}_i\rangle_{\Gamma_i} = 0 \qquad \forall \boldsymbol{v}_{h,i} \in \boldsymbol{V}_{h,i}, \qquad (3.57a)\\
&\qquad (\nabla\cdot\boldsymbol{u}_{h,i}^*, q_{h,i})_{\Omega_i} = 0 \qquad \forall q_{h,i} \in Q_{h,i}. \qquad (3.57b)
\end{aligned}
$$

Also, let $(\boldsymbol{u}^0_{h,i}, p^0_{h,i}) \in \boldsymbol{V}_{h,i} \times Q_{h,i}$ solve

$$
\begin{aligned}
(\boldsymbol{K}^{-1}\boldsymbol{u}^0_{h,i}, \boldsymbol{v}_{h,i})_{\Omega_i} + \boldsymbol{K}_{\boldsymbol{\nu}}^{-1}\langle \boldsymbol{u}^0_{h,i}\cdot\boldsymbol{\nu}_i, \boldsymbol{v}_{h,i}\cdot\boldsymbol{\nu}_i\rangle_{\Gamma_i} & \\
-(p^0_{h,i}, \nabla\cdot\boldsymbol{v}_{h,i})_{\Omega_i} &= 0 \qquad \forall \boldsymbol{v}_{h,i} \in \boldsymbol{V}_{h,i}, \qquad (3.58a)\\
(\nabla\cdot\boldsymbol{u}^0_{h,i}, q_{h,i})_{\Omega_i} &= (f, q_{h,i})_{\Omega_i} \qquad \forall q_{h,i} \in Q_{h,i}. \qquad (3.58b)
\end{aligned}
$$

Take $\chi_h \in \Lambda_h'$ such that $\langle \chi_h, \mu_h\rangle = \sum_{i=1}^{2} \langle \boldsymbol{u}^0_{h,i}\cdot\boldsymbol{\nu}_i, \mu_h\rangle$. Then (3.55) is equivalent to finding $(\boldsymbol{u}_{h,f}, \lambda_h) \in \boldsymbol{V}_{h,f} \times \Lambda_h$ such that

$$
\begin{aligned}
(\boldsymbol{K}^{-1}\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f} - (\lambda_h, \nabla\cdot\boldsymbol{v}_{h,f})_{\Omega_f} &= 0 \qquad \forall \boldsymbol{v}_{h,f} \in \boldsymbol{V}_{h,f}, \qquad (3.59a)\\
(\nabla\cdot\boldsymbol{u}_{h,f}, \mu_h)_{\Omega_f} + \langle S\lambda_h, \mu_h\rangle &= (f, \mu_h)_{\Omega_f} + \langle \chi_h, \mu_h\rangle \qquad \forall \mu_h \in \Lambda_h. \qquad (3.59b)
\end{aligned}
$$

and the subdomain solutions are given with $\boldsymbol{u}_{h,i} = \boldsymbol{u}^0_{h,i} + \boldsymbol{u}^*_{h,i}$ and $p_{h,i} = p^0_{h,i} + p^*_{h,i}$.

Notice that the application of the operator S replaces the jump term in equation (3.55d), which represents the influx contributions of the subdomains to the balancing of forces in the fracture domain. What differs the equations (3.59) from (3.54) is that now we need to solve the entire Darcy flow problem in the fracture. This can be done with a direct or an iterative method, though in each approach the dominating computational cost comes from the evaluation of the operator S. As described in Section 3.3.1, the step (ii) of the evaluation requires solving the subdomain problems. The following section describes in short one of the approaches to explicitly compute the evaluation of the Steklov-Poincaré operator, which will be useful in the context of the linearization methods in Section 3.4.

Remark 3.3.1 *In many cases, the interface problem* (3.59) *can contain a large number of unknowns and it can become too ill-conditioned to solve with iterative methods. To resolve that issue, we need to impose a preconditioner in* (3.59)*. One option is to use the block preconditioners developed in Section 3.2 in the case of mixed finite element discretizations, though other preconditioning approaches in the context of domain decomposition methods are possible [53].*

3.3.3 Multiscale Flux Basis

We remind that, to explicitly compute the application of the Steklov-Poincaré operator S, we need to evaluate it at each interface degree of freedom. This can get computationally expensive, especially if this needs to be repeated within an iterative method, such as the linearization methods in Section 3.4. One way to reduce the computational cost is to consider a coarser grid in the fracture domain and by employing the multiscale flux basis from [29]. In this way, we can pre-compute and store the subdomain responses associated with each fracture pressure degree of freedom before the iterative process. We apply this approach in Papers C and D.

Let $H \geq h$ be the typical mesh size of the fracture domain grid. Following [29], we define $(\phi_{H,f,\ell})_{\ell=1}^{N_f}$ to be the set of basis functions on the interface pressure space Λ_H, where

N_f is the number of pressure degrees of freedom on Ω_f. That is, we have for any $\lambda_H \in \Lambda_H$

$$\lambda_H = \sum_{\ell=1}^{N_f} \lambda_H^\ell \phi_{H,f,\ell}. \tag{3.60}$$

with some constant coefficients λ_H^ℓ for $1 \leq \ell \leq N_f$. Furthermore, introduce the projection operators $\Pi_{H,i} : \Lambda_H \to \boldsymbol{V}_{h,i} \cdot \boldsymbol{\nu}_i$ as the L^2-projections from the interface pressure space to the normal velocity trace on Γ_i. These projections are stable meaning that the unique solution of (3.59) still exists even with a coarser mesh on the interface [6, 16].

We can compute the multiscale flux basis functions $(\psi_{H,f,\ell})_{\ell=1}^{N_f}$ with the following algorithm:

Algorithm 3.3.1 *Enter a basis* $(\phi_{H,f,\ell})_{\ell=1}^{N_f}$ *and set* $\ell = 0$.
Do

1. *Increase* $\ell \to \ell + 1$.
2. *Project* $\phi_{H,f,\ell}$ *on the subdomain boundary* Γ_i*:* $\phi_{H,f,\ell} \xrightarrow{\Pi_{H,i}} \lambda_{h,i}$.
3. *Solve the subdomain problems* (3.57) *with a boundary condition* $\lambda_{h,i}$ *on* Γ_i.
4. *Project the boundary flux* $-\boldsymbol{u}_{h,i}^* \cdot \boldsymbol{\nu}_i$ *onto the dual of the fracture pressure space:* $-\boldsymbol{u}_{h,i}^* \cdot \boldsymbol{\nu}_i \xrightarrow{\Pi_{H,i}^T} \psi_{H,f,\ell}$.

While $\ell \leq N_f$.

Since the operator S is linear, we have that for λ_H in (3.60)

$$\langle S\lambda_H, \mu_H \rangle = \langle S \sum_{\ell=1}^{N_f} \lambda_H^\ell \phi_{H,f,\ell}, \mu_H \rangle = \sum_{\ell=1}^{N_f} \lambda_H^\ell \langle S\phi_{H,f,\ell}, \mu_H \rangle = \sum_{\ell=1}^{N_f} \lambda_H^\ell \langle \psi_{H,f,\ell}, \mu_H \rangle. \tag{3.61}$$

3.4 Linearization Methods

In this section, we finalize the presentation on numerical solvers with a choice of *linearization methods* to solve the non-linear fracture flow models introduced in Section 2.3. As described there, the non-linearity term (2.25) is imposed in the permeability coefficient in the fracture domain, while the rock matrix subdomains still preserve the standard Darcy's law. Therefore, the only change in the interface problem (3.59) is replacing the permeability $\boldsymbol{K}^{-1}$ with the term $\boldsymbol{k}(\boldsymbol{u}) = \boldsymbol{K}^{-1} + \xi(\boldsymbol{u})$ to get

$$(\boldsymbol{k}(\boldsymbol{u}_{h,f})\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f} - (\lambda_h, \nabla \cdot \boldsymbol{v}_{h,f})_{\Omega_f} = 0 \qquad \forall \boldsymbol{v}_{h,f} \in \boldsymbol{V}_{h,f}, \tag{3.62a}$$

$$(\nabla \cdot \boldsymbol{u}_{h,f}, \mu_h)_{\Omega_f} + \langle S\lambda_h, \mu_h \rangle = (f, \mu_h)_{\Omega_f} + \langle \chi_h, \mu_h \rangle \qquad \forall \mu_h \in \Lambda_h. \tag{3.62b}$$

The methods mentioned up to this point are designed to solve linear systems of equations, so we cannot straightforwardly apply them to the above problem. For that matter, we first need to linearize the non-linear terms in the system. In the following, we describe three approaches to linearization of the problem (3.62). Note that it is also possible to use these approaches in a time-dependent problem, such as the case of compressible flow in Section 2.3.2. That

is, in each time step, we solve the system (2.29), which basically expands (3.62) to a multi-fracture case and with an extra pressure term. Thus, to shorten the presentation, we only show how the linearization methods work on the problem (3.62). Also, we only give the iterative formulations of these methods applied to (3.62), while a thorough analysis of convergence and implementation can be found in Papers C and D.

The linearization methods considered in the following sections are the fixed-point method, Newton's method, and the L-scheme.

3.4.1 Fixed-point method

The simplest approach to linearize the term $(\boldsymbol{k}(\boldsymbol{u}_{h,f})\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f}$ is to use the *fixed-point method*. In general, the fixed-point method, as the name suggests, aims to find a fixed point of a given function. Let a continuous real valued function f have a fixed point x, meaning that $f(x) = x$. The fixed-point iterations produce a sequence $(x^{(k)})_{k\geq 0}$ of approximations of x from a given initial guess $x^{(0)}$ such that in each iteration

$$x^{(k)} = f(x^{(k-1)}), \qquad k = 1, 2, \dots \tag{3.63}$$

With certain conditions on the function f, it can be proven that this series linearly converges to the fixed point x. Moreover, if the function f is Lipschitz continuous, it converges to a unique fixed point.

To apply the fixed-point method to (3.62), give a initial guess $(\boldsymbol{u}_{h,f}^{(0)}, \lambda_h^{(0)})$. Then, in each iteration $k = 1, 2, \dots$ we solve

$$(\boldsymbol{k}(\boldsymbol{u}_{h,f}^{(k-1)})\boldsymbol{u}_{h,f}^{(k)}, \boldsymbol{v}_{h,f})_{\Omega_f} - (\lambda_h^{(k)}, \nabla \cdot \boldsymbol{v}_{h,f})_{\Omega_f} = 0 \qquad \forall \boldsymbol{v}_{h,f} \in \boldsymbol{V}_{h,f}, \tag{3.64a}$$

$$(\nabla \cdot \boldsymbol{u}_{h,f}^{(k)}, \mu_h)_{\Omega_f} + \langle S\lambda_h^{(k)}, \mu_h \rangle = (f, \mu_h)_{\Omega_f} + \langle \chi_h, \mu_h \rangle \qquad \forall \mu_h \in \Lambda_h, \tag{3.64b}$$

till convergence to a prescribed tolerance on the residual $r^{(k)} = (\boldsymbol{u}_{h,f}^{(k)} - \boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k)} - \lambda_h^{(k-1)})$. In each iteration k, the above linear system can now be solved directly or iteratively. For example, we can use the GMRES method since the system (3.64) has a saddle-point structure, making it indefinite. The evaluation of the operator S can be done with the multiscale flux basis described in Section 3.3.3. The initial guess $\boldsymbol{u}_{h,f}^{(0)}$ can be arbitrary, although it is preferable to start relatively close to the right solution. In Paper C we consider $\boldsymbol{u}_{h,f}^{(0)}$ to be the solution of the linear interface problem (3.59).

The result of this procedure is then used to generate the solution in the subdomains

$$\boldsymbol{u}_{h,i} = \boldsymbol{u}_{h,i}^{*,(\infty)} + \boldsymbol{u}_{h,i}^{0}, \tag{3.65a}$$

$$p_{h,i} = p_{h,i}^{*,(\infty)} + p_{h,i}^{0}, \tag{3.65b}$$

where $\boldsymbol{u}_{h,i}^{*,(\infty)}$ and $p_{h,i}^{*,(\infty)}$ indicate the solutions of (3.57) with the value of interface pressure $\lambda_h^{(\infty)}$ given at convergence of (3.64).

3.4.2 Newton's method

In the second approach, we use a special case of the fixed-point method called the *Newton's method*. Here, the basic idea is to provide a sequence of approximative solutions, with each new solution constructed as the root of the linear approximation of the non-linear function at the solution of the previous iteration. More specifically, let f be continuously differentiable function of real value x and let $\bar{x}$ be its unique root. With a given current approximation $x^{(k-1)}$ of $\bar{x}$, the next better approximation is computed with

$$x^{(k)} = x^{(k-1)} - \frac{f(x^{(k-1)})}{f'(x^{(k-1)})}, \tag{3.66}$$

where f' denotes the derivative of f. This procedure can be applied for general non-linear functions $F : \mathbb{R}^n \to \mathbb{R}^n$, provided that its derivative J, called the Jacobian, is invertible in each approximation $x^{(k-1)}$. In this case, in each iteration we compute the new approximation with

$$x^{(k)} = x^{(k-1)} - J(x^{(k-1)})^{-1} F(x^{(k-1)}). \tag{3.67}$$

The major advantage of the Newton's method in comparison to many other linearization methods is its quadratic convergence. However, this fast convergence depends on how close the initial guess $x^{(0)}$ is to the root we aim to find and to some properties of the function F itself. For example, if F has many roots, the initial guess $x^{(0)}$ should be sufficiently close to the root we want to find.

Let us now apply the Newton's method to our problem (3.62). We can rewrite this problem to a form suitable for using the procedure (3.67): find $(\boldsymbol{u}_{h,f}, \lambda_h) \in \boldsymbol{V}_{h,f} \times \Lambda_h$ that satisfies

$$F\left[(\boldsymbol{u}_{h,f}, \lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] = 0, \qquad \forall (\boldsymbol{v}_{h,f}, \mu_h) \in \boldsymbol{V}_{h,f} \times \Lambda_h, \tag{3.68}$$

where F is given as

$$\begin{aligned} F\left[(\boldsymbol{u}_{h,f}, \lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] = {} & (\boldsymbol{k}(\boldsymbol{u}_{h,f})\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f} + \langle S\lambda_h, \mu_h\rangle + (\nabla \cdot \boldsymbol{u}_{h,f}, \mu_h)_{\Omega_f} \\ & - (\lambda_h, \nabla \cdot \boldsymbol{v}_{h,f})_{\Omega_f} - (f, \mu_h)_{\Omega_f} - \langle \chi_h, \mu_h\rangle. \end{aligned} \tag{3.69}$$

Let $(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)})$ be the current approximation of the solution $(\boldsymbol{u}_{h,f}, \lambda_h)$. For simplicity of notation, we introduce the following

$$\boldsymbol{k}^{-1,(k-1)} = \boldsymbol{K}^{-1} + \xi(\boldsymbol{u}_{h,f}^{(k-1)}), \qquad \text{and} \qquad \boldsymbol{k}_{\partial}^{-1,(k-1)} = \frac{\partial \boldsymbol{k}^{-1,(k-1)}}{\partial \boldsymbol{u}_{h,f}^{(k-1)}} = \xi'(\boldsymbol{u}_{h,f}^{(k-1)}). \tag{3.70}$$

The Jacobian $J\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)}); (\delta\boldsymbol{u}_{h,f}, \delta\lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right]$ is given by taking the Gâteaux variation of $F\left[(\boldsymbol{u}_{h,f}, \lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right]$ at $(\boldsymbol{u}_{h,f}, \lambda_h) = (\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)})$ in the direction of $(\delta\boldsymbol{u}_{h,f}, \delta\lambda_h)$:

$$\begin{aligned} & J\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)}); (\delta\boldsymbol{u}_{h,f}, \delta\lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] = \\ & \left[\frac{F\left[(\boldsymbol{u}_{h,f}^{(k-1)} + \epsilon\delta\boldsymbol{u}_{h,f}, \lambda_h^{(k-1)} + \epsilon\delta\lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] - F\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)}), (\boldsymbol{v}_{h,f}, \mu_h)\right]}{\epsilon}\right]_{\epsilon \to 0}. \end{aligned} \tag{3.71}$$

In our case, this definition yields

$$J\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)});\ (\delta\boldsymbol{u}_{h,f}, \delta\lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] = \\ ([\boldsymbol{k}^{-1,(k-1)} + \boldsymbol{k}_{\partial}^{-1,(k-1)} \otimes \boldsymbol{u}_{h,f}^{(k-1)}]\delta\boldsymbol{u}_{h,f}, \boldsymbol{v}_{h,f})_{\Omega_f} + \langle S\delta\lambda_h, \mu_h\rangle \\ + (\nabla\cdot\delta\boldsymbol{u}_{h,f}, \mu_h)_{\Omega_f} - (\delta\lambda_h, \nabla\cdot\boldsymbol{v}_{h,f})_{\Omega_f}, \tag{3.72}$$

where $\otimes$ denotes the standard tensor product. Hence, at each Newton iteration, we solve the following linear problem: find $(\delta\boldsymbol{u}_{h,f}, \delta\lambda_h) \in \boldsymbol{V}_{h,f} \times \Lambda_h$ such that

$$J\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)}); (\delta\boldsymbol{u}_{h,f}, \delta\lambda_h), (\boldsymbol{v}_{h,f}, \mu_h)\right] = -F\left[(\boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k-1)}), (\boldsymbol{v}_{h,f}, \mu_h)\right] \\ \forall(\boldsymbol{v}_{h,f}, \mu_h) \in \boldsymbol{V}_{h,f} \times \Lambda_h. \tag{3.73}$$

This system can be solved using a suitable direct or iterative method. Finally, the new approximation $(\boldsymbol{u}_{h,f}^{(k)}, \lambda_h^{(k)})$ is constructed as

$$\boldsymbol{u}_{h,f}^{(k)} = \boldsymbol{u}_{h,f}^{(k-1)} + \delta\boldsymbol{u}_{h,f}, \tag{3.74a}$$

$$\lambda_h^{(k)} = \lambda_h^{(k-1)} + \delta\lambda_h, \tag{3.74b}$$

and the procedure is repeated until we reach the prescribed tolerance on the current residual $r^{(k)} = (\delta\boldsymbol{u}_{h,f}, \delta\lambda_h)$. At convergence, the subdomain solutions can again be retrieved via (3.65).

3.4.3 L-scheme

The remaining linearization method we present concerns the *L-scheme*. This method modifies the basic fixed-point iterations by adding a stabilization term. That is, for some constant $L > 0$, the algorithm (3.63) is updated to

$$x^{(k)} = f(x^{(k-1)}) + L(x^{(k)} - x^{(k-1)}), \qquad k = 1, 2, \ldots \tag{3.75}$$

with a given initial guess $x^{(0)}$. If the function is Lipschitz continuous, the parameter L can be seen as the Lipschitz constant and the above iterations converge to a unique fixed point of f. In fact, the L-scheme can be seen as a quasi-Newton method with the parameter L mimicking the Jacobian. Therefore, the L-scheme is simpler and cheaper to implement in comparison to the Newton's method, but the quadratic convergence rate is reduced to only linear convergence, which also depends on the value of the stabilization parameter L. Moreover, the L-scheme can be used in combination with the Newton's method, by first performing a few iterations with the L-scheme and then switching to the Newton's method. In this way one increases the robustness of the Newton's method, see [43]. A more detailed analysis of the convergence of the L-scheme in the context of porous media can also be found in [17, 42, 52, 58, 60] and in Paper D.

Lastly, we apply the L-scheme to our non-linear problem (3.62). Given an initial guess $(\boldsymbol{u}_{h,f}^{(0)}, \lambda_h^{(0)})$ and the parameter $L > 0$, the k-th L-scheme iteration considers solving

$$(\boldsymbol{k}(\boldsymbol{u}_{h,f}^{(k-1)})\boldsymbol{u}_{h,f}^{(k-1)}, \boldsymbol{v}_{h,f})_{\Omega_f} + (L(\boldsymbol{u}_{h,f}^{(k)} - \boldsymbol{u}_{h,f}^{(k-1)}), \boldsymbol{v}_{h,f})_{\Omega_f} - (\lambda_h^{(k)}, \nabla \cdot \boldsymbol{v}_{h,f})_{\Omega_f} = 0 \qquad \forall \boldsymbol{v}_{h,f} \in \boldsymbol{V}_{h,f}, \tag{3.76a}$$

$$(\nabla \cdot \boldsymbol{u}_{h,f}^{(k)}, \mu_h)_{\Omega_f} + \langle S\lambda_h^{(k)}, \mu_h \rangle = (f, \mu_h)_{\Omega_f} + \langle \chi_h, \mu_h \rangle \qquad \forall \mu_h \in \Lambda_h, \tag{3.76b}$$

until a prescribed tolerance on the residual $r^{(k)} = (\boldsymbol{u}_{h,f}^{(k)} - \boldsymbol{u}_{h,f}^{(k-1)}, \lambda_h^{(k)} - \lambda_h^{(k-1)})$ is reached. Again, the above linear system can now be solved directly or iteratively, and the subdomain solutions can be retrieved with (3.65).

Chapter 4

Summary and Outlook

We finalize Part I with a short overview of the scientific results of the thesis. We first summarize the main findings that are presented as five scientific articles in Part II, and then end with an outlook on the future research related to those findings.

4.1 Summary of the Papers

Paper A: Block Preconditioners for Mixed-Dimensional Discretization of Flow in Fractured Porous Media

In this paper, we provide a set of block preconditioners for Krylov methods based on the inf-sup stability of the mixed finite element discretization of the mixed-dimensional flow problem in [16].

To derive the preconditioners, we use the general approach to preconditioning saddle point problems [45]. Such problems, as the one in our paper, often consist of a coefficient operator with an unbounded spectrum. Therefore, the condition number of the linear system tends to infinity when refining the mesh, which makes the Krylov subspace methods difficult to converge. Using the well-posedness theory, this approach delivers a preconditioner that bounds the condition number of the preconditioned system independently of the discretization parameter. The structure of the preconditioner is based on the energy norm used in which the system is well-posed, hence the name norm-equivalent preconditioner. The standard choice is the Riesz operator corresponding to the inner product on the chosen discrete space, which is studied in Paper B, however here we derive simpler and efficient preconditioners that are easier to implement by reformulating the discrete problem.

We show that the alternative formulation of the discrete problem is still well-posed, but in different norms. These norms are based on the equivalency of the L^2-inner product on the flux space with a diagonal approximation. In the norm induced by the new inner product, we define a discrete gradient operator and a "broken" H^1 norm on the discrete pressure space. The inf-sup condition is now satisfied in these new norms and the constants are independent of the discretization and physical parameters, but remain dependent on the shape regularity

of the mesh. Using the well-posedness theory, we can derive a norm-equivalent robust block diagonal preconditioner for our problem that resembles a Schur complement with regards to the pressure variable. In the similar manner, we derive block triangular preconditioner that are field-of-value equivalent to the block diagonal one using the theory in [44].

Several numerical examples justify the theoretical results of this paper. We show on both two- and three-dimensional porous medium domains with fractures that the performance of the block preconditioners is stable and robust. We also show that the computational time of the preconditioned GMRES method has linear dependency on the number of degrees of freedom. Lastly, we test the dependency on the mesh shape regularity in an example of a complex fracture network structure and observe that even with a significant percentage of badly shaped elements, the block preconditioners still converge, but require a higher number of iterations.

Paper B: Mixed-Dimensional Auxiliary Space Preconditioners

As a follow-up to Paper A, this paper introduces nodal auxiliary space preconditioners for discretizations of mixed-dimensional partial differential equations.

The key ingredient to our preconditioners is the extension of the regular decomposition of differential k-forms to a mixed-dimensional setting. In the continuous case, we first introduce the space of regular mixed-dimensional k-forms as a product space of local regular subspaces on each sub-manifold of the mixed-dimensional domain. Such a construction allows to prove a mixed-dimensional regular decomposition exists since it can be combined from the conventional regular decompositions on each sub-manifold. Furthermore, we use the conforming mixed finite element spaces in [15] to obtain a discrete decomposition based on nodal Lagrange elements.

Having this at hand, we derive preconditioners for mixed-dimensional elliptic problems based on auxiliary space theory in [35]. We use the derived regular decomposition to make the auxiliary product space with a composite auxiliary inner product bounded by the inner product on the original function space. The linear operator induced by the auxiliary inner product is then a good approximation of the coefficient operator of our elliptic problem, which makes it possible to use it as a preconditioner. The bound is independent of discretization parameter, therefore making it a robust preconditioner and the condition number unconditionally bounded.

We apply the theory to a practical problem of flow in fractured porous media. As mentioned in Paper A, using the well-posedness theory we can derive a norm-equivalent preconditioner with a block diagonal structure. The blocks correspond to norms used to show that the inf-sup condition holds, that in this case are a mixed-dimensional $H(\mathrm{div})$-norm for the flux variable and a L^2-norm for the pressure. The main challenge of implementing a preconditioner on a discrete $H(\mathrm{div})$ space is overcome by the proposed auxiliary space preconditioner. In this way, the preconditioner is decomposed into inverses of several operators, including a smoother operator to handle high-frequency terms and additional operators projected from face or edge to nodal degrees of freedom. Computation-wise, the inverses are then replaced by the spectrally equivalent operators using standard solving techniques, such

as algebraic multigrid method (AMG).

The appropriate choice of the spectrally equivalent operators still requires investigation and a thorough comparison, however we rely on the available methods to show the robustness of the preconditioners. This is evident in several numerical examples in the context of flow in fractured porous media.

Paper C: A Multiscale Flux Basis for Mortar Mixed Discretizations of Reduced Darcy-Forchheimer Fracture Models

Paper C proposes a domain decomposition method to solve a non-linear flow problem in fractured porous media based on the multiscale mortar mixed finite element method in [30].

The mixed-dimensional linear flow model is extended to account for large flow rates in the conductive fractures. Therefore, Darcy's law is replaced by Forchheimer's law for modeling the flow in the fractures, while in the surrounding rock matrix we still keep the standard Darcy's law. The coupling between the models in the rock matrix and fractures is imposed by Robin type conditions on normal flux and pressure across the interface. We discretize the problem with stable mixed finite element pair – lowest order Raviart-Thomas-Nedelec finite elements for the flow velocity, and piecewise constants for the pressure variable. We also consider non-conforming grids between the rock matrix subdomains and the fracture network using the mortar variable as fracture pressure. The resulting system of equations is a large, non-linear and coupled problem. We reformulate the coupled system into an interface problem by eliminating the rock matrix variables via an linear Robin-to-Neumann operator. We are left with a smaller but still non-linear fracture flow problem that requires a robust solver and adequate handling of the contributions from the rock matrix subdomains.

The main novelties of this paper are the following. Firstly, we present and analyze two methods to solve the non-linear interface problem. In Method 1, we use the fixed-point method to linearize the interface problem, and the resulting linear system in each iteration is then solved by the GMRES method. On the other hand, in Method 2, the linearization is done by the Newton's method which in each iteration requires solving a linear system with a Jacobian. This can again be solved by the GMRES method. Although the proposed methods result in different linear systems, in each of them the inner iterations of the GMRES method consist of several actions of the Robin-to-Neumann operator that handles the mixed-dimensional coupling. This operator maps the fracture pressure data as on the interface of each rock matrix subdomain, solve the rock subdomain problems with fracture contributions as Robin boundary conditions, and then project back to the interface the normal flux subdomain contributions. Therefore, it is evident that the main computational cost of both methods are the actions of the Robin-to-Neumann operator in each of the inner GMRES iterations.

Secondly, we reduce the computational cost of both methods by introducing multiscale flux basis. In particular, we separate the linear and non-linear part of the interface problem. Linear part that consists of the rock matrix subdomain contributions is then computed offline before the iterations of the linearization methods start. This is possible since the non-linearity only occurs in the fractures and interface variable can be decomposed as the linear combination of the multiscale flux basis functions, one per each interface degree of freedom. Then,

the action of the Robin-to-Neumann operator can be computed on each of the basis functions and, due to linearity of the operator, the discrete version of the mapping can be stored and used in each GMRES iteration within both of the linearization methods. This can be used regardless of the model imposed within the fracture network, which allows an easy implementation in the more general fracture flow models, as it is done in Paper D. Additionally, using the mortar method on the interface, it is possible to coarsen the mortar grid to have a smaller basis and, hence, less storage and time consumption of the action of the interface mapping operator.

We investigate the performance of the proposed methods on several numerical examples where the performance measure is set to be the rock matrix subdomain solves, as they are by far the most costly part of the both methods. We show that using multiscale flux basis eliminates the unnecessary repeated computations of the rock matrix subdomains contributions which reduces the computational cost of GMRES solver in comparison to the standard domain decomposition method. This can also be seen in the study on coarsening and refinement of the mortar grid. The efficiency of the proposed methods combined with the multiscale flux basis is shown. Moreover, we study the effects effect of the physical parameters, such as the fracture permeability and the non-linearity constant. However, strong non-linearity and high permeability influence the total number of iterations of the solver, thus we aim to improve it in Paper D.

Paper D: Robust Linear Domain Decomposition Schemes for Reduced Nonlinear Fracture Flow Models

In extension to Paper C, this paper considers a general non-linear flow model in the fracture network. In addition to that, a compressibility term is added to both rock matrix and fractures domain, making the problem mixed-dimensional, non-linear and time-dependent. We develop two algorithms based on the domain decomposition approach in Paper C and the so-called L-scheme [52].

Firstly, the discrete problem is set up with mortar mixed finite element method in space and backward Euler method in time. Then, we reduce the problem to only the fracture network and consider the contributions from the rock matrix via the Robin-to-Neumann operator. In each time step, we end up with a non-linear problem that we linearize in two ways. In the monolithic method called MoLDD, we employ the L-scheme only as the linearization procedure similar to the Newton's method in Paper C, but with using a parameter $L > 0$ instead of a full Jacobian. MoLDD method shows to be unconditionally stable, but still requires an inner linear solver, such as GMRES, where again the computational cost is dominated by repeatedly solving the rock matrix subdomain problems. It is resolved by using multiscale flux basis functions in each time step as in Paper C, but strong non-linearity and conductivity of the fracture network can deteriorate the efficiency of the solver. Therefore, we also propose iterative method named ItLDD, where L-scheme is used both to handle the non-linear term and the coupling between the rock and the matrix imposed by the Robin-to-Neumann operator. In this way, the linearization and the domain decomposition part of the algorithm are combined in a single loop. If a fixed time step is used, the multiscale flux basis can be

computed once for all time steps, reducing significantly the computational cost of the methods.

In the numerical examples, we study the performance of of the proposed methods on two flow models in the fractures, Forchheimer's law and Cross' law. In accordance with the theoretical results, the methods show to be stable and robust with regards to discretization and physical parameters. Since both methods require a choice on the parameters for the L-scheme, we show that we can indeed optimize the methods for a specific parameter setup that is close to the theoretically derived one. MoLDD method shows a slightly better performance than the ItLDD method in most of the tests, especially when considering non-conforming grids between the rock matrix and the fracture network.

4.2 Outlook

The investigations into preconditioners and solvers for the mixed-dimensional flow models open up many possible directions to future research in the field. We provide an overview of some more immediate extensions, motivated by the findings of the scientific papers in this thesis.

First, there is still an open question on the optimal solver for the mixed-dimensional second-order elliptic problems. In Paper B, we have considered auxiliary space preconditioners that assume a good numerical method for the composite Laplacian-type problem is available. In fixed-dimensional case, a common choice is to apply AMG methods as spectrally equivalent approximations of the inverse of the second-order elliptic operator, which have shown to be quite fast and stable. However, this may not be the case in mixed-dimensional setting, which could cause the slow convergence of the AMG methods and, in turn, slow down the whole preconditioned iterative method. Therefore, it is important to investigate into the design of a proper method to solve such a composite problem with codimension one coupling.

Furthermore, a natural extension to the flow problem is to consider coupled models in the mixed-dimensional setting, such as the coupling with linear elasticity into a Biot's consolidation model or coupling Stokes flow inside the fracture to the porous medium flow in the rock. This still results in a saddle-point problem where the theory on preconditioning in Papers A and B can be applied. This is also important in general realistic applications of the mixed-dimensional theory in the subsurface reservoir models since many other physical processes drive the fluid flow, e.g. transport and chemical reactions.

On the other hand, the general framework presented in Paper B gives new insights into preconditioners for other mixed-dimensional elliptic problems that do not have to concern the application in porous media. For example, the linear elasticity models that describe the deformation of composite materials with thin inclusions [14] or Maxwell's equations of electromagnetic scattering by perfectly conducting screens [19].

Finally, it is possible to update the method in Papers C and D by introducing a multi-rate approach with domain decomposition in time. This way we can handle fast and slow flow within the fractures on different time scales. In addition, it may be beneficial to develop an adaptive stopping criteria for the iterative method using a posteriori error estimate theory.

Bibliography

[1] Adler, J. H., Hu, X., and Zikatanov, L. T. (2009–2019). HAZMATH: A simple finite element, graph and solver library. https://hazmath.net/.

[2] Ahmed, E., Jaffré, J., and Roberts, J. E. (2017). A reduced fracture model for two-phase flow with different rock types. *Mathematics and Computers in Simulation 137*, 49–70. doi: 10.1016/j.matcom.2016.10.005. MAMERN VI-2015: 6th International Conference on Approximation Methods and Numerical Modeling in Environment and Natural Resources.

[3] Alboin, C., Jaffré, J., Roberts, J. E., and Serres, C. (2002). Modeling fractures as interfaces for flow and transport in porous media. In Chen, Z. and Ewing, R. E. (editors), *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.

[4] Alboin, C., Jaffré, J., Roberts, J. E., Wang, X., and Serres, C. (1999). Domain Decomposition for Flow in Porous Media with Fractures. In Lai, C.-H., Bjorstad, P. E., Cross, M., and Widlund, O. B. (editors), *Domain Decomposition Methods in Sciences and Engineering*, pages 371–379. Domain Decomposition Press.

[5] Antonietti, P., De Ponti, J., Formaggia, L., and Scotti, A. (2019). Preconditioning techniques for the numerical solution of flow in fractured porous media. *MOX-Report No. 17/2019* .

[6] Arbogast, T., Pencheva, G., Wheeler, M. F., and Yotov, I. (2007). A multiscale mortar mixed finite element method. *Multiscale Model. Simul. 6*(1), 319–346. doi: 10.1137/060662587.

[7] Arnold, D. N., Falk, R. S., and Winther, R. (1997). Preconditioning in H(div) and Applications. *Math. Comput. 66*(219), 957–984. doi: 10.1090/S0025-5718-97-00826-0.

[8] Arrarás, A., Gaspar, F. J., Portero, L., and Rodrigo, C. (2018). Monolithic mixed-dimensional multigrid methods for single-phase flow in fractured porous media. *arXiv:1811.01264 [math.NA]* .

[9] Bastian, P., Chen, Z., Ewing, R. E., Helmig, R., Jakobs, H., and Reichenberger, V. (2000). Numerical Simulation of Multiphase Flow in Fractured Porous Media. In Chen, Z., Ewing, R. E., and Shi, Z.-C. (editors), *Numerical Treatment of Multiphase Flows in Porous Media*, pages 50–68. Springer Berlin Heidelberg, Berlin, Heidelberg.

[10] Bauer, S., Dahmke, A., and Kolditz, O. (2017). Subsurface energy storage: geological storage of renewable energy – capacities, induced effects and implications. *Environmental Earth Sciences 76*(695). doi: 10.1007/s12665-017-7007-9.

[11] Berre, I., Boon, W., Flemisch, B., Fumagalli, A., Gläser, D., Keilegavlen, E., Scotti, A., Stefansson, I., and Tatomir, A. (2018). Call for participation: Verification benchmarks for single-phase flow in three-dimensional fractured porous media. *arXiv:1809.06926 [math.NA]* .

[12] Berre, I., Doster, F., and Keilegavlen, E. (2018). Flow in fractured porous media: A review of conceptual models and discretization approaches. *arXiv:1805.05701 [physics.geo-ph]* .

[13] Boffi, D., Brezzi, F., and Fortin, M. (2013). Mixed Finite Element Methods and Applications, volume 44 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-36519-5. doi: 10.1007/978-3-642-36519-5.

[14] Boon, W. M. and Nordbotten, J. M. (2019). Stable Mixed Finite Elements for Linear Elasticity with Thin Inclusions. *arXiv preprint:1903.01757 [math.NA]* .

[15] Boon, W. M., Nordbotten, J. M., and Vatne, J. E. (2018). Functional Analysis and Exterior Calculus on Mixed-Dimensional Geometries. *arXiv preprint:1710.00556v3 [math.AP]* .

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

[17] Borregales, M., Radu, F. A., Kumar, K., and Nordbotten, J. M. (2018). Robust iterative schemes for non-linear poromechanics. *Comput. Geosci. 22*(4), 1021–1038. doi: 10.1007/s10596-018-9736-6.

[18] Brezzi, F. (1974). On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrangian Multipliers. *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge 8*(R-2), 129–151.

[19] Buffa, A. and Christiansen, S. (2003). The electric field integral equation on Lipschitz screens: definitions and numerical approximation. *Numerische Mathematik 94*(2), 229–267. doi: 10.1007/s00211-002-0422-0.

[20] D'Angelo, C. and Scotti, A. (2011). A mixed finite element method for Darcy flow in fractured porous media with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis 46*(2), 465–489. doi: 10.1051/m2an/2011148.

[21] Davis, T. A. (2006). Direct Methods for Sparse Linear Systems. Fundamentals of Algorithms. SIAM. ISBN 978-0-89871-613-9. doi: 10.1137/1.9780898718881.

[22] Eisenstat, S., Elman, H., and Schultz, M. (1983). Variational Iterative Methods for Nonsymmetric Systems of Linear Equations. *SIAM Journal on Numerical Analysis 20*(2), 345–357. doi: 10.1137/0720023.

[23] Elman, H. C. (1982). Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations. Ph.D. thesis, New Haven, CT, USA. AAI8222744.

[24] Ervin, V., Lee, H., and Salgado, A. (2016). Generalized Newtonian fluid flow through a porous medium. *Journal of Mathematical Analysis and Applications 433*(1), 603–621. doi: 10.1016/j.jmaa.2015.07.054.

[25] Flemisch, B., Berre, I., Boon, W., Fumagalli, A., Schwenck, N., Scotti, A., Stefansson, I., and Tatomir, A. (2018). Benchmarks for single-phase flow in fractured porous media. *Advances in Water Resources 111*, 239–258. doi: 10.1016/j.advwatres.2017.10.036.

[26] Formaggia, L., Fumagalli, A., Scotti, A., and Ruffo, P. (2014). A reduced model for Darcy's problem in networks of fractures. *ESAIM Math. Model. Numer. Anal. 48*(4), 1089–1116. doi: 10.1051/m2an/2013132.

[27] Frih, N., Martin, V., Elizabeth Roberts, J., and Saada, A. (2012). Modeling fractures as interfaces with nonmatching grids. *Computational Geosciences 16*, 1043–1060. doi: 10.1007/s10596-012-9302-6.

[28] Frih, N., Roberts, J. E., and Saada, A. (2008). Modeling fractures as interfaces: a model for Forchheimer fractures. *Computational Geosciences 12*(1), 91–104. doi: 10.1007/s10596-007-9062-x.

[29] Ganis, B. and Yotov, I. (2009). Implementation of a mortar mixed finite element method using a Multiscale Flux Basis. *Computer Methods in Applied Mechanics and Engineering 198*(49), 3989–3998. doi: 10.1016/j.cma.2009.09.009.

[30] Ganis, B. and Yotov, I. (2009). Implementation of a mortar mixed finite element method using a Multiscale Flux Basis. *Computer Methods in Applied Mechanics and Engineering 198*(49), 3989–3998. doi: 10.1016/j.cma.2009.09.009.

[31] Gläser, D., Helmig, R., Flemisch, B., and Class, H. (2017). A discrete fracture model for two-phase flow in fractured porous media. *Advances in Water Resources 110*, 335–348. doi: https://doi.org/10.1016/j.advwatres.2017.10.031.

[32] Golub, G. H. and Van Loan, C. F. (1996). Matrix Computations. The Johns Hopkins University Press, third edition.

[33] Hackbusch, W. (2016). Iterative Solution of Large Sparse Systems of Equations, volume 95 of *Applied Mathematical Sciences*. Springer-Verlag. ISBN 978-3-319-28483-5. doi: 10.1007/978-3-319-28483-5.

[34] Hestenes, M. R. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards 49*(6), 409–436.

[35] Hiptmair, R. and Xu, J. (2007). Nodal Auxiliary Space Preconditioning in H(curl) and H(div) Spaces. *SIAM Journal on Numerical Analysis 45*(6), 2483–2509. doi: 10.1137/060660588.

[36] Jones, E., Oliphant, T., Peterson, P., et al. (2001–). SciPy: Open source scientific tools for Python. [Online; accessed December 19, 2019].

[37] Keilegavlen, E., Berge, R., Fumagalli, A., Starnoni, M., Stefansson, I., Varela, J., and Berre, I. (2019). PorePy: An Open-Source Software for Simulation of Multiphysics Processes in Fractured Porous Media. *arXiv:1908.09869 [math.NA]* .

[38] Knabner, P. and Roberts, J. E. (2014). Mathematical analysis of a discrete fracture model coupling Darcy flow in the matrix with Darcy–Forchheimer flow in the fracture. *ESAIM:M2AN 48*(5), 1451–1472. doi: 10.1051/m2an/2014003.

[39] Kolev, T. and Vassilevski, P. (2012). Parallel Auxiliary Space AMG Solver for H(div) Problems. *SIAM Journal on Scientific Computing 34*(6), A3079–A3098. doi: 10.1137/110859361.

[40] Kourounis, D., Fuchs, A., and Schenk, O. (2018). Towards the Next Generation of Multiperiod Optimal Power Flow Solvers. *IEEE Transactions on Power Systems PP*(99), 1–10. doi: 10.1109/TPWRS.2017.2789187.

[41] Li, X. S. (2005). An Overview of SuperLU: Algorithms, Implementation, and User Interface. *ACM Transactions on Mathematical Software 31*(3), 302–325.

[42] List, F., Kumar, K., Pop, I. S., and Radu, F. A. (2018). Upscaling of unsaturated flow in fractured porous media. *arXiv preprint:1807.05993 [math.NA]* .

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

[44] Loghin, D. and Wathen, A. J. (2004). Analysis of preconditioners for saddle-point problems. *SIAM J. Sci. Comput. 25*(6), 2029–2049. doi: 10.1137/S1064827502418203.

[45] Mardal, K.-A. and Winther, R. (2011). Preconditioning discretizations of systems of partial differential equations. *Numer. Linear Algebra Appl. 18*(1), 1–40. doi: 10.1002/nla.716.

[46] Nédélec, J.-C. (1980). Mixed finite elements in $\mathbf{R}^3$. *Numer. Math. 35*(3), 315–341. doi: 10.1007/BF01396415.

[47] Nordbotten, J. M. and Boon, W. M. (2018). Modeling, Structure and Discretization of Mixed-dimensional Partial Differential Equations. In Bjorstad, P. E., Brenner, S. C., Halpern, L., Kim, H. H., Kornhuber, R., Rahman, T., and Widlund, O. B. (editors), *Domain Decomposition Methods in Sciences and Engineering XXIV*, volume 125 of *Lecture Notes in Computational Science and Engineering*, pages 22–34. Domain Decomposition Press.

[48] Nordbotten, J. M., Boon, W. M., Fumagalli, A., and Keilegavlen, E. (2018). Unified approach to discretization of flow in fractured porous media. *Computational Geosciences* pages 1–13. doi: 10.1007/s10596-018-9778-9.

[49] Nordbotten, J. M. and Celia, M. A. (2011). Geological Storage of CO2: Modeling Approaches for Large-Scale Simulation. Wiley. ISBN 9781118137079. doi: 10.1002/9781118137086.

[50] Paige, C. C. and Saunders, M. A. (1975). Solution of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal. 12*, 617–629.

[51] Pencheva, G. and Yotov, I. (2003). Balancing domain decomposition for mortar mixed finite element methods. *Numer. Linear Algebra Appl. 10*, 159–180. doi: 10.1002/nla.316.

[52] Pop, I. S., Radu, F., and Knabner, P. (2004). Mixed finite elements for the Richards' equation: linearization procedure. *J. Comput. Appl. Math. 168*(1-2), 365–373. doi: 10.1016/j.cam.2003.04.008.

[53] Quarteroni, A. and Valli, A. (1999). Domain Decomposition Methods for Partial Differential Equations. Oxford University Press. ISBN 9780198501787.

[54] Raviart, P.-A. and Thomas, J.-M. (1977). A mixed finite element method for 2nd order elliptic problems. In *Mathematical aspects of finite element methods, Lecture Notes in Math., Vol. 606*, pages 292–315. Springer, Berlin.

[55] Saad, Y. and Schulz, m. C. (1986). GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput. 7*, 856–869. doi: 10.1137/0907058.

[56] Sandve, T. H., Berre, I., and Nordbotten, J. M. (2012). An efficient multi-point flux approximation method for Discrete Fracture–Matrix simulations. *Journal of Computational Physics 231*(9), 3784 – 3800. doi: https://doi.org/10.1016/j.jcp.2012.01.023.

[57] Schwenck, N., Flemisch, B., Helmig, R., and Wohlmuth, B. I. (2015). Dimensionally reduced flow models in fractured porous media: crossings and boundaries. *Computational Geosciences 19*(6), 1219–1230. doi: 10.1007/s10596-015-9536-1.

[58] Seus, D., Mitra, K., Pop, I. S., Radu, F. A., and Rohde, C. (2018). A linear domain decomposition method for partially saturated flow in porous media. *Comput. Methods Appl. Mech. Engrg. 333*, 331–355. doi: 10.1016/j.cma.2018.01.029.

[59] Spivak, M. (2018). Calculus on manifolds: a modern approach to classical theorems of advanced calculus. CRC Press.

[60] Storvik, E., Both, J. W., Kumar, K., Nordbotten, J. M., and Radu, F. A. (2019). On the optimization of the fixed-stress splitting for Biot's equations. *International Journal for Numerical Methods in Engineering* pages 179–194. doi: 10.1002/nme.6130.

[61] Stüben, K. (2001). A review of algebraic multigrid. *Journal of Computational and Applied Mathematics 128*, 281–309. doi: 10.1016/S0377-0427(00)00516-1.

[62] Tuminaro, R. S., Xu, J., and Zhu, Y. (2009). Auxiliary Space Preconditioners for Mixed Finite Element Methods. In Bercovier, M., Gander, M. J., Kornhuber, R., and Widlund, O. (editors), *Domain Decomposition Methods Sci. Eng. XVIII*, pages 99–109. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-02677-5.

[63] Ueckert, M. and Baumann, T. (2019). Hydrochemical aspects of high-temperature aquifer storage in carbonaceous aquifers: evaluation of a field study. *Geothermal Energy 7*(4). doi: 10.1186/s40517-019-0120-0.

[64] Wheeler, M. F. and Glowinski, R. (1988). Domain Decomposition and Mixed Finite Element Methods for Elliptic Problems. In Glowinski, R., Golub, G. H., Meurant, G. A., and Periaux, J. (editors), *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 144–172. SIAM: Philadephia.

[65] Whitaker, S. (1986). Flow in porous media I: A theoretical derivation of Darcy's law. *Transport in Porous Media 1*(1), 3–25. doi: 10.1007/BF01036523.

[66] Xu, J. (1996). The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing 56*(3), 215–235. doi: 10.1007/BF02238513.

[67] Xu, J. and Zikatanov, L. (2017). Algebraic multigrid methods. *Acta Numerica 26*, 591–721. doi: 10.1017/S0962492917000083.

Part II

Scientific Results

Paper A

Block Preconditioners for Mixed-Dimensional Discretization of Flow in Fractured Porous Media

A. Budiša, X. Hu

Computational Geosciences. In review.

arXiv:1905.13513 [math.NA].

Computational Geosciences manuscript No.
(will be inserted by the editor)

Block Preconditioners for Mixed-dimensional Discretization of Flow in Fractured Porous Media

Ana Budiša · Xiaozhe Hu

Received: date / Accepted: date

Abstract In this paper, we are interested in an efficient numerical method for the mixed-dimensional approach to modeling single-phase flow in fractured porous media. The model introduces fractures and their intersections as lower-dimensional structures, and the mortar variable is used for flow coupling between the matrix and fractures. We consider a stable mixed finite element discretization of the problem, which results in a parameter-dependent linear system. For this, we develop block preconditioners based on the well-posedness of the discretization choice. The preconditioned iterative method demonstrates robustness with regards to discretization and physical parameters. The analytical results are verified on several examples of fracture network configurations, and notable results in reduction of number of iterations and computational time are obtained.

Keywords porous medium · fracture flow · mixed finite element · algebraic multigrid method · iterative method · preconditioning

Mathematics Subject Classification (2010) 65F08, 65F10, 65N30

1 Introduction

Fracture flow has become a case of intense study recently due to many possible subsurface applications, such as CO_2 sequestration or geothermal energy stor-

The first author acknowledges the financial support from the TheMSES project funded by Norwegian Research Council grant 250223. The work of the second author is partially supported by the National Science Foundation under grant DMS-1620063.

A. Budiša
Department of Mathematics, University of Bergen, P. O. Box 7800, N-5020 Bergen, Norway.
E-mail: Ana.Budisa@uib.no

X. Hu
Department of Mathematics, Tufts University, 503 Boston Ave, Medford, MA 02155, USA.
E-mail: Xiaozhe.Hu@tufts.edu

age and production. It has become clear that the dominating role of fractures in the flow process in the porous medium calls for reexamination of existing mathematical models, numerical methods and implementations in these cases.

Considering modeling and analysis, a popular and effective development is reduced fracture models [9, 18, 22] that represent fractures and fracture intersections as lower-dimensional manifolds embedded in a porous medium domain. The immediate advantages of such modeling are in more accurate representation of flow patterns, especially in case of highly conductive fractures, and easier handling of discontinuities over the interfaces. This has also allowed for implementation of various discretization methods, from finite volume methods [22,30] to (mixed) finite element methods [18] and other methods [17,19]. These methods mostly differ in two aspects: whether the fractures conform to the discrete grid of the porous medium [9] or are placed arbitrarily within the grid [12,16,31], or whether pressure or flux continuity is preserved. Comparison studies of different discretization methods and their properties can be found in [5,15,28].

Although there is a wide spectrum of discretization methods, little has been done to develop robust and efficient solvers. This aspect of implementation can be very important since applications of fractured porous media usually include large-scale simulations of subsurface reservoirs and the resulting discretized linear systems of equations can become ill-conditioned and quite difficult to solve. The linear system represents a discrete version of the partial differential equation (PDE) operator that has unbounded spectrum. Thus, its condition number tends to infinity when the mesh size is approaching zero. Moreover, the variability of the physical parameters, such as the permeabilities and aperture, can additionally influence the scale of the condition number of the system. Instead of using direct methods, we consider Krylov subspace iterative methods to solve such large scale problems. Since the convergence rate of the Krylov subspace methods depends on the condition number of the system, suitable preconditioning techniques are usually required to achieve a good performance. A recent study on a geometric multigrid method [4] for the fracture problem shows how standard iterative methods can be extended and perform well on mixed-dimensional discretizations, but still there are limitations that need to be overcome for general fractured porous media simulations.

In this paper, we aim to provide a general approach to preconditioning the mixed-dimensional flow problems based on suitable mixed finite element method discretization developed in [9]. Beside introducing the mixed-dimensional geometry, the main aspects of the discretization are flux coupling between subdomains using a mortar variable and inf-sup stability of the associated saddle-point problem. Moreover, this framework has been shown to be well incorporated within functional analysis as a concept of mixed-dimensional partial differential equations [8], allowing even further applications in poroelasticity and transport problems.

We propose a set of block preconditioners for Krylov subspace methods for solving the linear system of equations arising from the chosen discretization. Following the theory in [26] and [25], we derive uniform block precondition-

ers based on the well-posedness of an alternative but equivalent formulation. Proper weighted norm is chosen so that the well-posedness constants are robust with respect to the physical and discretization parameters but depend on the shape regularity of the meshes. Both block diagonal and triangular preconditioners are developed based on the framework [25, 26]. Those block preconditioners are not only theoretically robust and effective but can also be implemented straightforwardly by taking advantage of the block structure of the problem.

The rest of the paper is organized as follows. In Section 2 we first introduce the mixed-dimensional geometry and the governing equations of the single-phase flow in fractured porous media followed by the variational formulation and the stable mixed finite element discretization of the problem. The framework of the block preconditioners is briefly recalled in Section 3 and its application to mixed-dimensional discretization of flow in fractured porous media is proposed and analyzed in Section 4. We verify the theoretical results by testing several numerical examples in Section 5 and finalize the paper with concluding remarks in Section 6.

2 Preliminaries

In this section, we set up the problem of flow in fractured porous media following [9]. Let Ω^n be a domain of the porous medium of dimension $n = 2, 3$ that can be decomposed by fractures into $\Omega_i^n, i \in I^n$. The fractures and their intersections are represented as lower d-dimensional manifolds $\Omega_i^d, i \in I^d$, $0 \leqslant d < n$, and inherit the similar decomposition structure as the porous medium Ω^n (see Figure 1). Here, we use I^d as a local index set in dimension $0 \leqslant d \leqslant n$. Furthermore, we define Γ_{ij}^d for $j \in J_i^d \subseteq I^d$ as interfaces between Ω_i^{d+1} and adjacent Ω_j^d. Union over the subscript set I^d represents all d-dimensional subdomains, that is

$$\Omega^d = \bigcup_{i \in I^d} \Omega_i^d, \tag{2.1}$$

$$\Gamma^d = \bigcup_{i \in I^d} \Gamma_i^d = \bigcup_{i \in I^d} \bigcup_{j \in J_i^d} \Gamma_{ij}^d. \tag{2.2}$$

Finally, the fractured porous medium domain Ω with interface Γ is defined as

$$\Omega = \bigcup_{d=0}^{n} \Omega^d, \qquad \Gamma = \bigcup_{d=0}^{n-1} \Gamma^d. \tag{2.3}$$

Remark 2.1 . *Even though the theoretical results in [8, 9] allow for a more complex geometrical structure, for the sake of simplicity we restrict the model to domains of rectangular type. That is, we approximate fractures as lines on a plane for $n = 2$ or flat surfaces in a box for $n = 3$. However, we allow for any configuration of fractures or fracture intersections within, for example, very*

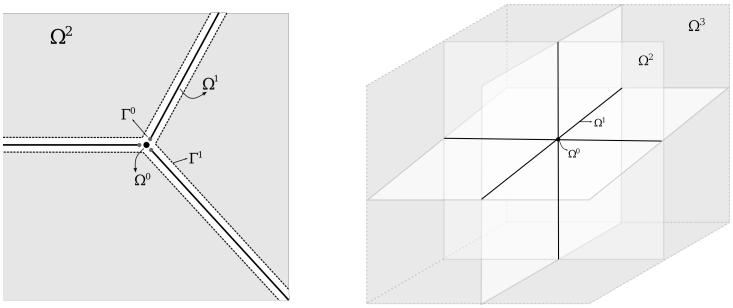


Fig. 1: An illustration of the dimensional decomposition of the domain of the fractured porous media, in two (left) and three (right) dimensions. The dimension of each subdomain Ω^d is given in the superscript d. In the case of intersecting fractures, Γ^d is set as a union of interfaces adjacent to all subdomains Ω^d.

acute angles of fracture intersections, multiple intersecting fractures or T-type intersections.

Now that we have set up the dimensional decomposition framework for the fractured porous medium, we introduce the governing laws in the subdomains and fractures. First, notation and properties of the physical parameters are introduced. For the sake of simplicity, we slightly abuse the notation by omitting subdomain subscripts and dimension superscripts in the following definitions. We only keep the indices in certain cases when clarification is necessary.

Assume that the boundary of Ω can be partitioned to $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ such that $\partial\Omega_D \cap \partial\Omega_N = \varnothing$ and $\partial\Omega_D$ is of positive measure. We adopt the notation in each dimension $0 < d \leqslant n$, that is

$$\partial\Omega_{iD}^d = \partial\Omega_i^d \cap \partial\Omega_D, \qquad \partial\Omega_{iN}^d = \partial\Omega_i^d \cap \partial\Omega_N. \tag{2.4}$$

The material permeability K and normal permeability $K_{\boldsymbol{\nu}}$ tensors are considered to be bounded both above and below, symmetric and positive definite, and we denote $\boldsymbol{\nu}$ as an outward unit normal on Ω. Furthermore, let γ_{ij}^d be the distance from Γ_{ij}^d to Ω_i^d, which for $d = n-1$ represents the fracture aperture. The physical parameters K and γ may vary spatially. However, to simplify the analysis, we assume that they are constant on each subdomain in each dimension.

In each Ω^d, we introduce the governing Darcy's law and mass conservation, find fluid velocity $\boldsymbol{u}^d$ and pressure p^d that satisfy

$$\boldsymbol{u}^d = -K\nabla p^d, \qquad \text{in } \Omega^d, \quad 0 \leqslant d \leqslant n, \tag{2.5a}$$

$$\nabla \cdot \boldsymbol{u}^d + [\![\lambda^d]\!] = f^d, \qquad \text{in } \Omega^d, \quad 0 \leqslant d \leqslant n, \tag{2.5b}$$

where we introduce an additional mortar variable λ^d, defined as

$$\lambda^d|_{\Gamma_{ij}^d} = \lambda_{ij}^d = \boldsymbol{u}^d \cdot \boldsymbol{\nu}, \qquad \text{on } \Gamma_{ij}^d, \qquad j \in J_i^d,\ i \in I^d,\ 0 \leqslant d \leqslant n-1, \tag{2.6}$$

to account for the mass transfer across each interface Γ_{ij}^d, and a jump operator $[\![\cdot]\!] : L^2(\Gamma^d) \to L^2(\Omega^d)$ as

$$[\![\lambda^d]\!]|_{\Omega_i^d} = -\sum_{j \in J_i^d} \lambda_{ij}^d, \qquad i \in I^d, \quad 0 \leqslant d \leqslant n. \tag{2.7}$$

Since there is no notion of interface Γ^n or flow in a point Ω^0, we extend the definion of λ^n and $\boldsymbol{u}^0$ by setting them equal to zero.

An additional interface law on Γ_{ij}^d is introduced to describe the normal flow due to the difference in pressure from Ω_i^d to Ω^{d+1},

$$\lambda_{ij}^d = -K_{\boldsymbol{\nu}} \frac{p_i^d - p^{d+1}|_{\Gamma_{ij}^d}}{\gamma_{ij}^d}, \quad \text{on } \Gamma_{ij}^d, \qquad j \in J_i^d,\ i \in I^d,\ 0 \leqslant d \leqslant n-1. \tag{2.8}$$

Finally, proper boundary conditions are needed. For example,

$$p^d = g^d, \qquad \text{on } \partial\Omega_D^d, \quad 0 \leqslant d \leqslant n, \tag{2.9}$$

$$\boldsymbol{u}^d \cdot \boldsymbol{\nu} = 0, \qquad \text{on } \partial\Omega_N^d, \quad 0 \leqslant d \leqslant n. \tag{2.10}$$

Remark 2.2 . *In the previous equations, we have used $\boldsymbol{u}^d$ as integrated flux and p^d as averaged pressure in each Ω^d, $0 \leqslant d \leqslant n$. Therefore, the scaling with the cross-sectional area ε of order $\mathcal{O}(\gamma^{n-d})$ due to the model reduction has been accounted for within the permeability parameters K and $K_{\boldsymbol{\nu}}$.*

2.1 Variational formulation

Now we consider the variational form of the problem (2.5)–(2.10). For any open bounded set $\omega \in \mathbb{R}^n$, let $L^2(\omega)$ and $H^s(\omega)$ denote the L^2 space and the standard Sobolev spaces on functions defined on ω, respectively. Also, denote $H^{\frac{1}{2}}(\partial\omega)$ as the space of L^2-traces on the boundary $\partial\omega$ of functions in $H^1(\omega)$. Let $(\cdot,\cdot)_\omega$ be the L^2-inner product and $\|\cdot\|_{L^2(\omega)}$ the induced L^2-norm. We define

$$\begin{aligned}
\boldsymbol{V}^d &= \{\boldsymbol{v} \in (L^2(\Omega^d))^d : \nabla \cdot \boldsymbol{v} \in L^2(\Omega^d), (\boldsymbol{v} \cdot \boldsymbol{\nu})|_{\partial\Omega_N^d} = 0\}, && 1 \leqslant d \leqslant n,\\
\Lambda^d &= L^2(\Gamma^d), && 0 \leqslant d \leqslant n-1,\\
Q^d &= L^2(\Omega^d), && 0 \leqslant d \leqslant n,
\end{aligned}$$

where $\boldsymbol{V}^d$ representing the flux function space on Ω^d, Q^d the pressure space on Ω^d, and Λ^d the function space of normal flux across interface Γ^d. Furthermore, let $\boldsymbol{V}_0^d$ be a subspace of $\boldsymbol{V}^d$ containing functions $\boldsymbol{v}_0$ such that $\boldsymbol{v}_0 \cdot \boldsymbol{\nu} = 0$ on Γ^{d-1}. In addition, define the extension operator $R^d : \Lambda^d \to \boldsymbol{V}^{d+1}$ as

$$R^d \lambda^d \cdot \boldsymbol{\nu} = \begin{cases} \lambda^d, & \text{on } \Gamma^d, \\ 0, & \text{elsewhere.} \end{cases} \tag{2.11}$$

To summarize the formulation, we compose function spaces over dimensions

$$\boldsymbol{V} = \bigoplus_{d=1}^{n} \boldsymbol{V}^d, \quad \boldsymbol{V}_0 = \bigoplus_{d=1}^{n} \boldsymbol{V}_0^{\boldsymbol{d}}, \quad \Lambda = \bigoplus_{d=0}^{n-1} \Lambda^d, \quad Q = \bigoplus_{d=0}^{n} Q^d, \tag{2.12}$$

and associate composite L^2-inner products

$$(\cdot,\cdot)_\Omega = \sum_{d=0}^{n} (\cdot,\cdot)_{\Omega^d} = \sum_{d=0}^{n} \sum_{i\in I^d} (\cdot,\cdot)_{\Omega_i^d}, \quad (\cdot,\cdot)_\Gamma = \sum_{d=0}^{n-1} (\cdot,\cdot)_{\Gamma^d} = \sum_{d=0}^{n-1} \sum_{i\in I^d} \sum_{j\in J_i^d} (\cdot,\cdot)_{\Gamma_{ij}^d}.$$

and induced composite L^2-norms

$$\|\cdot\|^2_{L^2(\Omega)} = \sum_{d=0}^{n} \|\cdot\|^2_{L^2(\Omega^d)}, \quad \|\cdot\|^2_{L^2(\Gamma)} = \sum_{d=0}^{n} \|\cdot\|^2_{L^2(\Gamma^d)}$$

Finally, let $R : \Lambda \to \boldsymbol{V}$ be defined as $R = \bigoplus_{d=0}^{n-1} R^d$.

The system (2.5)–(2.10) in the weak formulation reads: Find $(\boldsymbol{u}_0, \lambda, p) \in \boldsymbol{V}_0 \times \Lambda \times Q$ that satisfies

$$\left(K^{-1}(\boldsymbol{u}_0 + R\lambda), \boldsymbol{v}_0\right)_\Omega - (p, \nabla\cdot\boldsymbol{v}_0)_\Omega = -(g, \boldsymbol{v}_0\cdot\boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{v}_0 \in \boldsymbol{V}_0, \tag{2.13a}$$

$$\left(K^{-1}(\boldsymbol{u}_0 + R\lambda), R\mu\right)_\Omega - (p, \nabla\cdot R\mu)_\Omega + \left(\gamma K_{\boldsymbol{\nu}}^{-1}\lambda, \mu\right)_\Gamma - (p, [\![\lambda]\!])_\Omega = 0, \qquad \forall\, \mu \in \Lambda, \tag{2.13b}$$

$$-(\nabla\cdot(\boldsymbol{u}_0 + R\lambda), q)_\Omega - ([\![\lambda]\!], q)_\Omega = -(f, q)_\Omega, \qquad \forall\, q \in Q, \tag{2.13c}$$

with $g \in H^{\frac{1}{2}}(\partial\Omega_D)$ and $f \in L^2(\Omega)$. As before, functions $\boldsymbol{u}_0^0, \boldsymbol{v}_0^0, \lambda^n$ and μ^n are set to zero.

We end this section by observing the saddle point structure of the system (2.13). First, let $\boldsymbol{W} = \boldsymbol{V}_0 \times \Lambda$ be the function space of all flux variables, including mortar variable, and define the mixed-dimensional divergence operator $\boldsymbol{D}\cdot : \boldsymbol{W} \to Q$ as

$$\boldsymbol{D}\cdot\boldsymbol{w} = \boldsymbol{D}\cdot[\boldsymbol{u}_0, \lambda] = \nabla\cdot\boldsymbol{u}_0 + [\![\lambda]\!], \qquad \boldsymbol{w} \in \boldsymbol{W}. \tag{2.14}$$

Define the following two bilinear forms

$$a(\boldsymbol{w}, \boldsymbol{r}) = \left(K^{-1}(\boldsymbol{u}_0 + R\lambda), \boldsymbol{v}_0 + R\mu\right)_\Omega + \left(\gamma K_{\boldsymbol{\nu}}^{-1}\lambda, \mu\right)_\Gamma, \tag{2.15a}$$

$$b(\boldsymbol{r}, p) = -(p, \boldsymbol{D}\cdot[\boldsymbol{v}_0 + R\mu, \mu])_\Omega. \tag{2.15b}$$

Then the saddle point form of system (2.13) reads: Find $(\boldsymbol{w}, p) \in \boldsymbol{W} \times Q$ such that

$$a(\boldsymbol{w}, \boldsymbol{r}) + b(\boldsymbol{r}, p) = -(g, \boldsymbol{v}_0\cdot\boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{r} \in \boldsymbol{W}, \tag{2.16a}$$

$$b(\boldsymbol{w}, q) = -(f, q)_\Omega, \qquad \forall\, q \in Q. \tag{2.16b}$$

It has been shown in [9] that the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ are continuous with respect to the following norms for $\boldsymbol{r} = [\boldsymbol{v}_0, \mu] \in \boldsymbol{W}$ and $q \in Q$,

$$\begin{aligned}\|\boldsymbol{r}\|_{\boldsymbol{W}}^2 &= \|K^{-\frac{1}{2}}(\boldsymbol{v}_0 + R\mu)\|_{L^2(\Omega)}^2 + \|\gamma^{\frac{1}{2}} K_{\boldsymbol{\nu}}^{-\frac{1}{2}} \mu\|_{L^2(\Gamma)}^2 \\ &\quad + \|\boldsymbol{D} \cdot [\boldsymbol{v}_0 + R\mu, \mu]\|_{L^2(\Omega)}^2, \qquad (2.17a)\\ \|q\|_Q^2 &= \|q\|_{L^2(\Omega)}^2. \qquad (2.17b)\end{aligned}$$

In addition, $a(\cdot,\cdot)$ is shown to be coercive on the kernel of $b(\cdot,\cdot)$ in [9] as well. Finally, the following theorem states that $b(\cdot,\cdot)$ satisfies the inf-sup condition.

Theorem 2.1 [9]. *Let the bilinear form $b(\cdot,\cdot)$ be defined as in (2.15b). Then there exists a constant $\beta > 0$ independent of the physical parameters K, $K_{\boldsymbol{\nu}}$ and γ such that*

$$\inf_{q \in Q} \sup_{\boldsymbol{r} \in \boldsymbol{W}} \frac{b(\boldsymbol{r}, q)}{\|\boldsymbol{r}\|_{\boldsymbol{W}} \|q\|_Q} \geqslant \beta. \qquad (2.18)$$

Following the classical Brezzi theory [6, 11], we conclude that the saddle point system (2.16) is well-posed, i.e., there exists a unique solution of (2.16).

2.2 Discretization

We continue this section with discretizing the problem (2.16) by the mixed finite element approximation. Let $\mathcal{T}_\Omega^d$ and $\mathcal{T}_\Gamma^d$ denote a d-dimensional shape-regular triangulation of Ω^d and Γ^d, and $h = \max_{0 \leqslant d \leqslant n} h^d$ the characteristic mesh size parameter. Consider $\boldsymbol{V}_h^d \subset \boldsymbol{V}^d$, $\boldsymbol{V}_{0h}^d \subset \boldsymbol{V}_0^d$, $Q_h^d \subset Q^d$ and $\Lambda_h^d \subset \Lambda^d$ to be the lowest-order stable mixed finite element approximations on subdomain mesh $\mathcal{T}_\Omega^d$ and mortar mesh $\mathcal{T}_\Gamma^d$. That is, $\boldsymbol{V}_h^d = \mathbb{RT}_0(\mathcal{T}_\Omega^d)$, $\Lambda_h^d = \mathbb{P}_0(\mathcal{T}_\Gamma^d)$ and $Q_h^d = \mathbb{P}_0(\mathcal{T}_\Omega^d)$, where $\mathbb{RT}_0$ stands for lowest-order Raviart-Thomas(-Nédélec) spaces [27, 29] and $\mathbb{P}_0$ for the space of piecewise constants. Furthermore, define $\widehat{\Pi}_h^d : \Lambda_h^d \to \boldsymbol{V}_h^{d+1} \cdot \boldsymbol{\nu}|_{\Gamma^d}$ to be the L^2-projection operator such that, for any $\mu_h^d \in \Lambda_h^d$,

$$(\widehat{\Pi}_h^d \mu_h^d - \mu_h^d, \boldsymbol{v} \cdot \boldsymbol{\nu})_{L^2(\Gamma^d)} = 0, \qquad \forall \boldsymbol{v} \in \boldsymbol{V}_h^{d+1}. \qquad (2.19)$$

Then we can define the discrete extension operator $R_h^d : \Lambda_h^d \to \boldsymbol{V}_h^d$,

$$R_h^d \lambda^d \cdot \boldsymbol{\nu} = \begin{cases} \widehat{\Pi}_h^d \lambda^d, & \text{on } \Gamma^d, \\ 0, & \text{elsewhere.} \end{cases} \qquad (2.20)$$

Analogous to the continuous case, we define the discrete composite spaces

$$\boldsymbol{V}_h = \bigoplus_{d=1}^{n} \boldsymbol{V}_h^d, \quad \boldsymbol{V}_{0h} = \bigoplus_{d=1}^{n} \boldsymbol{V}_{0h}^d, \quad \Lambda_h = \bigoplus_{d=0}^{n-1} \Lambda_h^d, \quad Q_h = \bigoplus_{d=0}^{n} Q_h^d, \qquad (2.21)$$

and the linear operators $\widehat{\Pi}_h = \bigoplus_{d=0}^{n-1} \widehat{\Pi}_h^d$ and $R_h = \bigoplus_{d=0}^{n-1} R_h^d$.

With $\boldsymbol{W}_h = \boldsymbol{V}_{0h} \times \Lambda_h$, the finite element approximation of the system (2.13) is formulated as follows: Find $(\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$ such that,

$$a(\boldsymbol{w}_h, \boldsymbol{r}_h) + b(\boldsymbol{r}_h, p_h) = -(g, \boldsymbol{v}_{0h} \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{r}_h \in \boldsymbol{W}_h, \tag{2.22a}$$

$$b(\boldsymbol{w}_h, q_h) = -(f, q_h)_\Omega, \qquad \forall\, q_h \in Q_h. \tag{2.22b}$$

Due to our choice of the finite element spaces, the continuity of $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ and the coercivity of $a(\cdot,\cdot)$ on the kernel of $b(\cdot,\cdot)$ are preserved naturally. To show the well-posedness of the discrete saddle point system (2.22), we need the inf-sup condition to hold on the discrete spaces as well. This has been shown in [9] and is stated in the following theorem.

Theorem 2.2 [9]. *There exists a constant $\beta > 0$ independent of the discretization parameter h and the physical parameters K, $K_{\boldsymbol{\nu}}$ and γ such that*

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{r}_h \in \boldsymbol{W}_h} \frac{b(\boldsymbol{r}_h, q_h)}{\|\boldsymbol{r}_h\|_{\boldsymbol{W}_h} \|q_h\|_{Q_h}} \geqslant \beta. \tag{2.23}$$

Therefore, the finite element method (2.22) is well-posed by the Brezzi theory [6, 11].

We finalize this section with the block formulation of the discrete saddle point system (2.22). Let linear operators $A : \boldsymbol{W}_h \to \boldsymbol{W}_h'$ and $B : \boldsymbol{W}_h \to Q_h'$ be defined as $\langle A\boldsymbol{w}_h, \boldsymbol{r}_h\rangle = a(\boldsymbol{w}_h, \boldsymbol{r}_h)$ and $\langle B\boldsymbol{r}_h, p_h\rangle = b(\boldsymbol{r}_h, p_h)$, respectively. Here $\boldsymbol{W}_h'$ and Q_h' denote the dual spaces of $\boldsymbol{W}_h$ and Q_h, respectively, and $\langle\cdot,\cdot\rangle$ denotes the duality pairing. Then (2.22) is equivalent to the following operator form,

$$\mathcal{A}\begin{pmatrix} \boldsymbol{w}_h \\ p_h \end{pmatrix} = \begin{pmatrix} G \\ F \end{pmatrix} \quad \text{with } \mathcal{A} = \begin{pmatrix} A & B^T \\ -B & 0 \end{pmatrix}, \tag{2.24}$$

with $G([\boldsymbol{v}_{0h}, \lambda_h]) := -(g, \boldsymbol{v}_{0h} \cdot \boldsymbol{\nu})_{\partial\Omega_D}$ and $F(q_h) := (f, q_h)_\Omega$.

The well-posedness of the system (2.22) ensures that $\mathcal{A}$ is an isomorphism from $\boldsymbol{W}_h \times Q_h$ to its dual $\boldsymbol{W}_h' \times Q_h'$ and, therefore, (2.24) has a unique solution $(\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$.

3 Block preconditioners

In this section, we briefly present the general preconditioning theory for designing block preconditioners of Krylov subspace iterative methods [25, 26], which introduces necessary tools for the analysis in the following section.

The block preconditioning framework [25,26] is based on the well-posedness theory. Therefore, we first introduce the setup of the problem. Let $\boldsymbol{X}$ be a real separable Hilbert space and $(\cdot,\cdot)_{\boldsymbol{X}}$ represent the inner product on $\boldsymbol{X}$ that induces the norm $\|\cdot\|_{\boldsymbol{X}}$. Furthermore, denote $\boldsymbol{X}'$ as a dual space to $\boldsymbol{X}$ and $\langle\cdot,\cdot\rangle$ as the duality pairing between them. Let $\mathcal{L}(\cdot,\cdot)$ be a bilinear form on $\boldsymbol{X}$ that satisfies the continuity condition and the inf-sup condition,

$$\inf_{\boldsymbol{x}\in\boldsymbol{X}} \sup_{\boldsymbol{y}\in\boldsymbol{X}} \frac{\mathcal{L}(\boldsymbol{x},\boldsymbol{y})}{\|\boldsymbol{x}\|_{\boldsymbol{X}}\|\boldsymbol{y}\|_{\boldsymbol{X}}} \geqslant \beta \quad \text{and} \quad |\mathcal{L}(\boldsymbol{x},\boldsymbol{y})| \leqslant \alpha\|\boldsymbol{x}\|_{\boldsymbol{X}}\|\boldsymbol{y}\|_{\boldsymbol{X}},\ \forall \boldsymbol{x},\boldsymbol{y} \in \boldsymbol{X}, \tag{3.1}$$

for $\alpha, \beta > 0$. We aim to construct a robust preconditioner for the linear system

$$\mathcal{A}\boldsymbol{x} = \boldsymbol{b}, \tag{3.2}$$

where $\mathcal{A} : \boldsymbol{X} \to \boldsymbol{X}'$ is induced by the bilinear form $\mathcal{L}(\cdot, \cdot)$ such that $\langle \mathcal{A}\boldsymbol{x}, \boldsymbol{y} \rangle = \mathcal{L}(\boldsymbol{x}, \boldsymbol{y})$. The properties of the bilinear form ensure that $\mathcal{A}$ is a bounded and symmetric linear operator and the system (3.2) is well-posed. Our goal is to develop block preconditioners for solving (3.2).

3.1 Norm-equivalent Preconditioner

Consider a symmetric positive definite operator $\mathcal{M} : \boldsymbol{X}' \to \boldsymbol{X}$ which induces an inner product $(\boldsymbol{x}, \boldsymbol{y})_{\mathcal{M}^{-1}} := \langle \mathcal{M}^{-1}\boldsymbol{x}, \boldsymbol{y} \rangle$ on $\boldsymbol{X}$ and corresponding norm $\|\boldsymbol{x}\|^2_{\mathcal{M}^{-1}} := (\boldsymbol{x}, \boldsymbol{x})_{\mathcal{M}^{-1}}$. Naturally, $\mathcal{M}\mathcal{A} : \boldsymbol{X} \to \boldsymbol{X}$ is symmetric with respect to $(\cdot, \cdot)_{\mathcal{M}^{-1}}$ and we can use $\mathcal{M}$ as a preconditioner for the MINRES algorithm whose convergence rate is stated in the following theorem.

Theorem 3.1 [20]. *Let $\boldsymbol{x}^m$ be the m-th iteration of the MINRES method preconditioned with $\mathcal{M}$ and $\boldsymbol{x}$ be the exact solution, it follows that*

$$\|\mathcal{A}(\boldsymbol{x} - \boldsymbol{x}^m)\|_{\mathcal{M}} \leqslant 2\rho^m \|\mathcal{A}(\boldsymbol{x} - \boldsymbol{x}^0)\|_{\mathcal{M}},$$

where $\rho = \frac{\kappa(\mathcal{M}\mathcal{A})-1}{\kappa(\mathcal{M}\mathcal{A})+1}$ and $\kappa(\mathcal{M}\mathcal{A})$ denotes the condition number of $\mathcal{M}\mathcal{A}$.

As shown in [26], if (3.1) holds and $\mathcal{M}$ satisfies,

$$c_1 \|\boldsymbol{x}\|^2_{\boldsymbol{X}} \leqslant \|\boldsymbol{x}\|^2_{\mathcal{M}^{-1}} \leqslant c_2 \|\boldsymbol{x}\|^2_{\boldsymbol{X}}, \tag{3.3}$$

then $\mathcal{A}$ and $\mathcal{M}$ are called *norm-equivalent* and $\kappa(\mathcal{M}\mathcal{A}) \leqslant \frac{c_2\alpha}{c_1\beta}$. Thus, if the well-posedness constants α and β and the norm-equivalence constants c_1 and c_2 are all independent of the physical and discretization parameters, then $\mathcal{M}$ provides a robust preconditioner.

One natural choice of the norm-equivalent preconditioner is the *Riesz operator* $\mathcal{B} : \boldsymbol{X}' \to \boldsymbol{X}$ corresponding to the inner product $(\cdot, \cdot)_{\boldsymbol{X}}$

$$(\mathcal{B}\boldsymbol{f}, \boldsymbol{x})_{\boldsymbol{X}} = \langle \boldsymbol{f}, \boldsymbol{x} \rangle, \qquad \forall \boldsymbol{f} \in \boldsymbol{X}',\ \boldsymbol{x} \in \boldsymbol{X}. \tag{3.4}$$

It is easy to see that if we choose $\mathcal{M} = \mathcal{B}$, then (3.3) holds with constants $c_1 = c_2 = 1$ and, therefore, the preconditioned system

$$\mathcal{B}\mathcal{A}\boldsymbol{x} = \mathcal{B}\boldsymbol{b} \tag{3.5}$$

has a bounded condition number

$$\kappa(\mathcal{B}\mathcal{A}) = \|\mathcal{B}\mathcal{A}\|_{\mathscr{L}(\boldsymbol{X},\boldsymbol{X})} \|(\mathcal{B}\mathcal{A})^{-1}\|_{\mathscr{L}(\boldsymbol{X},\boldsymbol{X})} \leqslant \frac{\alpha}{\beta}. \tag{3.6}$$

If the constants α and β are independent of the discretization and physical parameters, we obtain a robust preconditioner.

3.2 Field-of-values-equivalent Preconditioner

In this section, we recall the class of field-of-values-equivalent (FOV-equivalent) preconditioners which allow more general preconditioners than the norm-equivalent ones.

Consider a general operator $\mathcal{M}_L : \boldsymbol{X}' \to \boldsymbol{X}$ which can be used as a preconditioner for the GMRES method. The following theorem, developed in [13,14], characterizes the convergence rate of the GMRES method.

Theorem 3.2 [13,14]. *Let $\boldsymbol{x}^m$ be the m-th iteration of the GMRES method preconditioner with $\mathcal{M}_L$ and $\boldsymbol{x}$ be the exact solution, it follows that*

$$\|\mathcal{M}_L\mathcal{A}(\boldsymbol{x}-\boldsymbol{x}^m)\|^2_{\mathcal{M}^{-1}} \leqslant \left(1-\frac{\Sigma^2}{\Upsilon^2}\right)\|\mathcal{M}_L\mathcal{A}(\boldsymbol{x}-\boldsymbol{x}^0)\|^2_{\mathcal{M}^{-1}},$$

where, for any $\boldsymbol{x} \in \boldsymbol{X}$,

$$\Sigma \leqslant \frac{(\mathcal{M}_L\mathcal{A}\boldsymbol{x},\boldsymbol{x})_{\mathcal{M}^{-1}}}{(\boldsymbol{x},\boldsymbol{x})_{\mathcal{M}^{-1}}}, \qquad \frac{\|\mathcal{M}_L\mathcal{A}\boldsymbol{x}\|_{\mathcal{M}^{-1}}}{\|\boldsymbol{x}\|_{\mathcal{M}^{-1}}} \leqslant \Upsilon.$$

$\mathcal{M}_L$ is referred to as an FOV-equivalent preconditioner if the constants Σ and Υ are independent of the physical and discretization parameters. Usually $\mathcal{M}_L$ provides a uniform left preconditioner for GMRES.

In a similar manner, we can introduce a right preconditioner for GMRES, $\mathcal{M}_U : \boldsymbol{X}' \to \boldsymbol{X}$ and consider the preconditioned system

$$\mathcal{A}\mathcal{M}_U\boldsymbol{y} = \boldsymbol{b}, \qquad \boldsymbol{x} = \mathcal{M}_U\boldsymbol{y}.$$

By introducing an inner product on $\boldsymbol{X}'$, defined as $(\boldsymbol{x}',\boldsymbol{y}')_{\mathcal{M}} := \langle \boldsymbol{x}', \mathcal{M}\boldsymbol{y}'\rangle$, we say $\mathcal{M}_U$ and $\mathcal{A}$ are FOV-equivalent if, for any $\boldsymbol{x}' \in \boldsymbol{X}'$,

$$\Sigma \leqslant \frac{(\mathcal{A}\mathcal{M}_U\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{M}}}{(\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{M}}}, \qquad \frac{\|\mathcal{A}\mathcal{M}_U\boldsymbol{x}'\|_{\mathcal{M}}}{\|\boldsymbol{x}'\|_{\mathcal{M}}} \leqslant \Upsilon,$$

where the constants Σ and Υ are independent of the physical and discretization parameters. Therefore, $\mathcal{M}_U$ can be used as a uniform right preconditioner for GMRES.

In many cases [1,2,25], the FOV-equivalent preconditioners can be derived based on the Riesz operator and the FOV-equivalence can be shown based on the well-posedness conditions (3.1).

4 Robust Preconditioners for Mixed-dimensional Model

In this section, we design block preconditioners based on the general framework mentioned in the previous section. Consider the finite element approximation (2.22). In this case, define $\boldsymbol{X} = \boldsymbol{W}_h \times Q_h$ associated with the following norm

$$\|\boldsymbol{y}\|^2_{\boldsymbol{X}} = \|(\boldsymbol{r}_h, q_h)\|^2_{\boldsymbol{X}} = \|\boldsymbol{r}_h\|^2_{\boldsymbol{W}} + \|q_h\|^2_Q. \tag{4.1}$$

Then, the operator $\mathcal{A} : \boldsymbol{X} \to \boldsymbol{X}'$ in (2.24) is induced by the bilinear form

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{y}) = a(\boldsymbol{w}_h, \boldsymbol{r}_h) + b(\boldsymbol{r}_h, p_h) - b(\boldsymbol{w}_h, q_h), \tag{4.2}$$

and satisfies the well-posedness conditions (3.1) due to Theorem 2.2, the continuity of the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$, and the coercivity of $a(\cdot,\cdot)$ on the kernal of $b(\cdot,\cdot)$. Moreover, the constants α and β are independent of parameters h, K, $K_{\boldsymbol{\nu}}$ and γ.

The Riesz operator corresponding to the norm $\|\cdot\|_{\boldsymbol{X}}$ in (4.1) is

$$\mathcal{B} = \begin{pmatrix} A + B^T B & 0 \\ 0 & I_p \end{pmatrix}^{-1}, \tag{4.3}$$

where A and B are defined as in (2.24) and I_p is the identity operator on Q, i.e., $\langle I_p q_h, q_h \rangle = \|q_h\|_Q^2$. The main challenge in implementation of this preconditioner is to solve for the upper block $A + B^T B$ that corresponds to $I + \text{grad div}$ problem. One way of resolving this is to use auxiliary space theory (see for example [21,24]). However, in our case, additional theory resulting from the mixed-dimensional exterior calculus in [8] is needed, which is the topic of our ongoing work [7]. However, in this paper, we consider an alternative formulation of the problem (2.22) and show the well-posedness with respect to a different weighted norm, which allows for a simpler robust preconditioner.

4.1 An Alternative Formulation

In order to introduce the alternative formulation, we need to define a discrete gradient operator $\boldsymbol{D}_h : Q_h \to \boldsymbol{W}_h$ such that, for $\boldsymbol{r}_h = [\boldsymbol{v}_{0h}, \mu_h]$,

$$a^D(\boldsymbol{D}_h p_h, \boldsymbol{r}_h) = b(\boldsymbol{r}_h, p_h) = -\left(p_h, \boldsymbol{D} \cdot [\boldsymbol{v}_{0h} + R_h \mu_h, \mu_h]\right)_\Omega, \tag{4.4}$$

where, for $\boldsymbol{w}_h = [\boldsymbol{u}_{0h}, \lambda_h]$ and $\boldsymbol{r}_h = [\boldsymbol{v}_{0h}, \mu_h]$,

$$a^D(\boldsymbol{w}_h, \boldsymbol{r}_h) := (K^{-1}(\boldsymbol{u}_{0h} + R_h \lambda_h), \boldsymbol{v}_{0h} + R_h \mu_h)_{D,\Omega} + (\gamma K_{\boldsymbol{\nu}}^{-1} \lambda_h, \mu_h)_\Gamma,$$

with

$$(K^{-1}(\boldsymbol{u}_{0h} + R_h \lambda_h), \boldsymbol{v}_{0h} + R_h \mu_h)_{D,\Omega} :=$$
$$\sum_{d=0}^{n} \left\{ \sum_{T^d \in \mathcal{T}_\Omega^d} \left[\sum_{f^d \in \partial T^d} ((\boldsymbol{u}_{0h} + R_h \lambda_h) \cdot \boldsymbol{\nu}_{f^d})((\boldsymbol{v}_{0h} + R_h \mu_h) \cdot \boldsymbol{\nu}_{f^d})(K^{-1} \boldsymbol{\phi}_{f^d}, \boldsymbol{\phi}_{f^d})_{T^d} \right] \right\}.$$

Here $T^d \in \mathcal{T}_\Omega^d$ is either a tetrahedron for $d = 3$, a triangle for $d = 2$ or a line segment for $d = 1$. Furthermore, $f^d \in \partial T$ corresponds to a face of the element T^d, $\boldsymbol{\nu}_{f^d}$ denotes the unit outer normal of face f^d, and $\boldsymbol{\phi}_{f^d} \in \mathbb{RT}_0(T^d)$ is the basis function on face f^d. Using the discrete gradient operator, an alternative

formulation of the system (2.22) is given as follows: Find $(\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$ such that,

$$a(\boldsymbol{w}_h, \boldsymbol{r}_h) + a^D(\boldsymbol{D}_h p_h, \boldsymbol{r}_h) = -(g, \boldsymbol{v}_{0h} \cdot \boldsymbol{\nu})_{\partial\Omega_D}, \qquad \forall\, \boldsymbol{r}_h \in \boldsymbol{W}_h, \tag{4.5a}$$

$$a^D(\boldsymbol{D}_h q_h, \boldsymbol{w}_h) = -(f, q_h)_\Omega, \qquad \forall\, q_h \in Q_h. \tag{4.5b}$$

The well-posedness of the alternative formulation (4.5) with respect to the norm (4.1) follows directly from the well-posedness of the original formulation (2.22) because the two formulations are equivalent. However, in order to derive a block preconditioner different from (4.3), we shall consider the same coefficient operator $\mathcal{A}$ (2.24) with a different weak interpretation and the well-posedness in a different setting.

The alternative weighted norm we consider for the alternative formulation (4.5) is defined as

$$|||(\boldsymbol{r}_h, q_h)|||^2 := \|\boldsymbol{r}_h\|_a^2 + \|\boldsymbol{D}_h q_h\|_{a^D}^2, \tag{4.6}$$

where $\|\boldsymbol{r}_h\|_a^2 := a(\boldsymbol{r}_h, \boldsymbol{r}_h)$ and $\|\boldsymbol{r}_h\|_{a^D}^2 := a^D(\boldsymbol{r}_h, \boldsymbol{r}_h)$. In order to show (4.5) (or the operator form (2.24)) is well-posed with respect to this alternative norm (4.6), we need the following two lemmas. The first lemma shows that the forms $a(\cdot,\cdot)$ and $a^D(\cdot,\cdot)$ are spectrally equivalent.

Lemma 4.1 . *There exist constants $c_1, c_2 > 0$, depending only on the shape regularity of the mesh $\mathcal{T}_\Omega$, such that the following inequalities hold,*

$$c_1 \|\boldsymbol{r}_h\|_{a^D} \leqslant \|\boldsymbol{r}_h\|_a \leqslant c_2 \|\boldsymbol{r}_h\|_{a^D}, \qquad \forall\ \boldsymbol{r}_h \in \boldsymbol{W}_h. \tag{4.7}$$

Proof. Recall that

$$\begin{aligned}
\|\boldsymbol{r}_h\|_a^2 &= a([\boldsymbol{v}_{0h}, \mu_h], [\boldsymbol{v}_{0h}, \mu_h]) \\
&= (K^{-1}(\boldsymbol{v}_{0h} + R_h\mu_h), (\boldsymbol{v}_{0h} + R_h\mu_h))_\Omega + (\gamma K_{\boldsymbol{\nu}}^{-1}\mu_h, \mu_h)_\Gamma, \\
\|\boldsymbol{r}_h\|_{a^D}^2 &= a^D([\boldsymbol{v}_{0h}, \mu_h], [\boldsymbol{v}_{0h}, \mu_h]) \\
&= (K^{-1}(\boldsymbol{v}_{0h} + R_h\mu_h), (\boldsymbol{v}_{0h} + R_h\mu_h))_{D,\Omega} + (\gamma K_{\boldsymbol{\nu}}^{-1}\mu_h, \mu_h)_\Gamma.
\end{aligned}$$

Obviously, (4.7) holds if $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h), (\boldsymbol{v}_{0h}+R_h\mu_h))_\Omega$ and $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h), (\boldsymbol{v}_{0h}+R_h\mu_h))_{D,\Omega}$ are spectrally equivalent. Note that

$$(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h), \boldsymbol{v}_{0h}+R_h\mu_h)_\Omega = \sum_{d=0}^{n} \sum_{T^d \in \mathcal{T}_\Omega^d} (K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h), \boldsymbol{v}_{0h}+R_h\mu_h)_{T^d},$$

where

$$\begin{aligned}
&(K^{-1}(\boldsymbol{v}_{0h} + R_h\mu_h), \boldsymbol{v}_{0h} + R_h\mu_h)_{T^d} = \\
&\sum_{f^d, \tilde{f}^d \in \partial T^d} ((\boldsymbol{v}_{0h} + R_h\mu_h) \cdot \boldsymbol{\nu}_{f^d})((\boldsymbol{v}_{0h} + R_h\mu_h) \cdot \boldsymbol{\nu}_{\tilde{f}^d})(K^{-1}\boldsymbol{\phi}_{f^d}, \boldsymbol{\phi}_{\tilde{f}^d})_{T^d}.
\end{aligned}$$

and

$$(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,\Omega}=\sum_{d=0}^{n}\sum_{T^d\in\mathcal{T}_\Omega^d}(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,T^d},$$

where

$$\begin{aligned}&(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,T^d}=\\&\sum_{f^d\in\partial T^d}((\boldsymbol{v}_{0h}+R_h\mu_h)\cdot\boldsymbol{\nu}_{f^d})((\boldsymbol{v}_{0h}+R_h\mu_h)\cdot\boldsymbol{\nu}_{f^d})(K^{-1}\boldsymbol{\phi}_{f^d},\boldsymbol{\phi}_{f^d})_{T^d}.\end{aligned}$$

Therefore, we can immediately observe that it is enough to show that $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{T^d}$ and $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,T^d}$ are spectrally equivalent on each element T^d, $0<d\leqslant n$. In addition, by using the scaling argument [10, Section 4.5.2], we only need to show they are spectrally equivalent on a reference element $\hat{T}^d$, i.e.,

$$\begin{aligned}&\tilde{c}_1(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,\hat{T}^d}\\&\qquad\leqslant(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{\hat{T}^d}\\&\qquad\leqslant\tilde{c}_2(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,\hat{T}^d}.\end{aligned}\tag{4.8}$$

We show the proof for $d=n=3$. For other cases the proof follows similarly.

For $d=n=3$, the reference element $\hat{T}^d$ is a tetrahedron with vertices $(0,0,0)$, $(1,0,0)$, $(0,1,0)$ and $(0,0,1)$ in the Cartesian coordinates. The local matrix $A_{\hat{T}^d}$, representing $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{\hat{T}^d}$, takes the following form

$$A_{\hat{T}^d}=\frac{K^{-1}}{120}\begin{pmatrix}18&\sqrt{3}&\sqrt{3}&\sqrt{3}\\\sqrt{3}&16&-4&-4\\\sqrt{3}&-4&16&-4\\\sqrt{3}&-4&-4&16\end{pmatrix},$$

By the definition, $(K^{-1}(\boldsymbol{v}_{0h}+R_h\mu_h),\boldsymbol{v}_{0h}+R_h\mu_h)_{D,\hat{T}^d}$ is represented by the diagonal of $A_{\hat{T}^d}$, which we denote as $D_{A_{\hat{T}^d}}=\frac{K^{-1}}{120}\mathrm{diag}(18,16,16,16)$. To show (4.8) on $\hat{T}^d$, it is enough to notice that, under our assumption that K is constant on each T^d, the generalized eigenvalue problem $A_{\hat{T}^d}\boldsymbol{y}=\chi D_{A_{\hat{T}^d}}\boldsymbol{y}$ gives all eigenvalues $\chi>0$ independent of physical and discretization parameters. Therefore, (4.8) holds with $\tilde{c}_1=\sqrt{\chi_{\min}}$ and $\tilde{c}_2=\sqrt{\chi_{\max}}$, where $\chi_{\min}$ and $\chi_{\max}$ denote the smallest and largest eigenvalue, respectively. The spectral equivalent result (4.7) follows directly by the scaling argument [10, Section 4.5.2] and summing over all $T^d\in\mathcal{T}_\Omega^d$, $0\leqslant d\leqslant n$. The constants c_1 and c_2 depend on the shape regularity of the mesh due to the scaling argument but do not depend on the physical and discretization parameters. □

Based on the spectral equivalence Lemma (4.1), we have the following inf-sup condition regarding the discrete gradient $\boldsymbol{D}_h$.

Lemma 4.2 . *Let the discrete gradient operator $\boldsymbol{D}_h$ be defined as in (4.4). Then there exists a constant $\beta_\star > 0$ independent of the discretization and physical parameters such that*

$$\inf_{q_h \in Q_h} \sup_{\boldsymbol{r}_h \in \boldsymbol{W}_h} \frac{a^D(\boldsymbol{D}_h q_h, \boldsymbol{r}_h)}{\|\boldsymbol{r}_h\|_a \|\boldsymbol{D}_h q_h\|_{a^D}} \geqslant \beta_\star. \tag{4.9}$$

Proof. Using Lemma 4.1, we have for any $q_h \in Q_h$

$$\begin{aligned}\sup_{\boldsymbol{r}_h \in \boldsymbol{W}_h} \frac{a^D(\boldsymbol{D}_h q_h, \boldsymbol{r}_h)}{\|\boldsymbol{r}_h\|_a} &\geqslant \sup_{\boldsymbol{r}_h \in \boldsymbol{W}_h} \frac{a^D(\boldsymbol{D}_h q_h, \boldsymbol{r}_h)}{c_2 \|\boldsymbol{r}_h\|_{a^D}} \\ &= c_2^{-1} \|\boldsymbol{D}_h q_h\|_{a^D}.\end{aligned}$$

Now the result follows taking infimum over all $q_h \in Q_h$ and $\beta_\star = c_2^{-1}$. □

Based on Lemma 4.1 and 4.2, by Babuska-Brezzi theory [6, 11], we can conclude that the alternative formulation (4.5) is well-posed with respect to the norm (4.6) as stated in the following theorem.

Theorem 4.3 . *Consider the composite bilinear form on the space $\boldsymbol{W}_h \times Q_h$,*

$$\mathcal{L}(\boldsymbol{w}_h, p_h; \boldsymbol{r}_h, q_h) := a(\boldsymbol{w}_h, \boldsymbol{r}_h) + a^D(\boldsymbol{D}_h p_h, \boldsymbol{r}_h) + a^D(\boldsymbol{D}_h q_h, \boldsymbol{w}_h).$$

It satisfies the continuity condition and the inf-sup condition with respect to $|||(\boldsymbol{r}_h, q_h)|||$, i.e., for any $(\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$ and $(\boldsymbol{r}_h, q_h) \in \boldsymbol{W}_h \times Q_h$,

$$|\mathcal{L}(\boldsymbol{w}_h, p_h; \boldsymbol{r}_h, q_h)| \leqslant \alpha |||(\boldsymbol{w}_h, p_h)||| |||(\boldsymbol{r}_h, q_h)|||,$$

$$\inf_{(\boldsymbol{r}_h, q_h) \in \boldsymbol{W}_h \times Q_h} \sup_{(\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h} \frac{\mathcal{L}(\boldsymbol{w}_h, p_h; \boldsymbol{r}_h, q_h)}{|||(\boldsymbol{w}_h, p_h)||| |||(\boldsymbol{r}_h, q_h)|||} \geqslant \beta,$$

with constants α and β dependent on the shape regularity of the mesh but independent of discretization and physical parameters.

4.2 Block diagonal preconditioners

The well-posedness Theorem 4.3 provides alternative block preconditioners for solving the linear system (2.24) effectively. To this end, we introduce a linear operators $D_A : \boldsymbol{W}_h \to \boldsymbol{W}_h'$ which is defined as $\langle D_A \boldsymbol{w}_h, \boldsymbol{r}_h \rangle = a^D(\boldsymbol{w}_h, \boldsymbol{r}_h)$ for $\boldsymbol{w}_h, \boldsymbol{r}_h \in \boldsymbol{W}_h$. The reason we use the notation D_A here is that, by the definitions of $a(\cdot,\cdot)$ and $a^D(\cdot,\cdot)$, the matrix representation of linear operator D_A is exactly the diagonal of the matrix representation of linear operator A. Then, by the definition of the discrete gradient operator $\boldsymbol{D}_h$ (4.4), we have $D_A \boldsymbol{D}_h = B^T$ and, therefore,

$$\|\boldsymbol{D}_h q_h\|_{a^D}^2 = \langle D_A \boldsymbol{D}_h q_h, \boldsymbol{D}_h q_h \rangle = \langle B^T q_h, D_A^{-1} B^T q_h \rangle = \langle B D_A^{-1} B^T q_h, q_h \rangle,$$

for $q_h \in Q_h$. Based on the above operator form of the $\|\cdot\|_{a^D}$ norm, the Riesz operator corresponding to the $|||\cdot|||$ norm (4.6) is

$$\mathcal{B}_D = \begin{pmatrix} A & 0 \\ 0 & BD_A^{-1}B^T \end{pmatrix}^{-1}. \tag{4.10}$$

As discussed in Section 3.1, $\mathcal{B}_D$ is a norm-equivalent preconditioner for solving the system (2.24) and we have the following theorem regarding the condition number of $\mathcal{B}_D\mathcal{A}$.

Theorem 4.4 . *Let $\mathcal{B}_D$ be as in (4.10). Then $\kappa(\mathcal{B}_D\mathcal{A}) \leqslant \frac{\alpha}{\beta}$.*

Remark 4.1 . *Notice that Theorem 4.3 (essentially Lemma 4.1) ensures that $\kappa(\mathcal{B}_D\mathcal{A})$ is bounded independently of h and parameters K, $K_{\boldsymbol{\nu}}$ and γ, but remains dependent on the shape regularity of the mesh.*

In practice, applying the preconditioner $\mathcal{B}_D$ implies inverting the diagonal block exactly, which can be expensive and sometimes infeasible. Thus, we consider the following preconditioner

$$\mathcal{M}_D = \begin{pmatrix} H_{\boldsymbol{w}} & 0 \\ 0 & H_p \end{pmatrix}, \tag{4.11}$$

where the diagonal blocks $H_{\boldsymbol{w}}$ and H_p are symmetric positive definite and spectrally equivalent to diagonal blocks in A and $BD_A^{-1}B^T$, respectively, i.e.

$$c_{1,\boldsymbol{w}}(H_{\boldsymbol{w}}\boldsymbol{r}_h, \boldsymbol{r}_h) \leqslant (A^{-1}\boldsymbol{r}_h, \boldsymbol{r}_h) \leqslant c_{2,\boldsymbol{w}}(H_{\boldsymbol{w}}\boldsymbol{r}_h, \boldsymbol{r}_h), \tag{4.12a}$$

$$c_{1,p}(H_p q_h, q_h) \leqslant ((BD_A^{-1}B^T)^{-1}q_h, q_h) \leqslant c_{2,p}(H_p q_h, q_h), \tag{4.12b}$$

where the constants $c_{1,\boldsymbol{w}}$, $c_{1,p}$, $c_{2,\boldsymbol{w}}$, and $c_{2,p}$ are independent of discretization and physical parameters. In practice, $H_{\boldsymbol{w}}$ can be defined by a diagonal scaling, i.e., D_A^{-1} and H_p can be defined by standard multigrid methods. In general, the choice of $H_{\boldsymbol{w}}$ and H_p are not very restrictive, provided it handles possible heterogeneity in physical parameters K, K_ν, and γ.

$\mathcal{M}_D$ is a norm-equivalent preconditioner as well. Following [26], we can directly estimate the condition number of $\mathcal{M}_D\mathcal{A}$ in the following theorem.

Theorem 4.5 . *Let $\mathcal{M}_\mathcal{D}$ be as in (4.11) and let (4.12) hold. Then it follows that $\kappa(\mathcal{M}_D\mathcal{A}) \leqslant \frac{\alpha c_2}{\beta c_1}$, where $c_2 = \max\{c_{2,\boldsymbol{w}}, c_{2,p}\}$ and $c_1 = \min\{c_{1,\boldsymbol{w}}, c_{1,p}\}$.*

Remark 4.2 . *Again, $\kappa(\mathcal{M}_D\mathcal{A})$ is bounded independently of h and parameters K, $K_{\boldsymbol{\nu}}$ and γ, but remains dependent on the shape regularity of the mesh.*

4.3 Block triangular preconditioners

In this subsection, we consider the block triangular preconditioners based on the FOV-equivalent preconditioners we discussed in Section 3.2. Here, we analyze the robustness of block triangular preconditioners and show the corresponding FOV-equivalence, which leads to uniform convergence rate of the GMRES method.

The block lower triangular preconditioners take the following form

$$\mathcal{B}_L = \begin{pmatrix} A & 0 \\ -B & BD_A^{-1}B^T \end{pmatrix}^{-1} \text{ and } \mathcal{M}_L = \begin{pmatrix} H_{\boldsymbol{w}}^{-1} & 0 \\ -B & H_p^{-1} \end{pmatrix}^{-1}. \tag{4.13}$$

On the other hand, the block upper triangular preconditioners are given as

$$\mathcal{B}_U = \begin{pmatrix} A & B^T \\ 0 & BD_A^{-1}B^T \end{pmatrix}^{-1} \text{ and } \mathcal{M}_U = \begin{pmatrix} H_{\boldsymbol{w}}^{-1} & B^T \\ 0 & H_p^{-1} \end{pmatrix}^{-1}. \tag{4.14}$$

Basically, $\mathcal{M}_L$ and $\mathcal{M}_U$ are inexact versions of $\mathcal{B}_L$ and $\mathcal{B}_U$ when the diagonal blocks are replaced by spectrally equivalent approximations (4.12).

Next theorem shows that $\mathcal{B}_L$ and $\mathcal{A}$ are FOV-equivalent.

Theorem 4.6 . *There exist constants $\xi_1, \xi_2 > 0$, independent of discretization and physical parameters, such that for every $\boldsymbol{x} = (\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$, $\boldsymbol{x} \neq \boldsymbol{0}$,*

$$\xi_1 \leqslant \frac{(\mathcal{B}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{x})_{\mathcal{B}_D^{-1}}}{(\boldsymbol{x}, \boldsymbol{x})_{\mathcal{B}_D^{-1}}}, \quad \textit{and} \quad \frac{\|\mathcal{B}_L\mathcal{A}\boldsymbol{x}\|_{\mathcal{B}_D^{-1}}}{\|\boldsymbol{x}\|_{\mathcal{B}_D^{-1}}} \leqslant \xi_2.$$

Proof. By the definition of the linear operators A and D_A, we naturally have $\|\boldsymbol{w}_h\|_a = \|\boldsymbol{w}_h\|_A$ and $\|\boldsymbol{w}_h\|_{a^D} = \|\boldsymbol{w}_h\|_{D_A}$, respectively. Here $\|\boldsymbol{w}_h\|_A^2 := \langle A\boldsymbol{w}_h, \boldsymbol{w}_h\rangle$ and $\|\boldsymbol{w}_h\|_{D_A}^2 := \langle D_A\boldsymbol{w}_h, \boldsymbol{w}_h\rangle$ for $\boldsymbol{w}_h \in \boldsymbol{W}_h$.

Then Lemma 4.1 states that the norms $\|\cdot\|_{D_A}$ and $\|\cdot\|_A$ are equivalent, which also implies the equivalence between the norms $\|\cdot\|_{D_A^{-1}}$ and $\|\cdot\|_{A^{-1}}$, which are defined as $\|\boldsymbol{w}_h'\|_{A^{-1}}^2 := \langle A^{-1}\boldsymbol{w}_h', \boldsymbol{w}_h'\rangle$ and $\|\boldsymbol{w}_h'\|_{D_A^{-1}}^2 := \langle D_A^{-1}\boldsymbol{w}_h', \boldsymbol{w}_h'\rangle$ for $\boldsymbol{w}_h' \in \boldsymbol{W}_h'$.

Using that and Cauchy-Schwarz inequality, we have

$$\begin{aligned}
(\mathcal{B}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{x})_{\mathcal{B}_D^{-1}} &= \|\boldsymbol{w}_h\|_A^2 + \langle B^T p_h, \boldsymbol{w}_h\rangle + \|BA^{-1}B^T p_h\|^2 \\
&\geqslant \|\boldsymbol{w}_h\|_A^2 - \|B^T p_h\|_{A^{-1}}\|\boldsymbol{w}_h\|_A + \|B^T p_h\|_{A^{-1}}^2 \\
&= \begin{pmatrix} \|\boldsymbol{w}_h\|_A \\ \|B^T p_h\|_{A^{-1}} \end{pmatrix}^T \begin{pmatrix} 1 & -\frac{1}{2} \\ -\frac{1}{2} & 1 \end{pmatrix} \begin{pmatrix} \|\boldsymbol{w}_h\|_A \\ \|B^T p_h\|_{A^{-1}} \end{pmatrix} \\
&\geqslant \frac{1}{2}(\|\boldsymbol{w}_h\|_A^2 + \|B^T p_h\|_{A^{-1}}^2) \\
&\geqslant \frac{1}{2}(\|\boldsymbol{w}_h\|_A^2 + c_2^{-1}\|B^T p_h\|_{D_A^{-1}}^2) \\
&\geqslant \xi_1\|\boldsymbol{x}\|_{\mathcal{B}_D^{-1}}^2,
\end{aligned}$$

with $\xi_1 = \frac{1}{2}\min\{1, c_2^{-1}\}$. On the other hand, again using the Cauchy-Schwarz inequality and equivalence of the norms $\|\cdot\|_{D_A^{-1}}$ and $\|\cdot\|_{A^{-1}}$ we get

$$\begin{aligned}
(\mathcal{B}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{y})_{\mathcal{B}_D^{-1}} &= \langle A\boldsymbol{w}_h, \boldsymbol{r}_h\rangle + \langle B^T p_h, \boldsymbol{r}_h\rangle + \langle BA^{-1}B^T p_h, q_h\rangle \\
&\leqslant \|\boldsymbol{w}_h\|_A\|\boldsymbol{r}_h\|_A + \|B^T p_h\|_{A^{-1}}\|\boldsymbol{r}_h\|_A + \|B^T p_h\|_{A^{-1}}\|B^T q_h\|_{A^{-1}} \\
&\leqslant \left(\|\boldsymbol{w}_h\|_A^2 + 2\|B^T p_h\|_{A^{-1}}^2\right)^{\frac{1}{2}}\left(2\|\boldsymbol{r}_h\|_A^2 + \|B^T q_h\|_{A^{-1}}^2\right)^{\frac{1}{2}} \\
&\leqslant \left(\|\boldsymbol{w}_h\|_A^2 + 2c_1^{-1}\|B^T p_h\|_{D_A^{-1}}^2\right)^{\frac{1}{2}}\left(2\|\boldsymbol{r}_h\|_A^2 + c_1^{-1}\|B^T q_h\|_{D_A^{-1}}^2\right)^{\frac{1}{2}} \\
&\leqslant \xi_2\|\boldsymbol{x}\|_{\mathcal{B}_D^{-1}}\|\boldsymbol{y}\|_{\mathcal{B}_D^{-1}},
\end{aligned}$$

for each $\boldsymbol{y} = (\boldsymbol{r}_h, q_h) \in \boldsymbol{W}_h \times Q_h, \boldsymbol{y} \neq \boldsymbol{0}$ with $\xi_2 = \max\{2, 2c_1^{-1}\}$, which concludes the proof. □

The next theorem states that if the conditions (4.12) hold then $\mathcal{M}_L$ and $\mathcal{A}$ are FOV-equivalent.

Theorem 4.7 . *If the conditions (4.12) hold and $\|I - H_{\boldsymbol{w}}A\|_A \leqslant \rho$ for $0 \leqslant \rho < 1$, then there exist constants $\xi_1, \xi_2 > 0$ independent of discretization and physical parameters such that for every $\boldsymbol{x} = (\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$, $\boldsymbol{x} \neq \boldsymbol{0}$,*

$$\xi_1 \leqslant \frac{(\mathcal{M}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{x})_{\mathcal{M}_D^{-1}}}{(\boldsymbol{x}, \boldsymbol{x})_{\mathcal{M}_D^{-1}}}, \quad \text{and} \quad \frac{\|\mathcal{M}_L\mathcal{A}\boldsymbol{x}\|_{\mathcal{M}_D^{-1}}}{\|\boldsymbol{x}\|_{\mathcal{M}_D^{-1}}} \leqslant \xi_2.$$

Proof. From the assumptions of the theorem we have $\|H_{\boldsymbol{w}}A\|_A \leqslant 1 + \rho$ in combination with Lemma 4.1, (4.12) and the Cauchy-Schwarz inequality, we have that

$$\begin{aligned}
(\mathcal{M}_L\mathcal{A}\boldsymbol{x}, \boldsymbol{x})_{\mathcal{M}_D^{-1}} &= \|\boldsymbol{w}_h\|_A^2 + \langle B^T p_h, \boldsymbol{w}_h\rangle + \langle B(H_{\boldsymbol{w}}A - I)\boldsymbol{w}_h, p_h\rangle + \|B^T p_h\|_{H_{\boldsymbol{w}}}^2 \\
&= \|\boldsymbol{w}_h\|_A^2 + \langle H_{\boldsymbol{w}}A\boldsymbol{w}_h, B^T p_h\rangle + \|B^T p_h\|_{H_{\boldsymbol{w}}}^2 \\
&\geqslant \|\boldsymbol{w}_h\|_A^2 - (1+\rho)\|\boldsymbol{w}_h\|_A\|B^T p_h\|_{H_{\boldsymbol{w}}} + \|B^T p_h\|_{H_{\boldsymbol{w}}}^2 \\
&= \begin{pmatrix} \|\boldsymbol{w}_h\|_A \\ \|B^T p_h\|_{H_{\boldsymbol{w}}} \end{pmatrix}^T \begin{pmatrix} 1 & -\frac{1+\rho}{2} \\ -\frac{1+\rho}{2} & 1 \end{pmatrix} \begin{pmatrix} \|\boldsymbol{w}_h\|_A \\ \|B^T p_h\|_{H_{\boldsymbol{w}}} \end{pmatrix} \\
&\geqslant \frac{1-\rho}{2}(\|\boldsymbol{w}_h\|_A^2 + \|B^T p_h\|_{H_{\boldsymbol{w}}}^2) \\
&\geqslant \frac{1-\rho}{2}\left(c_{2,\boldsymbol{w}}^{-1}\|\boldsymbol{w}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_{1,\boldsymbol{w}}c_2^{-1}\|B^T p_h\|_{D_A^{-1}}^2\right) \\
&\geqslant \frac{1-\rho}{2}\left(c_{2,\boldsymbol{w}}^{-1}\|\boldsymbol{w}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_{1,\boldsymbol{w}}c_2^{-1}c_{2,p}^{-1}\|p_h\|_{H_p^{-1}}^2\right) \\
&\geqslant \xi_1\|\boldsymbol{x}\|_{\mathcal{M}_D^{-1}}^2,
\end{aligned}$$

with $\xi_1 = \frac{1-\rho}{2}\min\{c_{2,\boldsymbol{w}}^{-1}, c_{1,\boldsymbol{w}}c_2^{-1}c_{2,p}^{-1}\}$.

Using the same conditions to show the upper bound, we obtain

$$
\begin{aligned}
(\mathcal{M}_L\mathcal{A}\boldsymbol{x},\boldsymbol{y})_{\mathcal{M}_D^{-1}} &= \langle A\boldsymbol{w}_h,\boldsymbol{r}_h\rangle + \langle B^T p_h,\boldsymbol{r}_h\rangle + \langle B(H_{\boldsymbol{w}}A-I)\boldsymbol{w}_h,q_h\rangle + \langle BH_{\boldsymbol{w}}B^T p_h,q_h\rangle \\
&\leqslant \|\boldsymbol{w}_h\|_A\|\boldsymbol{r}_h\|_A + \|B^T p_h\|_{A^{-1}}\|\boldsymbol{r}_h\|_A + \|(H_{\boldsymbol{w}}A-I)\boldsymbol{w}_h\|_A\|B^T q_h\|_{A^{-1}} \\
&\quad + \|B^T p_h\|_{H_{\boldsymbol{w}}}\|B^T q_h\|_{H_{\boldsymbol{w}}} \\
&\leqslant \|\boldsymbol{w}_h\|_A\|\boldsymbol{r}_h\|_A + \|B^T p_h\|_{A^{-1}}\|\boldsymbol{r}_h\|_A + \rho\|\boldsymbol{w}_h\|_A\|B^T q_h\|_{A^{-1}} \\
&\quad + \|B^T p_h\|_{H_{\boldsymbol{w}}}\|B^T q_h\|_{H_{\boldsymbol{w}}} \\
&\leqslant \left((1+\rho^2)\|\boldsymbol{w}_h\|_A^2 + \|B^T p_h\|_{A^{-1}}^2 + \|B^T p_h\|_{H_{\boldsymbol{w}}}^2\right)^{\frac{1}{2}} \\
&\quad \left(2\|\boldsymbol{r}_h\|_A^2 + \|B^T q_h\|_{A^{-1}}^2 + \|B^T q_h\|_{H_{\boldsymbol{w}}}^2\right)^{\frac{1}{2}} \\
&\leqslant \left((1+\rho^2)c_{1,\boldsymbol{w}}^{-1}\|\boldsymbol{w}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_1^{-1}(1+c_{1,\boldsymbol{w}}^{-1})\|B^T p_h\|_{D_A^{-1}}^2\right)^{\frac{1}{2}} \\
&\quad \left(2c_{1,\boldsymbol{w}}^{-1}\|\boldsymbol{r}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_1^{-1}(1+c_{1,\boldsymbol{w}}^{-1})\|B^T q_h\|_{D_A^{-1}}^2\right)^{\frac{1}{2}} \\
&\leqslant \left((1+\rho^2)c_{1,\boldsymbol{w}}^{-1}\|\boldsymbol{w}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_{1,p}^{-1}c_1^{-1}(1+c_{1,\boldsymbol{w}}^{-1})\|p_h\|_{H_p^{-1}}^2\right)^{\frac{1}{2}} \\
&\quad \left(2c_{1,\boldsymbol{w}}^{-1}\|\boldsymbol{r}_h\|_{H_{\boldsymbol{w}}^{-1}}^2 + c_{1,p}^{-1}c_1^{-1}(1+c_{1,\boldsymbol{w}}^{-1})\|q_h\|_{H_p^{-1}}^2\right)^{\frac{1}{2}} \\
&\leqslant \xi_2\|\boldsymbol{x}\|_{\mathcal{M}_D^{-1}}\|\boldsymbol{y}\|_{\mathcal{M}_D^{-1}}.
\end{aligned}
$$

This gives the upper bound with $\xi_2 = \max\{2c_{1,\boldsymbol{w}}^{-1}, c_{1,p}^{-1}c_1^{-1}(1+c_{1,\boldsymbol{w}}^{-1})\}$, which concludes the proof. □

Remark 4.3 . *Due to Lemma 4.1, the constants ξ_1 and ξ_2 are independent of h and parameters K, $K_{\boldsymbol{\nu}}$ and γ, but remain dependent on the shape regularity of the mesh. This means that the convergence rate of the preconditioned GMRES method with preconditioner $\mathcal{B}_L$ or $\mathcal{M}_L$ depends only on the shape regularity of the mesh.*

Similarly, we can derive the FOV-equivalence of $\mathcal{B}_U$ and $\mathcal{M}_U$ with $\mathcal{A}$. Since the proofs are similar to the two previous theorems, we omit them and only state the results here.

Theorem 4.8 . *There exist constants $\xi_1, \xi_2 > 0$ independent of discretization and physical parameters such that for any $\boldsymbol{x}' = \mathcal{B}_U^{-1}\boldsymbol{x}$ with $\boldsymbol{x} = (\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$, $\boldsymbol{x} \neq \boldsymbol{0}$,*

$$
\xi_1 \leqslant \frac{(\mathcal{A}\mathcal{B}_U\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{B}_D}}{(\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{B}_D}}, \quad \textit{and} \quad \frac{\|\mathcal{A}\mathcal{B}_U\boldsymbol{x}'\|_{\mathcal{B}_D}}{\|\boldsymbol{x}'\|_{\mathcal{B}_D}} \leqslant \xi_2.
$$

Theorem 4.9 . *If the conditions (4.12) hold and $\|I - H_{\boldsymbol{w}}A\|_A \leqslant \rho$ for $0 \leqslant \rho < 1$, then there exist constants $\xi_1, \xi_2 > 0$ independent of discretization and physical parameters such that for any $\boldsymbol{x}' = \mathcal{M}_U^{-1}\boldsymbol{x}$ with $\boldsymbol{x} = (\boldsymbol{w}_h, p_h) \in \boldsymbol{W}_h \times Q_h$, $\boldsymbol{x} \neq \boldsymbol{0}$,*

$$
\xi_1 \leqslant \frac{(\mathcal{A}\mathcal{M}_U\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{M}_D}}{(\boldsymbol{x}',\boldsymbol{x}')_{\mathcal{M}_D}}, \quad \textit{and} \quad \frac{\|\mathcal{A}\mathcal{M}_U\boldsymbol{x}'\|_{\mathcal{M}_D}}{\|\boldsymbol{x}'\|_{\mathcal{M}_D}} \leqslant \xi_2.
$$

Remark 4.4 . *Similarly, the constants ξ_1 and ξ_2 here are independent of h and parameters K, K_ν and γ, but remain dependent on the shape regularity of the mesh. This means that the convergence rate of the preconditioned GMRES method with preconditioner $\mathcal{B}_U$ or $\mathcal{M}_U$ depends only on the shape regularity of the mesh.*

5 Numerical results

In this section, we propose several test cases to verify the theory on the robustness of the preconditioners derived above. Both two and three dimensional examples emphasize common challenges in fracture flow simulations such as large aspect ratios of rock and fractures, complex fracture network structures and high heterogeneity in the permeability fields.

In each example below, a set of mixed-dimensional simplicial grids is generated on rock and fracture subdomains, where the coupling between the rock and fracture is employed by a separate mortar grid. Since our main objective is to show the robustness of our preconditioners for standard Krylov iterative methods, for the sake of simplicity, we take the mortar grid to be matching with the adjacent subdomain grids. However, the theory in Section 4 shows no restrictions to relative grid resolution between the rock, fracture and mortar grids. Furthermore, in [28] the discrete system remains well-posed with varying coarsening/refinement ratio for non-degenerate (normal) permeability values, which is one of our assumptions. Therefore, we expect that our block preconditioners give similar performance for general grids between the rock, fracture, and coupling part.

To solve the system (2.24), we use a Flexible Generalized Minimal Residual (FGMRES) method as an *outer* iterative solver, with the tolerance for the relative residual set to 10^{-6}. The block preconditioners designed in Section 4 are used to accelerate the convergence rate of FGMRES. Each preconditioner $\mathcal{B}_D, \mathcal{B}_L$ and $\mathcal{B}_U$ requires inversion of the diagonal blocks corresponding to flux and pressure degrees of freedom, while the spectrally equivalent versions $\mathcal{M}_D, \mathcal{M}_L$ and $\mathcal{M}_U$ approximate the inverses with appropriate iterative methods. For that, we implement both exact and inexact *inner* solvers. Solving each diagonal blocks exactly means we use the GMRES method with a relative residual tolerance set to 10^{-10}, while in the inexact case it is set to 10^{-3}. Inner GMRES is preconditioned with unsmoothed aggregation Algebraic Multigrid method (AMG) in a W-cycle.

For obtaining the mixed-dimensional geometry and discretization, we use the PorePy library [23], an open-source simulation tool for fractured and deformable porous media written in Python. Our preconditioners are implemented in HAZMATH library [3], a finite element solver library written in C, also where all solving computations are performed. The numerical tests were performed on a workstation with an 8-core 3GHz Intel Xeon "Sandy Bridge" CPU and 256 GB of RAM.

5.1 Example: two-dimensional Geiger network

In the first example, we consider the test case presented in the benchmark study [15]. The domain $\Omega = (0,1)^2$, depicted in Figure 2, has unitary permeability $K = \boldsymbol{I}$ for the rock matrix and it is divided into 10 sub-domains by a set of fractures with aperture γ. In our case, we set the tangential and normal permeability of the fractures to be constant throughout the whole network, and vary the value from blocking to conducting the flow. The tangential fracture permeability is denoted as K_f to avoid confusion with the rock permeability. At the boundary, we impose zero flux condition on the top and bottom, unitary pressure on the right, and flux equal to -1 on the left. The boundary conditions are applied to both the rock matrix and the fracture network. The numerical solution to this problem is also illustrated in Figure 2.

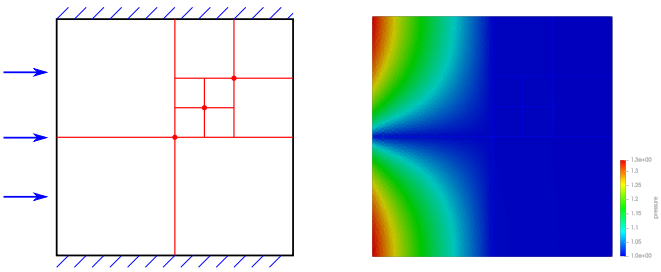


Fig. 2: (Left) Graphical representation of the domain and fracture network geometry of Example 5.1. (Right) Pressure solution for a case of conducting fractures.

Our goal is to investigate the robustness of the block preconditioners with respect to discretization parameter h and physical parameters γ, K_f and $K_{\boldsymbol{\nu}}$. To this end, we generate a series of tests in which we vary the magnitude of one of the parameters, while setting others to a fixed value. This also tests the heterogeneity ratios between the porous medium and the fractures, since we keep spatial and physical parameters of the porous medium unitary. We compute and compare number of iterations of the outer solver for both exact and inexact implementations of the proposed preconditioners. This way we clearly see if the stability of the proposed preconditioners depends on one or a combination of given parameters.

The results of these robustness tests on are summarized in Tables 1 – 3. We start with setting $K_f = K_{\boldsymbol{\nu}} = \boldsymbol{I}$ that, together with rock permeability K, gives a global homogeneous unitary permeability field. We also fix the aperture to $\gamma = 10^{-2}$. Refining the initial coarse grid by a factor of 2 recursively, Table 1 demonstrates the robustness of all block preconditioners with respect to the mesh size h. Additionally, the different implementations of the preconditioners result in similar behavior of the solver. We notice that the block triangular

	Inexact			*Exact*		
h	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$	$\mathcal{B}_D$	$\mathcal{B}_L$	$\mathcal{B}_U$
1/4	20	13	12	19	10	10
1/8	19	13	11	19	10	10
1/16	19	13	11	19	10	10
1/32	19	13	11	19	10	10
1/64	19	13	11	19	10	10

Table 1: Number of iterations of outer FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.1. Varying mesh size h while aperture is set to $\gamma = 1/100$ and all the permeabilities are set to $K = K_f = K_{\boldsymbol{\nu}} = \boldsymbol{I}$.

	Inexact			*Exact*		
γ	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$	$\mathcal{B}_D$	$\mathcal{B}_L$	$\mathcal{B}_U$
1	21	16	14	21	11	11
1/10	19	13	12	19	10	10
1/100	19	13	11	19	10	10
1/1000	19	13	11	19	10	10
1/10000	19	13	11	19	10	10

Table 2: Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.1. Varying aperture γ while mesh size is set to $h = 1/16$ and all the permeabilities are set to $K = K_f = K_{\boldsymbol{\nu}} = \boldsymbol{I}$.

preconditioners $\mathcal{B}_L$ and $\mathcal{B}_U$ show a slightly better performance compared the block diagonal $\mathcal{B}_D$ as expected. The same behavior can be observed for inexact preconditioners $\mathcal{M}_L$ and $\mathcal{M}_U$ in comparison to $\mathcal{M}_D$. This is expected since the block triangular preconditioners better approximate the inverse of the stiffness matrix in (2.24). It is noteworthy to mention that the action of the block triangular preconditioners is more expensive computationally than the action of the block diagonal preconditioners. Similar performance can also be observed in Table 2, where we scale down the fracture width on a fixed grid of mesh size $h = 1/16$. Lastly, in Table 3 we test the influence of the heterogeneity in the permeability fields. We keep the mesh size to be $h = 1/16$ and fracture aperture to be $\gamma = 10^{-2}$, while introducing both conducting and blocking fracture network in the porous medium. Again, the robustness is evident in terms of the number of outer FGMRES iterations with both exact and inexact block preconditioners. The block triangular preconditioners, $\mathcal{B}_L$, $\mathcal{B}_U$, $\mathcal{M}_L$, and $\mathcal{M}_U$, provide somewhat lower values comparing to their block diagonal counterpart.

5.2 Example: two-dimensional complex network

This example is chosen to demonstrate the robustness of the block preconditioners on a more realistic fracture network. Such a complex fracture configuration often occurs in geological rock simulations and the geometrical and physical properties of the fracture network can significantly influence the stability of the solving method. This is especially seen in mpartitioning the frac-

	Inexact			*Exact*		
K	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$	$\mathcal{B}_D$	$\mathcal{B}_L$	$\mathcal{B}_U$
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	13	10	8	11	7	6
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	13	8	8	13	7	7
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	13	8	8	13	7	7
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	22	16	13	19	11	10
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	19	13	11	19	10	10
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	19	13	12	19	10	10
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	26	19	19	21	13	12
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	23	17	15	23	13	12
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	23	17	15	23	14	12

Table 3: Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.1. Varying the permeability K_f and $K_{\boldsymbol{\nu}}$ while mesh size is set to $h = 1/16$ and aperture is set to $\gamma = 1/100$.

tured porous medium domain where sharp tips and very acute intersections may decrease the shape regularity of the mesh. Since our analysis shows that the performance of our block preconditioners only depends on the shape regularity of the mesh, for this complex network example, we expect to see that the preconditioners are still robust with respect to physical and discretization parameters, but slightly more iterations may be required due to the worse shape regularity of the mesh when comparing to Example 5.1.

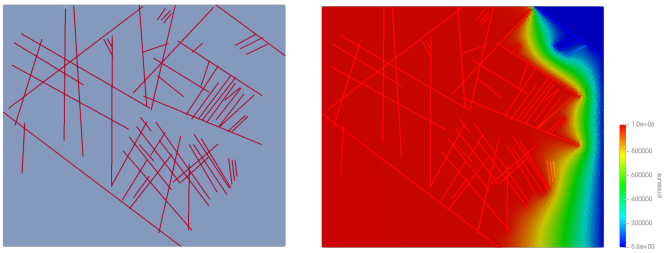
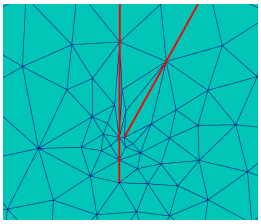


Fig. 3: (Left) Graphical representation of the two-dimensional domain and fracture network geometry of Example 5.2. (Right) Pressure solution for a case of conducting fractures.

This example is chosen from benchmark study [15] – a set of fractures from an interpreted outcrop in the Sotra island, near Bergen in Norway. The set includes 64 fractures grouped in 13 different connected networks. The porous medium domain has size 700 m $\times$ 600 m with uniform matrix permeability $K = 10^{-14}\boldsymbol{I}$ m^2. All the fractures have the same scalar permeability $K_f = 10^{-8}\boldsymbol{I}$ m^2 and aperture $\gamma = 10^{-2}$ m. Also, no-flow boundary condition are

imposed on top and bottom, with pressure 1013250 Pa on the left and 0 Pa on the right boundary.



	Inexact		
h	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
$L/4$	63	51	40
$L/8$	67	50	44
$L/16$	61	47	42
$L/32$	55	39	34
$L/64$	47	33	29

Fig. 4: (Left) Mesh around one of the complex tips in the fracture network, where $h = L/64$. (Right) Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.2. We refine the mesh relatively to domain length $L = 600$.

For the comparison with the previous example, we refine the mesh size h with respect to the width of the domain $L = 600$. However, due to the complex fracture structure, it is possible to end up with smaller and badly shaped elements in the rock matrix grid around the tips and intersections of the fractures. For example, see Figure 4 on the left. The coarser the mesh is, the more irregular the elements are, especially when partisioning in between many tightly packed fractures. Therefore, we expect that the solver requires more iterations to converge on coarser meshes. This is evident in the table on the right in Figure 4. We see the reduction of number of iterations when refining the mesh in all the cases, with the lowest number required by the block upper triangular $\mathcal{M}_U$. We also notice that the solver manages to provide the correct solution on all given meshes in an acceptable number of iterations. The results are slightly worse than the previous example, but keep in mind that the complex geometry is still an important factor in the mesh structure and, therefore, influences the convergence rate since the shape regularity of the mesh deteriorates. For complex fracture networks, it is beneficial to invest in constructing a more regular mesh of the fractured porous medium and then applying the proposed block preconditioners in the iterative solvers.

5.3 Example: three-dimensional Geiger network

This last example considers the simulations of a 3D problem taken from another benchmark study [5], a three-dimensional analogue to the test case in Subsection 5.1. The geometry is extended to the unit cube and the fracture network now consists of nine intersecting planes (see Figure 5). As before, we take the rock matrix permeability K to be the identity tensor, while we vary the tangential K_f and the normal $K_{\boldsymbol{\nu}}$ permeability, as well as the fracture aperture γ.

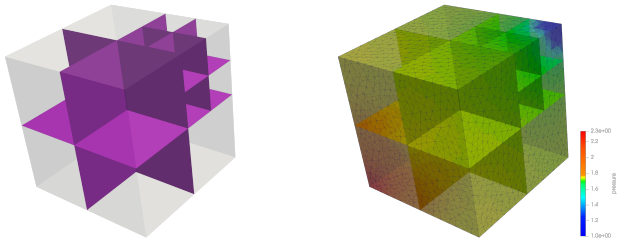


Fig. 5: (Left) Graphical representation of the three-dimensional domain and fracture network geometry of Example 5.3. (Right) Pressure solution for a case of conducting fractures.

For a fair comparison with the two-dimensional case, we perform similar robustness tests of the preconditioners to study the effect of mesh refinement, as well as permeability and aperture changes. However, we stick to only inexact preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ since they are less computationally expensive and perform comparably well, which makes them good choices in practice. The results are presented in Tables 4–6. We can see that the simulations confirm the findings of Section 4: all block preconditioners show robustness with respect to the discretization and physical parameters. The block diagonal preconditioner requires a slightly higher number of iterations to converge compared to block triangular ones, as we saw in the previous example.

	Inexact		
h	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
1/4	26	18	15
1/8	26	17	15
1/16	24	16	14
1/32	24	16	13
1/64	24	16	12

Table 4: Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.2. Varying mesh size h while aperture is set to $\gamma = 1/100$ and all permeabilities are set to $K = K_f = K_{\boldsymbol{\nu}} = \boldsymbol{I}$.

In 3D simulations it is also important to study the overall computational complexity of the solving method. For that, we analyze in Figure 6 the required CPU time of the FGMRES solver preconditioned with each block preconditioner $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$. All preconditioners show a optimal $\mathcal{O}(N_{dof})$ complexity, where N_{dof} is the number of degrees of freedom of the discretized system. Notice that even though the block triangular pair of preconditioners

	Inexact		
γ	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
1	24	16	14
1/10	24	16	13
1/100	24	16	14
1/1000	26	16	14
1/10000	26	17	14

Table 5: Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.3. Varying aperture γ while mesh size is set to $h = 1/16$ and all permeabilities are set to $K = K_f = K_{\boldsymbol{\nu}} = \boldsymbol{I}$.

	Inexact		
K	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	28	19	20
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	26	17	14
$K_f = 10^{-4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	28	17	14
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	26	21	18
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	24	16	14
$K_f = \boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	26	17	14
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{-4}\boldsymbol{I}$	24	16	17
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = \boldsymbol{I}$	22	15	13
$K_f = 10^{4}\boldsymbol{I}, K_{\boldsymbol{\nu}} = 10^{4}\boldsymbol{I}$	22	15	13

Table 6: Number of outer iterations of FGMRES solver with exact and inexact block preconditioners for the case study in Example 5.3. Varying the permeability K_f and $K_{\boldsymbol{\nu}}$ while mesh size is set to $h = 1/16$ and aperture is set to $\gamma = 1/100$.

require solving a denser system, it is still time-wise less expensive due to a lower number of iterations needed to converge.

6 Conclusions

We have presented block preconditioners for linear systems arising in mixed-dimensional modeling of single-phase flow in fractured porous media. Our approach is based on the stability theory of the mixed finite element discretization of the model which we extended to provide an efficient way to solve large systems with standard Krylov subspace iterative methods. We have thoroughly analyzed the robustness of the derived preconditioners with regard to discretization and physical parameters by proving norm and field-of-value equivalence to the original system. Our theory has also been supported by several numerical examples of 2D and 3D flow simulations.

It is noteworthy to mention that even though our analysis depends on a more regular mesh, the numerical results show that the preconditioners still perform well since the mixed-dimensional discretization approach handles fractures independently of the rock matrix and, therefore, generates simpler meshes in most fracture network cases. The large aspect ratios that parametrize the model then become the main stability problem, which we have successfully overcome with the proposed block preconditioners. This is impor-

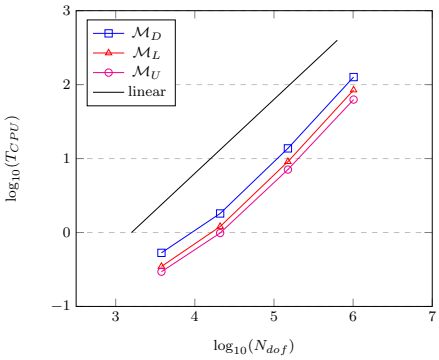


Fig. 6: CPU time T_{CPU} of the preconditioned FGMRES algorithm verses number of degrees of freedom N_{dof} of the discretized system in Example 5.3.

tant for implementations in general geological simulations where the rock-fracture configuration can be quite complex and can contain a large number of fractures of different width and length.

We conclude by recalling that the alternative approach to block preconditioners mentioned in the beginning of Section 3 is a non-trivial extension to this work and a part of an ongoing research.

7 Acknowledgements

A special thanks is extended to James Adler, Alessio Fumagalli and Eirik Keilegavlen for valuable comments and discussions on the presented work. The authors also would like to thank Casey Cavanaugh for improving the style of the presentation.

References

1. Adler, J.H., Gaspar, F.J., Hu, X., Ohm, P., Rodrigo, C., Zikatanov, L.T.: Robust preconditioners for a new stabilized discretization of the poroelastic equations. arXiv:1905.10353 [math.NA] (2019). URL https://arxiv.org/abs/1905.10353
2. Adler, J.H., Gaspar, F.J., Hu, X., Rodrigo, C., Zikatanov, L.T.: Robust block preconditioners for biot's model. In: P.E. Bjørstad, S.C. Brenner, L. Halpern, H.H. Kim, R. Kornhuber, T. Rahman, O.B. Widlund (eds.) Domain Decomposition Methods in Science and Engineering XXIV, pp. 3–16. Springer International Publishing, Cham (2018)
3. Adler, J.H., Hu, X., Zikatanov, L.T.: HAZMATH: A simple finite element, graph and solver library
4. Arrarás, A., Gaspar, F.J., Portero, L., Rodrigo, C.: Monolithic mixed-dimensional multigrid methods for single-phase flow in fractured porous media. arXiv:1811.01264 [math.NA] (2018). URL https://arxiv.org/abs/1811.01264

5. Berre, I., Boon, W., Flemisch, B., Fumagalli, A., Gläser, D., Keilegavlen, E., Scotti, A., Stefansson, I., Tatomir, A.: Call for participation: Verification benchmarks for single-phase flow in three-dimensional fractured porous media. arXiv:1809.06926 [math.NA] (2018). URL https://arxiv.org/abs/1809.06926
6. Boffi, D., Brezzi, F., Fortin, M.: Mixed Finite Element Methods and Applications, *Springer Series in Computational Mathematics*, vol. 44. Springer Berlin Heidelberg, Berlin, Heidelberg (2013). DOI 10.1007/978-3-642-36519-5
7. Boon, W., Budiša, A., Hu, X.: Mixed-dimensional auxiliary space preconditioners. In preparation (2019)
8. Boon, W.M., Nordbotten, J.M., Vatne, J.E.: Functional analysis and exterior calculus on mixed-dimensional geometries. arXiv:1710.00556v3 [math.AP] (2018). URL https://arxiv.org/abs/1710.00556v3
9. 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). DOI 10.1137/17M1139102. URL https://doi.org/10.1137/17M1139102
10. Brenner, S.C., Scott, L.R.: The mathematical theory of finite element methods, *Texts in Applied Mathematics*, vol. 15, third edn. Springer, New York (2008). DOI 10.1007/978-0-387-75934-0. URL https://doi.org/10.1007/978-0-387-75934-0
11. Brezzi, F.: On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge **8**(R-2), 129–151 (1974)
12. D'Angelo, C., Scotti, A.: A mixed finite element method for darcy flow in fractured porous media with non-matching grids. ESAIM: Mathematical Modelling and Numerical Analysis **46**(2), 465–489 (2011). URL http://eudml.org/doc/222154
13. Eisenstat, S., Elman, H., Schultz, M.: Variational iterative methods for nonsymmetric systems of linear equations. SIAM Journal on Numerical Analysis **20**(2), 345–357 (1983). DOI 10.1137/0720023. URL https://doi.org/10.1137/0720023
14. Elman, H.C.: Iterative methods for large, sparse, nonsymmetric systems of linear equations. Ph.D. thesis, New Haven, CT, USA (1982). AAI8222744
15. 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. Advances in Water Resources **111**, 239–258 (2018). DOI 10.1016/j.advwatres.2017.10.036. URL https://www.sciencedirect.com/science/article/pii/S0309170817300143
16. Formaggia, L., Fumagalli, A., Scotti, A., Ruffo, P.: A reduced model for Darcy's problem in networks of fractures. ESAIM Math. Model. Numer. Anal. **48**(4), 1089–1116 (2014). DOI 10.1051/m2an/2013132. URL https://doi.org/10.1051/m2an/2013132
17. Formaggia, Luca, Scotti, Anna, Sottocasa, Federica: Analysis of a mimetic finite difference approximation of flows in fractured porous media. ESAIM: M2AN **52**(2), 595–630 (2018). DOI 10.1051/m2an/2017028. URL https://doi.org/10.1051/m2an/2017028
18. Frih, N., Martin, V., Elizabeth Roberts, J., Saada, A.: Modeling fractures as interfaces with nonmatching grids. Computational Geosciences **16**, 1043–1060 (2012). DOI 10.1007/s10596-012-9302-6
19. Fumagalli, Alessio, Keilegavlen, Eirik: Dual virtual element methods for discrete fracture matrix models. Oil Gas Sci. Technol. - Rev. IFP Energies nouvelles **74**, 41 (2019). DOI 10.2516/ogst/2019008. URL https://doi.org/10.2516/ogst/2019008
20. Greenbaum, A.: Iterative Methods for Solving Linear Systems. Society for Industrial and Applied Mathematics (1997). DOI 10.1137/1.9781611970937. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611970937
21. Hiptmair, R., Xu, J.: Nodal auxiliary space preconditioning in h(curl) and h(div) spaces. SIAM Journal on Numerical Analysis **45**(6), 2483–2509 (2007). DOI 10.1137/060660588
22. Karimi-Fard, M., J. Durlofsky, L., Aziz, K.: An efficient discrete-fracture model applicable for general-purpose reservoir simulators. SPE Journal - SPE J **9**, 227–236 (2004). DOI 10.2118/88812-PA
23. Keilegavlen, E., Fumagalli, A., Berge, R., Stefansson, I., Berre, I.: `PorePy`: An open source simulation tool for flow and transport in deformable fractured rocks. arXiv:1712.00460 [cs.CE] (2017). URL https://arxiv.org/abs/1712.00460
24. Kolev, T., Vassilevski, P.: Parallel auxiliary space amg solver for $h(div)$ problems. SIAM Journal on Scientific Computing **34**(6), A3079–A3098 (2012). DOI 10.1137/110859361

25. Loghin, D., Wathen, A.J.: Analysis of preconditioners for saddle-point problems. SIAM J. Sci. Comput. **25**(6), 2029–2049 (2004). DOI 10.1137/S1064827502418203. URL https://doi.org/10.1137/S1064827502418203
26. Mardal, K.A., Winther, R.: Preconditioning discretizations of systems of partial differential equations. Numer. Linear Algebra Appl. **18**(1), 1–40 (2011). DOI 10.1002/nla.716. URL https://doi.org/10.1002/nla.716
27. Nédélec, J.C.: Mixed finite elements in $\mathbf{R}^3$. Numer. Math. **35**(3), 315–341 (1980). DOI 10.1007/BF01396415. URL https://doi.org/10.1007/BF01396415
28. Nordbotten, J.M., Boon, W.M., Fumagalli, A., Keilegavlen, E.: Unified approach to discretization of flow in fractured porous media. Computational Geosciences pp. 1–13 (2018). DOI 10.1007/s10596-018-9778-9. URL https://doi.org/10.1007/s10596-018-9778-9
29. Raviart, P.A., Thomas, J.M.: A mixed finite element method for 2nd order elliptic problems. In: Mathematical aspects of finite element methods, Lecture Notes in Math., Vol. 606, pp. 292–315. Springer, Berlin (1977)
30. Sandve, T.H., Berre, I., Nordbotten, J.M.: An efficient multi-point flux approximation method for discrete fracture–matrix simulations. Journal of Computational Physics **231**(9), 3784 – 3800 (2012). DOI https://doi.org/10.1016/j.jcp.2012.01.023. URL http://www.sciencedirect.com/science/article/pii/S0021999112000447
31. Schwenck, N., Flemisch, B., Helmig, R., Wohlmuth, B.I.: Dimensionally reduced flow models in fractured porous media: crossings and boundaries. Computational Geosciences **19**(6), 1219–1230 (2015). DOI 10.1007/s10596-015-9536-1. URL https://doi.org/10.1007/s10596-015-9536-1

Paper B

Mixed-Dimensional Auxiliary Space Preconditioners

A. Budiša, W. M. Boon, X. Hu

SIAM Journal on Scientific Computing. In review.

arXiv:1910.04704 [math.NA].

Mixed-Dimensional Auxiliary Space Preconditioners

Ana Budiša*, Wietse M. Boon†, Xiaozhe Hu‡

Abstract

This work introduces nodal auxiliary space preconditioners for discretizations of mixed-dimensional partial differential equations. We first consider the continuous setting and generalize the regular decomposition to this setting. With the use of conforming mixed finite element spaces, we then expand these results to the discrete case and obtain a decomposition in terms of nodal Lagrange elements. In turn, nodal preconditioners are proposed analogous to the auxiliary space preconditioners of Hiptmair and Xu [16]. Numerical experiments show the performance of this preconditioner in the context of flow in fractured porous media.

1 Introduction

In recent work [7,8,25], exterior calculus and its finite element discretization has been extended to the mixed-dimensional geometries. More precisely, for an n-dimensional domain, sub-manifolds of dimension $n-1$ and their intersections of dimension $n-2$, $n-3$, and so on are considered. Suitable spaces of alternating k-forms are introduced and equipped with proper inner products and norms. Based on well-defined differential operators and codifferential operators, a de Rham complex for the mixed-dimensional geometry is proposed as well. Such a generalization of fixed-dimensional finite element exterior calculus [4] provides a unified theoretical framework for mixed-dimensional partial differential equations (PDEs) as well as their finite element discretizations. This has wide applications in mathematical modeling and simulation, e.g., shells, membranes, fractures, and geological formations [5, 10, 26].

One important result in the fixed-dimensional finite element exterior calculus is the stable regular decomposition and its discrete variant [14,16]. Understanding the stable regular decompositions is at the heart of designing robust preconditioners for solving $H(\nabla\times)$- and $H(\nabla\cdot)$-elliptic problems based on the auxiliary space preconditioning framework [24, 29]. Based on the discrete regular decomposition, preconditioners for $H(\nabla\times)$- and $H(\nabla\cdot)$-elliptic problems can be developed, which consists of solving several $H(\nabla)$-elliptic problems and simple smoothing steps in the original space. Numerical results [19, 20] have shown the effectiveness of such preconditioners.

In this work, we extend the stable regular decomposition to the mixed-dimensional geometries. Unlike the fixed-dimensional case, where the stable regular decomposition is usually derived based on the corresponding regular inverse, in the mixed-dimensional setting, we construct the regular decomposition directly by establishing such a regular decomposition on each individual sub-manifold and then combining them together properly. Discrete regular decomposition is also generalized to the mixed-dimensional geometries. The construction of the discrete version is similar with the fixed-dimensional counterpart. The resulting discrete regular decomposition also involves an extra high-frequency term comparing with the stable regular decomposition as expected. Based on discrete regular decomposition and auxiliary space preconditioning framework, we are able to develop robust preconditioners for solving abstract model mixed-dimensional PDEs (5.1).

In order to demonstrate the effectiveness of the proposed auxiliary space preconditioner for solving mixed-dimensional PDEs, we consider flow in fractured porous media as an example, which is modeled by Darcy's law and conservation of mass in the mixed-dimensional setting. After discretization, robust block preconditioners are designed based on the well-posedness of the discrete

[1]Department of Mathematics, University of Bergen, Allegaten 41, P. O. Box 7803, N-5020 Bergen, Norway. Ana.Budisa@uib.no

[2]Department of Mathematics, KTH Royal Institute of Technology, Lindstedtsvägen 25, 10044 Stockholm, Sweden.

[3]Department of Mathematics, Tufts University, 503 Boston Ave, Medford, MA 02155, USA.

PDEs based on the framework developed in [22,23]. The mixed-dimensional auxiliary space preconditioner is used to invert one of the diagonal blocks in the block preconditioners. The effectiveness of the preconditioners are verified both theoretically and numerically.

The rest of the paper is organized as follows. Section 2 introduces the mixed-dimensional geometries and function spaces. Mixed-dimensional regular decomposition is derived in Section 3 and the discrete version is proposed in Section 4. In Section 5, we describe the mixed-dimensional auxiliary space preconditioner for abstract mixed-dimensional PDEs and an example, flow in fractured porous media, is introduced in Section 6. Numerical results are shown in Section 7 to demonstrate the robustness and effectiveness of the proposed preconditioners, and the conclusions are given in Section 8.

2 Preliminaries

In this section, we first introduce the definition of a mixed-dimensional geometry and the conventions used when referring to certain structures. Next, we summarize the relevant concepts from functional analysis for the fixed-dimensional case as well as the generalization to the mixed-dimensional setting. For a more rigorous and detailed exposition of these results, we refer the readers to [7].

2.1 Geometry

Given a contractible Lipschitz domain $Y \subset \mathbb{R}^n$ with $n \leq 3$. Within Y, we introduce disjoint manifolds $\Omega_i^{d_i}$ with i being the index from a global set I and d_i being the dimension. The superscript generally is omitted. Let I^d be the subset of I containing all indices i with $d_i = d$.

We refer to the union of all manifolds Ω_i as the mixed-dimensional geometry Ω and denote the subset of d-manifolds as Ω^d, i.e.

$$\Omega := \bigcup_{i \in I} \Omega_i, \qquad \Omega^d := \bigcup_{i \in I^d} \Omega_i.$$

For each Ω_i with $i \in I$, we form a connection to each lower-dimensional manifold that coincides with (a portion of) its boundary. Each of these connections is endowed with a unique index j. Then, let i_j be the index of the lower-dimensional manifold such that $\Omega_{i_j} \subseteq \partial\Omega_i$. We denote $\partial_j\Omega_i$ as the corresponding boundary of dimension $d_j := d_{i_j}$.

For each $i \in I$ and $d < d_i$, we define I_i^d as the set of indices j such that $\partial_j\Omega_i$ is d-dimensional. Moreover, let I_i contain all indices j for which $\partial_j\Omega_i$ is not empty:

$$I_i^d := \left\{ j : \ \exists i_j \in I^d \text{ such that } \partial\Omega_i \cap \Omega_{i_j} \neq \emptyset \right\}, \qquad I_i := \bigcup_{d=0}^{d_i - 1} I_i^d.$$

To exemplify a mixed-dimensional geometry Ω, let us consider Figure 1 and its corresponding index sets I_i^d. In this case, we have $I_2^1 = \{12, 13\}$ with $i_{12} = 5$ and $i_{13} = 3$ and $I_2^0 = 16$ with $i_{16} = 6$. We note that two distinct portions of the boundary $\partial\Omega_1$ coincide geometrically with Ω_4. This is represented by the two indices $\{9, 10\} \subset I_1^1$ with $i_9 = i_{10} = 4$.

2.2 Function Spaces

The next step is to define a function $\mathfrak{a}$ on the mixed-dimensional geometry. We do this using the language of exterior calculus [27]. We first introduce local function spaces on each subdomain which form the building blocks for the mixed-dimensional generalization.

For $i \in I$, let $\Lambda^k(\Omega_i)$ denote the space of differential k-forms on Ω_i. Let $L^2\Lambda^k(\Omega_i)$ denote the space of square integrable k-forms and $H\Lambda^k(\Omega_i)$ denote its subspace of forms with square integrable differential. In other words, let

$$L^2\Lambda^k(\Omega_i) := \{a_i \in \Lambda^k(\Omega_i) : \ \|a_i\|_{L^2(\Omega_i)} < \infty\},$$
$$H\Lambda^k(\Omega_i) := \{a_i \in L^2\Lambda^k(\Omega_i) : \ \mathrm{d}a_i \in L^2\Lambda^{k+1}(\Omega_i)\}.$$

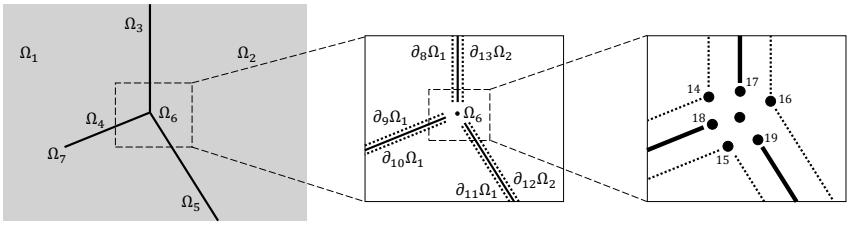


Figure 1: Example of a mixed-dimensional geometry with $n = 2$. On the left, Ω_i labels each d_i-manifold with $i \in I$. The index sets I^d for this geometry are as follows. $I^2 = \{1, 2\}$ represents the 2-manifolds, $I^1 = \{3, 4, 5\}$ denotes the 1-manifolds and the 0-manifolds in this geometry have indices $i \in I^0 = \{6, 7\}$. The middle of the figure illustrates the enumeration of boundaries $\partial_j\Omega_i$ for $i \in I^2$ and $j \in I^1_i$. On the right, the indices j are shown with the property $i_j = 6$.

With the exterior derivative d, the spaces $H\Lambda^k$ form a cochain complex, known as the de Rham complex:

$$H\Lambda^0(\Omega_i) \xrightarrow{\mathrm{d}} H\Lambda^1(\Omega_i) \xrightarrow{\mathrm{d}} \cdots \xrightarrow{\mathrm{d}} H\Lambda^{d_i-1}(\Omega_i) \xrightarrow{\mathrm{d}} H\Lambda^{d_i}(\Omega_i).$$

We often use the correspondence of this complex to conventional Sobolev spaces. For $d_i = 3$, this representation of the de Rham complex is given by

$$H(\nabla, \Omega_i) \xrightarrow{\nabla} H(\nabla\times, \Omega_i) \xrightarrow{\nabla\times} H(\nabla\cdot, \Omega_i) \xrightarrow{\nabla\cdot} L^2(\Omega_i). \tag{2.1}$$

Here, $L^2(\Omega_i)$ is the space of square-integrable functions on Ω_i and the space $H(\nabla, \Omega_i)$ is its subspace of functions with square-integrable gradients, typically denoted by $H^1(\Omega_i)$. The spaces $H(\nabla\times, \Omega_i)$ and $H(\nabla\cdot, \Omega_i)$ are defined analogously.

We use the local spaces $L^2\Lambda^k$ and $H\Lambda^k$ to introduce the Sobolev spaces containing mixed-dimensional differential k-forms on Ω. For brevity, we omit the reference to the geometry and define

$$L^2\mathfrak{L}^k := \prod_{i\in I}\{a_i \in L^2\Lambda^{k_i}(\Omega_i) : \mathrm{Tr}_j a_i \in L^2\Lambda^{k_i}(\Omega_{i_j}),\ \forall j \in I_i\},$$
$$H\mathfrak{L}^k := \prod_{i\in I}\{a_i \in H\Lambda^{k_i}(\Omega_i) : \mathrm{Tr}_j a_i \in H\Lambda^{k_i}(\Omega_{i_j}),\ \forall j \in I_i\},$$

with $k_i := d_i - (n-k)$. Here, and in the following, we interpret $\Lambda^k(\Omega_i)$ as zero for $k < 0$ and $k > d_i$. Thus, we emphasize that $H\mathfrak{L}^k$ is zero on manifolds Ω_i with $d_i < n-k$. The operator Tr_j is a trace operator that restricts a form a_i to $\partial_j\Omega_i$. We emphasize that for a given $\mathfrak{a} = (a_i)_{i\in I} \in H\mathfrak{L}^k$, the component a_i has a well-defined trace on each $\partial_j\Omega_i$ for $j \in I_i$ with $k_i \le d_j < d_i$ by definition.

The Gothic font is used to denote a mixed-dimensional differential form $\mathfrak{a} \in H\mathfrak{L}^k$ and we revert to classic fonts with a subscript i to denote its component defined on Ω_i. In the analysis, we often use the corresponding restriction operator ι_i defined such that

$$\iota_i\mathfrak{a} = a_i.$$

Next, we define the jump operator $\mathbb{d} : H\mathfrak{L}^k \mapsto H\mathfrak{L}^{k+1}$. For each $i \in I$, let

$$\iota_i\mathbb{d}\mathfrak{a} = (-1)^{n-k}\sum_{l\in I^{d_i+1}}\sum_{\{j\in I_l:\ i_j=i\}}\mathrm{Tr}_j a_l.$$

For more details on the definition of $\mathbb{d}$, we refer to [7]. The mixed-dimensional differential $\mathfrak{d}$ is formed as the sum of $\mathbb{d}$ and the exterior derivative d such that

$$\iota_i\mathfrak{d}\mathfrak{a} = \mathrm{d}a_i + \iota_i\mathbb{d}\mathfrak{a}, \qquad\qquad i \in I.$$

We introduce the following norms for $\mathfrak{a} \in H\mathfrak{L}^k$:

$$\|\mathfrak{a}\|^2_{L^2\mathfrak{L}^k} := \sum_{i\in I} \|a_i\|^2_{L^2(\Omega_i)} + \sum_{d=k_i}^{d_i} \sum_{j\in I_i^d} \|\mathrm{Tr}_j a_i\|^2_{L^2(\Omega_{i_j})},$$
$$\|\mathfrak{a}\|^2_{H\mathfrak{L}^k} := \|\mathfrak{a}\|^2_{L^2\mathfrak{L}^k} + \|\mathfrak{d}\mathfrak{a}\|^2_{L^2\mathfrak{L}^{k+1}}.$$

The inner products that naturally induce these norms are denoted by $(\cdot,\cdot)_{L^2\mathfrak{L}^k}$ and $(\cdot,\cdot)_{H\mathfrak{L}^k}$, respectively. The spaces $H\mathfrak{L}^k$ form a cochain complex which we refer to as the mixed-dimensional de Rham complex:

$$H\mathfrak{L}^0 \xrightarrow{\mathfrak{d}} H\mathfrak{L}^1 \xrightarrow{\mathfrak{d}} \dots \xrightarrow{\mathfrak{d}} H\mathfrak{L}^{n-1} \xrightarrow{\mathfrak{d}} H\mathfrak{L}^n. \tag{2.2}$$

This complex has several key properties, which we present in the following Lemma.

Lemma 2.1. *The complex* (2.2) *satisfies the following:*

- *All exact forms are closed: each* $\mathfrak{a} \in H\mathfrak{L}^k$ *satisfies*

$$\mathfrak{d}(\mathfrak{d}\mathfrak{a}) = 0. \tag{2.3}$$

- *All closed forms are exact: for each* $\mathfrak{a} \in H\mathfrak{L}^k$ *with* $\mathfrak{d}\mathfrak{a} = 0$*, there exists a* $\mathfrak{b} \in H\mathfrak{L}^{k-1}$ *such that*

$$\mathfrak{d}\mathfrak{b} = \mathfrak{a}, \qquad \|\mathfrak{b}\|_{H\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{a}\|_{H\mathfrak{L}^k} \tag{2.4}$$

Proof. The proof can be found in [7]. □

We represent this complex in terms of local spaces for each dimension. For $n = 3$, these local spaces are then organized in the following diagram:

$$\begin{array}{ccccccccc}
H\mathfrak{L}^0 & & H(\nabla,\mathrm{Tr};\Omega^3) & & & & & & \\
\downarrow \mathfrak{d} & & \downarrow \nabla & \searrow \mathfrak{d} & & & & & \\
H\mathfrak{L}^1 & & H(\nabla\times,\mathrm{Tr};\Omega^3) & & H(\nabla^\perp,\mathrm{Tr};\Omega^2) & & & & \\
\downarrow \mathfrak{d} & & \downarrow \nabla\times & \searrow \mathfrak{d} & \downarrow \nabla^\perp & \searrow \mathfrak{d} & & & \\
H\mathfrak{L}^2 & & H(\nabla\cdot,\mathrm{Tr};\Omega^3) & & H(\nabla\cdot,\mathrm{Tr};\Omega^2) & & H(\nabla\cdot,\mathrm{Tr};\Omega^1) & & \\
\downarrow \mathfrak{d} & & \downarrow \nabla\cdot & \searrow \mathfrak{d} & \downarrow \nabla\cdot & \searrow \mathfrak{d} & \downarrow \nabla\cdot & \searrow \mathfrak{d} & \\
H\mathfrak{L}^3 & & L^2(\Omega^3) & & L^2(\Omega^2) & & L^2(\Omega^1) & & L^2(\Omega^0)
\end{array} \tag{2.5}$$

Here, $\nabla\cdot$ denotes the divergence tangential to each manifold Ω_i, regardless of dimension. The curl is denoted by $\nabla\times$ in three dimensions and is given by the rotated gradient $\nabla^\perp$ on two-dimensional manifolds. The operator ∇ at the top of the diagram represents the gradient on three-dimensional subdomains.

The local function spaces are given by subspaces of conventional Sobolev spaces with extra trace regularity. By defining ν as the outward, unit normal vector on $\partial\Omega_i$ for $i \in I$, we define

$$\begin{aligned}
H(\nabla\cdot,\mathrm{Tr};\Omega_i) &:= \{a_i \in H(\nabla\cdot,\Omega_i) : && (\nu\cdot a_i)|_{\partial_j\Omega_i} \in L^2(\Omega_{i_j}), && \forall j \in I_i^{d_i-1}\},\\
H(\nabla\times,\mathrm{Tr};\Omega_i) &:= \{a_i \in H(\nabla\times,\Omega_i) : && (-\nu\times a_i)|_{\partial_j\Omega_i} \in H(\nabla\cdot,\mathrm{Tr};\Omega_{i_j}), && \forall j \in I_i^{d_i-1}\},\\
H(\nabla^\perp,\mathrm{Tr};\Omega_i) &:= \{a_i \in H(\nabla^\perp,\Omega_i) : && (\nu^\perp a_i)|_{\partial_j\Omega_i} \in H(\nabla\cdot,\mathrm{Tr};\Omega_{i_j}), && \forall j \in I_i^{d_i-1}\},\\
H(\nabla,\mathrm{Tr};\Omega_i) &:= \{a_i \in H(\nabla,\Omega_i) : && (a_i)|_{\partial_j\Omega_i} \in H(\nabla\times,\mathrm{Tr};\Omega_{i_j}), && \forall j \in I_i^{d_i-1}\}.
\end{aligned}$$

The spaces in diagram (2.5) are then defined as the product of these spaces over all $i \in I^d$ for a given dimension d.

3 Regular Decomposition

The aim of this section is to show that the conventional regular decomposition of differential k-forms can be generalized to the mixed-dimensional setting. For that purpose, we first recall the fixed-dimensional regular decomposition in the continuous case. Then, we introduce the subspace of $H\Lambda^k$, that contains functions with higher regularity, and the analogous subspace of $H\mathfrak{L}^k$. In turn, this gives the main ingredients in the derivation of the mixed-dimensional regular decomposition.

3.1 Fixed-dimensional Regular Decomposition

We start with presenting the regular decomposition in the context of the de Rham complex (2.1). Given Ω_i with $d_i = 3$, we follow the results in [16] and provide the regular decomposition of $H(\nabla\cdot, \Omega_i)$ and $H(\nabla\times, \Omega_i)$ in the following theorems.

Theorem 3.1 (Regular decomposition of $H(\nabla\times, \Omega_i)$)**.** *For any $\boldsymbol{q} \in H(\nabla\times, \Omega_i)$, there exist functions $\boldsymbol{a} \in (H(\nabla, \Omega_i))^3$ and $c \in H(\nabla, \Omega_i)$ such that*

$$\boldsymbol{q} = \boldsymbol{a} + \nabla c, \tag{3.1a}$$

$$\|\boldsymbol{a}\|_{H(\nabla,\Omega_i)} + \|c\|_{H(\nabla,\Omega_i)} \lesssim \|\boldsymbol{q}\|_{H(\nabla\times,\Omega_i)}. \tag{3.1b}$$

Theorem 3.2 (Regular decomposition of $H(\nabla\cdot, \Omega_i)$)**.** *For any $\boldsymbol{q} \in H(\nabla\cdot, \Omega_i)$, there exist functions $\boldsymbol{a}, \boldsymbol{c} \in (H(\nabla, \Omega_i))^3$ such that*

$$\boldsymbol{q} = \boldsymbol{a} + \nabla \times \boldsymbol{c}, \tag{3.2a}$$

$$\|\boldsymbol{a}\|_{H(\nabla,\Omega_i)} + \|\boldsymbol{c}\|_{H(\nabla,\Omega_i)} \lesssim \|\boldsymbol{q}\|_{H(\nabla\cdot,\Omega_i)}. \tag{3.2b}$$

Now, let $H_0(\nabla\times, \Omega_i)$ and $H_0(\nabla\cdot, \Omega_i)$ be the subspaces of $H(\nabla\cdot, \Omega_i)$ and $H(\nabla\times, \Omega_i)$, respectively, with zero trace on the boundary $\partial\Omega_i$. Also, denote the vector function space $\boldsymbol{H}_0(\nabla, \Omega_i) = \{\boldsymbol{u} \in (H(\nabla, \Omega_i))^3, \boldsymbol{u}|_{\partial\Omega_i} = \boldsymbol{0}\}$. The "boundary aware" regular decompositions from [15] are given in the following theorems.

Theorem 3.3 (Regular decomposition of $H_0(\nabla\times, \Omega_i)$)**.** *For any $\boldsymbol{q} \in H_0(\nabla\times, \Omega_i)$, there exist functions $\boldsymbol{a} \in \boldsymbol{H}_0(\nabla, \Omega_i)$ and $c \in H_0(\nabla, \Omega_i)$ such that*

$$\boldsymbol{q} = \boldsymbol{a} + \nabla c, \tag{3.3a}$$

$$\|\boldsymbol{a}\|_{\boldsymbol{H}_0(\nabla,\Omega_i)} + \|c\|_{H_0(\nabla,\Omega_i)} \lesssim \|\boldsymbol{q}\|_{H_0(\nabla\times,\Omega_i)}. \tag{3.3b}$$

Theorem 3.4 (Regular decomposition of $H_0(\nabla\cdot, \Omega_i)$)**.** *For any $\boldsymbol{q} \in H_0(\nabla\cdot, \Omega_i)$, there exist functions $\boldsymbol{a}, \boldsymbol{c} \in \boldsymbol{H}_0(\nabla, \Omega_i)$ such that*

$$\boldsymbol{q} = \boldsymbol{a} + \nabla \times \boldsymbol{c}, \tag{3.4a}$$

$$\|\boldsymbol{a}\|_{\boldsymbol{H}_0(\nabla,\Omega_i)} + \|\boldsymbol{c}\|_{\boldsymbol{H}_0(\nabla,\Omega_i)} \lesssim \|\boldsymbol{q}\|_{H_0(\nabla\cdot,\Omega_i)}. \tag{3.4b}$$

Our derivation of the mixed-dimensional regular decomposition relies on the fact that the regular decompositions in above theorems are possible on the individual sub-manifolds Ω_i and then combined together, by taking special care of the traces.

3.2 Mixed-dimensional Regular Decomposition

For $i \in I$, let us first introduce the subspace of k-forms with increased regularity, denoted by $H^1\Lambda^k(\Omega_i) \subseteq H\Lambda^k(\Omega_i)$ such that

$$H^1\Lambda^k(\Omega_i) \cong (H(\nabla, \Omega_i))^{C_{d_i,k}}.$$

Here the notation $\cong$ means that the spaces are isomorphic. The exponent is given by the binomial coefficient $C_{d_i,k} := \binom{d_i}{k}$, which is the dimension of the space of differential k-forms on a d_i-manifold (see e.g. [27], Thm 4-5). We consider the space as zero if the exponent is zero, e.g. if $k > d_i$.

With the local spaces defined, let $H^1\mathfrak{L}^k \subseteq H\mathfrak{L}^k$ be the space of regular mixed-dimensional k-forms, given by

$$H^1\mathfrak{L}^k := \prod_{i\in I}\{a_i \in H^1\Lambda^{k_i}(\Omega_i) : \mathrm{Tr}_j a_i \in H^1\Lambda^{k_i}(\Omega_{i_j}),\ \forall j \in I_i\}, \tag{3.5}$$

and endowed with the norm

$$\|\mathfrak{a}\|_{H^1\mathfrak{L}^k} := \sum_{i\in I}\|a_i\|_{H^1\Lambda^{k_i}(\Omega_i)} + \sum_{j\in I_i}\|\mathrm{Tr}_j a_i\|_{H^1\Lambda^{k_i}(\Omega_{i_j})}. \tag{3.6}$$

We note two properties of the space $H^1\mathfrak{L}^k$. First, in the special case of $i \in I^{n-k}$, we have $k_i = 0$. Since $H^1\Lambda^0(\Omega_i) = H\Lambda^0(\Omega_i)$, it follows that

$$\iota_i H^1\mathfrak{L}^k = \iota_i H\mathfrak{L}^k, \qquad \forall i \in I^{n-k}. \tag{3.7a}$$

Secondly, the jump operator preserves the increased regularity of $H^1\mathfrak{L}^k$:

$$\mathbb{d}H^1\mathfrak{L}^k \subseteq H^1\mathfrak{L}^{k+1}. \tag{3.7b}$$

The following lemma provides the local regular decompositions on each Ω_i by using the known results in fixed-dimensional setting in Section 3.1.

Lemma 3.1. *Given $q_i \in \iota_i H\mathfrak{L}^k$ with $i \in I$, then there exists a pair $(a_i, c_i) \in \iota_i H^1\mathfrak{L}^k \times \iota_i H^1\mathfrak{L}^{k-1}$ such that*

$$q_i = a_i + \mathrm{d}c_i \qquad \text{and} \qquad \|a_i\|_{H^1\mathfrak{L}^k} + \|c_i\|_{H^1\mathfrak{L}^{k-1}} \lesssim \|q_i\|_{H\mathfrak{L}^k}.$$

Proof. We consider the four possible cases for $n \leq 3$. With reference to diagram (2.5), these cases are represented by the main diagonal (Case A) and the off-diagonal components in the three bottom rows (Cases B–D).

Case A: $k = n - d_i$. We note that this means that $k_i = 0$ and $\iota_i H\mathfrak{L}^k = \iota_i H^1\mathfrak{L}^k$ by (3.7a). Setting $a_i := q_i$ yields the result.

Case B: $k = n,\ d_i > 0$. In this case, we have $q_i \in L^2(\Omega_i)$. We introduce $a_i \in H^1_0(\Omega_i)$ as the solution to the following minimization problem,

$$\min_{a_i\in H^1_0(\Omega_i)} \tfrac{1}{2}\|a_i\|^2_{1,\Omega_i} \qquad \text{subject to } \Pi_{\mathbb{R},i}a_i = \Pi_{\mathbb{R},i}q_i, \tag{3.8a}$$

with $\Pi_{\mathbb{R},i}$ denoting the L^2-projection onto constants on Ω_i. Secondly, we define the bounded $c_i \in (H^1_0(\Omega_i))^{d_i}$ such that

$$\nabla\cdot c_i = (I - \Pi_{\mathbb{R},i})(q_i - a_i). \tag{3.8b}$$

Since the divergence represents the exterior derivative d in this case, it follows that $q_i = a_i + \mathrm{d}c_i$ with $a_i \in \iota_i H^1\mathfrak{L}^k$ and $c_i \in \iota_i H^1\mathfrak{L}^{k-1}$. Importantly, a_i and c_i have zero trace on $\partial\Omega_i$. This property will be advantageous in the remaining cases.

Case C: $k = n-1,\ d_i > 1$. In this case, we require L^2 regularity of traces on manifolds of codimension one. For $j \in I_i^{d_i-1}$, let us denote

$$q_j := \mathrm{Tr}_j q_i.$$

By definition of $H\mathfrak{L}^{n-1}$, we have that $q_j \in L^2(\Omega_{i_j}) = L^2(\partial_j\Omega_i)$. Since $\partial_j\Omega_i$ is a manifold of dimension $d_i - 1$, we can use Case B to find $a_j \in H^1_0(\partial_j\Omega_i)$ and $c_j \in (H^1_0(\partial_j\Omega_i))^{d_i-1}$ such that

$$q_j = a_j + \mathrm{d}c_j.$$

Note that both a_j and c_j have zero trace on the boundary of $\partial_j\Omega_i$. Hence, all a_j (respectively c_j) can be combined to form a function in $H^1(\partial\Omega_i)$ (respectively $(H^1(\partial\Omega_i))^{d_i-1}$). These boundary functions are extended harmonically into Ω_i to form $a_i^* \in \iota_i H^1\mathfrak{L}^k$ and $c_i^* \in \iota_i H^1\mathfrak{L}^{k-1}$ such that

$$\mathrm{Tr}_j a_i^* = a_j, \qquad \mathrm{Tr}_j c_i^* = c_j, \qquad \forall j \in I_i^{d_i-1}$$

The regularity of these extensions in the domain Ω_i is a result of the fact that all a_j and c_j are zero at tips and reentrant corners.

Next, we note that $q_i - (a_i^* + \mathrm{d}c_i^*)$ has zero trace on $\partial_j\Omega_i$ for all $j \in I_i^{d_i-1}$. Hence, we apply a regular decomposition respecting homogeneous boundary conditions to obtain

$$q_i - (a_i^* + \mathrm{d}c_i^*) = a_i^0 + \mathrm{d}c_i^0,$$

such that $a_i^0 \in H^1\Lambda^{k_i}(\Omega_i)$ and $c_i^0 \in H^1\Lambda^{k_i-1}(\Omega_i)$ have zero trace on each $\partial_j\Omega_i$ with $j \in I_i^{d_i-1}$. It follows that $a_i^0 \in \iota_i H^1\mathfrak{L}^k$ and $c_i^0 \in \iota_i H^1\mathfrak{L}^{k-1}$. We conclude the construction by setting $a_i := a_i^* + a_i^0$ and $c_i := c_i^* + c_i^0$.

Case D: $k = n-2$, $d_i > 2$. The only case not covered so far is $d_i = n = 3$. Following the same arguments as above, we first denote $q_j := \mathrm{Tr}_j q_i$ and then use the construction from Case C to obtain

$$q_j = a_j + \mathrm{d}c_j, \qquad j \in I_i^2.$$

Next, without loss of generality, we let $j_1, j_2 \in I_i^2$ and consider the index $j_{1,2} \in I_i^1$ such that $\partial_{j_{1,2}}\Omega_i$ forms a one-dimensional interface between $\partial_{j_1}\Omega_i$ and $\partial_{j_2}\Omega_i$. It follows that

$$\mathrm{Tr}_{j_{1,2}} a_{j_1} = \mathrm{Tr}_{j_{1,2}} a_{j_2} \qquad \mathrm{Tr}_{j_{1,2}} c_{j_1} = \mathrm{Tr}_{j_{1,2}} c_{j_2}$$

since both traces are equal to the unique constructions on $\partial_{j_{1,2}}\Omega_i$ from Case B, in particular (3.8). This means that, when there are more interfaces, by combining all a_j with $j \in I_i^2$, a function is formed in $(H^1(\partial\Omega_i))^2$. Analogously, the combination of all c_j with $j \in I_i^2$ forms a function in $H^1(\partial\Omega_i)$.

The construction is finalized in the same way as in Case C. In short, we first introduce a harmonic extension of the boundary functions to form $a_i^* \in \iota_i H^1\mathfrak{L}^k$ and $c_i^* \in \iota_i H^1\mathfrak{L}^{k-1}$. Then, $a_i^0 \in \iota_i H^1\mathfrak{L}^k$ and $c_i^0 \in \iota_i H^1\mathfrak{L}^{k-1}$ are constructed using a regular decomposition of $q_i - (a_i^* + \mathrm{d}c_i^*)$ respecting homogeneous boundary conditions. Finally, we set $a_i := a_i^* + a_i^0$ and $c_i := c_i^* + c_i^0$.

□

Remark 3.1. *In the previous lemma, we have frequently used the fact that operators* Tr *and* d *commute. The derivation of this property for functions on each Ω_i is straightforward so, to shorten the presentation, we leave it out and use the commutative property in the proofs when needed.*

Now we are ready to present the main result of this section, namely the mixed-dimensional regular decomposition, in the following theorem.

Theorem 3.5 (Mixed-dimensional Regular Decomposition)**.** *Given $\mathfrak{q} \in H\mathfrak{L}^k$, then there exists a pair $(\mathfrak{a}, \mathfrak{c}) \in H^1\mathfrak{L}^k \times H^1\mathfrak{L}^{k-1}$ such that*

$$\mathfrak{q} = \mathfrak{a} + \mathfrak{d}\mathfrak{c} \qquad \textit{and} \qquad \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} + \|\mathfrak{c}\|_{H^1\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{q}\|_{H\mathfrak{L}^k}$$

Proof. Given k, we construct $\mathfrak{a} = (a_i)_{i\in I}$ and $\mathfrak{c} = (c_i)_{i\in I}$ by marching through the corresponding row in diagram (2.5). We initialize both functions as $\mathfrak{a} = 0$ and $\mathfrak{c} = 0$, and redefine each component according to the following four, sequential steps.

1. If $k > 0$, consider $i \in I^n$. Lemma 3.1 gives us $a_i \in \iota_i H^1\mathfrak{L}^k$ and $c_i \in \iota_i H^1\mathfrak{L}^{k-1}$ such that

$$q_i = a_i + \mathrm{d}c_i. \tag{3.9a}$$

We repeat the above for all $i \in I^n$ and continue with step 2.

2. If $k > 1$, consider $i \in I^{n-1}$. We use Lemma 3.1 to define $\tilde{a}_i \in \iota_i H^1\mathfrak{L}^k$ and $c_i \in \iota_i H^1\mathfrak{L}^{k-1}$ such that

$$q_i = \tilde{a}_i + \mathrm{d}c_i.$$

Noting that $(q_i - \mathrm{d}c_i) \in \iota_i H^1\mathfrak{L}^k$ and using (3.7b), we set

$$a_i := q_i - \mathrm{d}c_i - \iota_i \mathfrak{d}\mathfrak{c}. \tag{3.9b}$$

where $\mathfrak{c}$ has non-zero components c_j, $j \in I^n$ defined in step 1. We repeat this construction for all $i \in I^{n-1}$ and continue with step 3.

3. If $k > 2$, repeat step 2 with $i \in I^{n-2}$ and continue with step 4.

4. In any case, consider $i \in I^{n-k}$. We have $k_i = 0$ and note that $\iota_i H\mathfrak{L}^k = \iota_i H^1\mathfrak{L}^k$ from (3.7a). Hence, we use (3.7b) to set

$$a_i := q_i - \iota_i \mathfrak{d}\mathfrak{c}. \tag{3.9c}$$

where $\mathfrak{c}$ has non-zero components defined in steps 1–3. This construction is repeated for all $i \in I^{n-k}$.

The four steps give us $\mathfrak{a} := (a_i)_{i\in I} \in H^1\mathfrak{L}^k$ and $\mathfrak{c} := (c_i)_{i\in I} \in H^1\mathfrak{L}^{k-1}$ and we collect (3.9) to conclude

$$\mathfrak{q} = \mathfrak{a} + \mathfrak{d}\mathfrak{c}.$$

The bound follows by the construction and Lemma 3.1. □

A byproduct of the mixed-dimensional regular decomposition is the so-called regular inverse of the mixed-dimensional differential $\mathfrak{d}$ as shown in the following corollary.

Corollary 3.1 (Mixed-dimensional Regular Inverse)**.** *Given $\mathfrak{q} \in H\mathfrak{L}^k$, then there exists $\mathfrak{a} \in H^1\mathfrak{L}^k$ such that*

$$\mathfrak{d}(\mathfrak{q} - \mathfrak{a}) = 0 \qquad \textit{and} \qquad \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{q}\|_{H\mathfrak{L}^k}$$

Proof. Follows from Theorem 3.5 and the fact that $\mathfrak{d}\mathfrak{d}\mathfrak{c} = 0$ from (2.3). □

4 Discrete Regular Decomposition

In this section, we introduce the discrete version of the regular decomposition (Theorem 3.5). To this end, let h be the typical mesh size and the subscript h describe discrete entities. We first present the conventional discrete regular decomposition in a fixed-dimensional setting. Then, we introduce the structure-preserving discretization of the mixed-dimensional geometry Ω and function spaces $H\mathfrak{L}^k$. We finalize the section with deriving the discrete mixed-dimensional regular decomposition.

4.1 Fixed-dimensional Discrete Regular Decomposition

Let $H_h(\nabla, \Omega_i)$, $H_h(\nabla\times, \Omega_i)$, and $H_h(\nabla\cdot, \Omega_i)$ denote the conforming finite element approximations of the functions spaces $H(\nabla, \Omega_i)$, $H(\nabla\times, \Omega_i)$ and $H(\nabla\cdot, \Omega_i)$. In addition, let $\Pi_h^{\nabla\times} : H(\nabla\times, \Omega_i) \to H_h(\nabla\times, \Omega_i)$ and $\Pi_h^{\nabla\cdot} : H(\nabla\cdot, \Omega_i) \to H_h(\nabla\cdot, \Omega_i)$ be the stable projection operators. In connection to Section 3.1, the discrete analogues of Theorem 3.1 and Theorem 3.2 are given below.

Theorem 4.1 (Regular decomposition of $H_h(\nabla\times, \Omega_i)$)**.** *For any $\boldsymbol{q}_h \in H_h(\nabla\times, \Omega_i)$, there exist vector functions $\boldsymbol{a}_h \in (H_h(\nabla, \Omega_i))^3$, $\boldsymbol{b}_h \in H_h(\nabla\times, \Omega_i)$ and a scalar function $c_h \in H_h(\nabla, \Omega_i)$ such that*

$$\boldsymbol{q}_h = \Pi_h^{\nabla\times}\boldsymbol{a}_h + \boldsymbol{b}_h + \nabla c_h, \tag{4.1a}$$

$$\|\Pi_h^{\nabla\times}\boldsymbol{a}_h\|_{H(\nabla\times,\Omega_i)} + \|h^{-1}\boldsymbol{b}_h\|_{L^2(\Omega_i)} + \|c_h\|_{H(\nabla,\Omega_i)} \lesssim \|\boldsymbol{q}_h\|_{H(\nabla\times,\Omega_i)}. \tag{4.1b}$$

Theorem 4.2 (Regular decomposition of $H_h(\nabla\cdot,\Omega_i)$)**.** *For any $\boldsymbol{q}_h \in H_h(\nabla\cdot,\Omega_i)$, there exist vector functions $\boldsymbol{a}_h, \boldsymbol{e}_h \in (H_h(\nabla,\Omega_i))^3$, $\boldsymbol{b}_h \in H_h(\nabla\cdot,\Omega_i)$ and $\boldsymbol{f}_h \in H_h(\nabla\times,\Omega_i)$ such that*

$$\boldsymbol{q}_h = \Pi_h^{\nabla\cdot}\boldsymbol{a}_h + \boldsymbol{b}_h + \nabla\times(\Pi_h^{\nabla\times}\boldsymbol{e}_h + \boldsymbol{f}_h), \tag{4.2a}$$

$$\|\Pi_h^{\nabla\cdot}\boldsymbol{a}_h\|_{H(\nabla\cdot,\Omega_i)} + \|h^{-1}\boldsymbol{b}_h\|_{L^2(\Omega_i)} + \|\boldsymbol{e}_h\|_{H(\nabla\times,\Omega_i)} + \|h^{-1}\boldsymbol{f}_h\|_{L^2(\Omega_i)} \lesssim \|\boldsymbol{q}_h\|_{H(\nabla\cdot,\Omega_i)}. \tag{4.2b}$$

These discrete regular decompositions reveal the structure that we aim to preserve in the mixed-dimensional setting. Specifically, the stability of the decompositions in the sense of bounds (4.1b) and (4.2b) will in turn provide us with robust preconditioners by the theory of the auxiliary space methods. We give a short overview of the auxiliary space preconditioning theory later in Section 5.1 and focus first on the derivation of the mixed-dimensional analogue to Theorem 4.1 and Theorem 4.2.

4.2 Mixed-dimensional Discretization

First, we introduce a shape-regular simplicial partition of Ω, denoted by $\Omega_h = \bigcup_{i\in I}\Omega_{i,h}$. The grid is constructed such that it conforms to all lower-dimensional manifolds and all grids are matching. In order to preserve the regular decomposition on the discrete level, structure preserving discretization in the mixed-dimensional setting should be considered. Let us introduce $H_h\mathfrak{L}^k$ as the discretization of $H\mathfrak{L}^k$ defined on Ω_h. Using the notation of finite element exterior calculus [4], we consider the family of reduced finite elements (i.e. elements of the first kind) and set

$$H_h\mathfrak{L}^k = \prod_{i\in I} P_r^-\Lambda^{k_i}(\Omega_{i,h}). \tag{4.3}$$

The lowest-order case ($r = 1$) in the three-dimensional setting ($n = 3$) gives us the following, discrete de Rham complex:

$$\begin{array}{ccccccccc}
H_h\mathfrak{L}^0 & & \mathbb{P}_1(\Omega_h^3) & & & & & & \\
\downarrow\mathfrak{d} & & \downarrow\nabla & \searrow^{\mathfrak{d}} & & & & & \\
H_h\mathfrak{L}^1 & & \mathbb{N}_0^e(\Omega_h^3) & & \mathbb{P}_1(\Omega_h^2) & & & & \\
\downarrow\mathfrak{d} & & \downarrow\nabla\times & \searrow^{\mathfrak{d}} & \downarrow\nabla^\perp & \searrow^{\mathfrak{d}} & & & \\
H_h\mathfrak{L}^2 & & \mathbb{N}_0^f(\Omega_h^3) & & \mathbb{RT}_0(\Omega_h^2) & & \mathbb{P}_1(\Omega_h^1) & & \\
\downarrow\mathfrak{d} & & \downarrow\nabla\cdot & \searrow^{\mathfrak{d}} & \downarrow\nabla\cdot & \searrow^{\mathfrak{d}} & \downarrow\nabla\cdot & \searrow^{\mathfrak{d}} & \\
H_h\mathfrak{L}^3 & & \mathbb{P}_0(\Omega_h^3) & & \mathbb{P}_0(\Omega_h^2) & & \mathbb{P}_0(\Omega_h^1) & & \mathbb{P}_0(\Omega_h^0)
\end{array} \tag{4.4}$$

Here, $\mathbb{P}_1$, $\mathbb{RT}_0$, and $\mathbb{P}_0$ denote linear Lagrange, lowest-order Raviart-Thomas, and piecewise constant finite element spaces, respectively. $\mathbb{N}_0^e$ and $\mathbb{N}_0^f$ represent the edge-based and face-based Nédélec elements of lowest order, respectively.

We introduce the stable projection operators $\Pi_h^k : H\mathfrak{L}^k \mapsto H_h\mathfrak{L}^k$ such that

$$\|(I - \Pi_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} \lesssim h\|\mathfrak{a}\|_{H^1\mathfrak{L}^k}, \quad \forall\,\mathfrak{a} \in H^1\mathfrak{L}^k, \tag{4.5}$$

and the following diagram commutes,

$$\begin{array}{ccccccc}
H\mathfrak{L}^0 & \xrightarrow{\mathfrak{d}} & H\mathfrak{L}^1 & \xrightarrow{\mathfrak{d}} & H\mathfrak{L}^2 & \xrightarrow{\mathfrak{d}} & H\mathfrak{L}^3 \\
\downarrow\Pi_h^0 & & \downarrow\Pi_h^1 & & \downarrow\Pi_h^2 & & \downarrow\Pi_h^3 \\
H_h\mathfrak{L}^0 & \xrightarrow{\mathfrak{d}} & H_h\mathfrak{L}^1 & \xrightarrow{\mathfrak{d}} & H_h\mathfrak{L}^2 & \xrightarrow{\mathfrak{d}} & H_h\mathfrak{L}^3
\end{array} \tag{4.6}$$

In the mixed-dimensional setting, such a bounded projection can be constructed by combining fixed-dimensional cochain projections on each Ω_i together. For the construction of fixed-dimensional cochain projections, we refer to [11].

Lemma 4.1 (Exactness)**.** *Given the commuting projection operators Π_h^k exist, then all discrete closed forms are exact. Thus, for each $\mathfrak{q}_h \in H_h\mathfrak{L}^k$ with $\mathfrak{d}\mathfrak{q}_h = 0$, there exists a $\mathfrak{c}_h \in H_h\mathfrak{L}^{k-1}$ such that*

$$\mathfrak{d}\mathfrak{c}_h = \mathfrak{q}_h.$$

Proof. Assume $\mathfrak{q}_h \in H_h\mathfrak{L}^k$ with $\mathfrak{d}\mathfrak{q}_h = 0$ given. Since $H_h\mathfrak{L}^k \subset H\mathfrak{L}^k$, we use the exactness of the mixed-dimensional De Rham complex (2.4) to find $\mathfrak{c} \in H\mathfrak{L}^{k-1}$ such that

$$\mathfrak{d}\mathfrak{c} = \mathfrak{q}_h.$$

Setting $\mathfrak{c}_h := \Pi_h^{k-1}\mathfrak{c}$, we obtain

$$\mathfrak{d}\mathfrak{c}_h = \mathfrak{d}\Pi_h^{k-1}\mathfrak{c} = \Pi_h^k\mathfrak{d}\mathfrak{c} = \Pi_h^k\mathfrak{q}_h = \mathfrak{q}_h,$$

which completes the proof. □

4.3 Mixed-dimensional Discrete Regular Decomposition

This section is devoted to deriving the discrete regular decomposition, i.e. the discrete analogue to Theorem 3.5. We first require the following preparatory lemma.

Lemma 4.2. *Given $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, then there exists a semi-discrete pair $(\mathfrak{a}, \mathfrak{f}_h) \in H^1\mathfrak{L}^k \times H_h\mathfrak{L}^{k-1}$ such that*

$$\mathfrak{q}_h = \Pi_h^k\mathfrak{a} + \mathfrak{d}\mathfrak{f}_h \qquad \text{and} \qquad \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} + \|\mathfrak{f}_h\|_{H\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}.$$

Proof. Since $\mathfrak{q}_h \in H_h\mathfrak{L}^k \subset H\mathfrak{L}^k$, we use Corollary 3.1 to construct $\mathfrak{a} \in H^1\mathfrak{L}^k$ such that

$$\mathfrak{d}(\mathfrak{q}_h - \mathfrak{a}). = 0$$

Next, we use $\mathfrak{d}H_h\mathfrak{L}^k \subseteq H_h\mathfrak{L}^{k+1}$ and the commutativity of the projection operators to derive

$$\mathfrak{d}\mathfrak{q}_h = \Pi_h^{k+1}\mathfrak{d}\mathfrak{q}_h = \Pi_h^{k+1}\mathfrak{d}\mathfrak{a} = \mathfrak{d}\Pi_h^k\mathfrak{a}.$$

We thus have $\mathfrak{d}(\mathfrak{q}_h - \Pi_h^k\mathfrak{a}) = 0$, i.e. $\mathfrak{q}_h - \Pi_h^k\mathfrak{a}$ is a closed form in $H_h\mathfrak{L}^k$. From Lemma 4.1, a $\mathfrak{f}_h \in H_h\mathfrak{L}^{k-1}$ exists such that

$$\mathfrak{d}\mathfrak{f}_h = \mathfrak{q}_h - \Pi_h^k\mathfrak{a}.$$

By Corollary 3.1, we have $\|\mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}$. In addition, by (4.5), we have

$$\begin{aligned}\|\mathfrak{d}\mathfrak{f}_h\|_{L^2\mathfrak{L}^k} &\lesssim \|\mathfrak{q}_h\|_{L^2\mathfrak{L}^k} + \|\Pi_h^k\mathfrak{a}\|_{L^2\mathfrak{L}^k} \leq \|\mathfrak{q}_h\|_{L^2\mathfrak{L}^k} + \|\mathfrak{a}\|_{L^2\mathfrak{L}^k} + \|(I - \Pi_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} \\ &\lesssim \|\mathfrak{q}_h\|_{L^2\mathfrak{L}^k} + \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}.\end{aligned}$$

Since the choice of $\mathfrak{f}_h$ is not unique, we choose a special $\mathfrak{f}_h$ such that the following Poincaré inequality holds

$$\|\mathfrak{f}_h\|_{H\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{d}\mathfrak{f}_h\|_{L^2\mathfrak{L}^k}.$$

Then we have $\|\mathfrak{f}_h\|_{H\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}$, which completes the proof. □

To further develop the decomposition in the discrete setting, let us introduce $H_h^1\mathfrak{L}^k \subseteq H^1\mathfrak{L}^k$ as the discretization of the regular k-forms from (3.5). For the given choice of discrete spaces (4.3), the regular spaces are given by

$$H_h^1\mathfrak{L}^k = \prod_{i\in I}(P_r^-\Lambda^0(\Omega_{i,h}))^{C_{d_i,k_i}}.$$

Again, the exponent is given by $C_{d_i,k_i} := \binom{d_i}{k_i}$. In the lowest-order case with $r = 1$, this means that

$$H_h^1\mathfrak{L}^k = \prod_{i\in I}(\mathbb{P}_1(\Omega_{i,h}))^{C_{d_i,k_i}}.$$

In other words, all discrete forms with increased regularity are given by (tuples of) nodal Lagrange elements. Similar to (3.7), we note that

$$\iota_i H_h^1 \mathfrak{L}^k = \iota_i H_h \mathfrak{L}^k, \qquad i \in I^{n-k}, \tag{4.7a}$$

$$\mathrm{d} H_h^1 \mathfrak{L}^k \subseteq H_h^1 \mathfrak{L}^{k+1}. \tag{4.7b}$$

Let $\mathcal{P}_h^k$ be the projection operator onto the discretized space of regular k-forms. $\mathcal{P}_h^k : H^1\mathfrak{L}^k \mapsto H_h^1\mathfrak{L}^k$ is stable and has the following property:

$$\|\mathcal{P}_h^k \mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{a}\|_{H^1\mathfrak{L}^k}, \qquad \|(I - \mathcal{P}_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} \lesssim h\|\mathfrak{a}\|_{H^1\mathfrak{L}^k}, \tag{4.8}$$

for all $\mathfrak{a} \in H^1\mathfrak{L}^k$. Now we are ready to present the mixed-dimensional discrete regular decomposition as follows.

Theorem 4.3 (Mixed-dimensional Discrete Regular Decomposition)**.** *Given $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, then there exists a pair $(\mathfrak{a}_h, \mathfrak{f}_h) \in H_h^1\mathfrak{L}^k \times H_h\mathfrak{L}^{k-1}$ and a high-frequency term $\mathfrak{b}_h \in H_h\mathfrak{L}^k$ such that*

$$\mathfrak{q}_h = \Pi_h^k \mathfrak{a}_h + \mathfrak{b}_h + \mathfrak{d}\mathfrak{f}_h \qquad \textit{and} \qquad \|\mathfrak{a}_h\|_{H^1\mathfrak{L}^k} + \|h^{-1}\mathfrak{b}_h\|_{L^2\mathfrak{L}^k} + \|\mathfrak{f}_h\|_{H\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}.$$

Proof. Given the decomposition from Lemma 4.2, we further decompose $\mathfrak{a}$ using the projection operator $\mathcal{P}_h^k$ from (4.8):

$$\mathfrak{q}_h = \Pi_h^k \mathcal{P}_h^k \mathfrak{a} + \Pi_h^k (I - \mathcal{P}_h^k)\mathfrak{a} + \mathfrak{d}\mathfrak{f}_h.$$

By defining $\mathfrak{a}_h := \mathcal{P}_h^k \mathfrak{a}$ and $\mathfrak{b}_h := \Pi_h^k (I - \mathcal{P}_h^k)\mathfrak{a}$, we obtain the desired format. To prove the boundedness, we use the stability of $\mathcal{P}_h^k$ and approximation properties (4.5) and (4.8) to derive

$$\begin{aligned}
\|\mathfrak{a}_h\|_{H^1\mathfrak{L}^k} &= \|\mathcal{P}_h^k \mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}, \\
\|h^{-1}\mathfrak{b}_h\|_{L^2\mathfrak{L}^k} &= \|h^{-1}\Pi_h^k (I - \mathcal{P}_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} \lesssim \|h^{-1}(I - \mathcal{P}_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} + \|h^{-1}(I - \Pi_h^k)(I - \mathcal{P}_h^k)\mathfrak{a}\|_{L^2\mathfrak{L}^k} \\
&\lesssim \|\mathfrak{a}\|_{H^1\mathfrak{L}^k} + \|(I - \mathcal{P}_h^k)\mathfrak{a}\|_{H^1\mathfrak{L}^k} \lesssim \|\mathfrak{a}\|_{H^1\mathfrak{L}^k}.
\end{aligned}$$

Combining these estimates with the bound of Lemma 4.2 proves the result. □

In practice, it is useful to integrate discrete regular decompositions so that all the components have improved regularity except the high-frequency parts. Such a result is shown as follows.

Corollary 4.1 (Integrated Mixed-dimensional Discrete Regular Decomposition)**.** *Given $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, then there exist a regular pair $(\mathfrak{a}_h, \mathfrak{c}_h) \in H_h^1\mathfrak{L}^k \times H_h^1\mathfrak{L}^{k-1}$ and a high-frequency pair $(\mathfrak{b}_h, \mathfrak{e}_h) \in H_h\mathfrak{L}^k \times H_h\mathfrak{L}^{k-1}$ such that*

$$\Pi_h^k \mathfrak{a}_h + \mathfrak{b}_h + \mathfrak{d}(\Pi_h^{k-1}\mathfrak{c}_h + \mathfrak{e}_h) = \mathfrak{q}_h \tag{4.9a}$$

$$\|\mathfrak{a}_h\|_{H^1\mathfrak{L}^k} + \|h^{-1}\mathfrak{b}_h\|_{L^2\mathfrak{L}^k} + \|\mathfrak{c}_h\|_{H^1\mathfrak{L}^{k-1}} + \|h^{-1}\mathfrak{e}_h\|_{L^2\mathfrak{L}^{k-1}} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k}. \tag{4.9b}$$

Proof. Since $H_h^1\mathfrak{L}^0 = H_h\mathfrak{L}^0$ by (4.7a), the result is trivial for $k = 0$. We use the same argument in combination with Lemma 4.3 to conclude that the case $k = 1$ follows with $\mathfrak{e}_h = 0$.

We continue with $k > 1$. Using the decomposition from Lemma 4.3, we obtain $(\mathfrak{a}_h, \mathfrak{b}_h, \mathfrak{f}_h) \in H_h^1\mathfrak{L}^k \times H_h\mathfrak{L}^k \times H_h\mathfrak{L}^{k-1}$ such that

$$\mathfrak{q}_h = \Pi_h^k \mathfrak{a}_h + \mathfrak{b}_h + \mathfrak{d}\mathfrak{f}_h,$$

with the associated bound. Applying Lemma 4.3 once more on $\mathfrak{f}_h \in H_h\mathfrak{L}^{k-1}$, we have $(\mathfrak{c}_h, \mathfrak{e}_h, \mathfrak{g}_h) \in H_h^1\mathfrak{L}^{k-1} \times H_h\mathfrak{L}^{k-1} \times H_h\mathfrak{L}^{k-2}$ such that

$$\mathfrak{f}_h = \Pi_h^{k-1}\mathfrak{c}_h + \mathfrak{e}_h + \mathfrak{d}\mathfrak{g}_h$$

Due to (2.3), we have $\mathfrak{d}\mathfrak{d}\mathfrak{g}_h = 0$ and the result follows. □

Similarly, in the discrete case, we also have the regular inverse as the byproduct of the discrete regular decomposition, which is stated in the following corollary.

Corollary 4.2 (Discrete Regular Inverse)**.** *Given $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, then there exist $\mathfrak{a}_h \in H_h^1\mathfrak{L}^k$ and a high-frequency term $\mathfrak{b}_h \in H_h\mathfrak{L}^k$ such that*

$$\mathfrak{d}(\mathfrak{q}_h - \Pi_h^k \mathfrak{a}_h - \mathfrak{b}_h) = 0 \qquad \textit{and} \qquad \|\mathfrak{a}_h\|_{H^1\mathfrak{L}^k} + \|h^{-1}\mathfrak{b}_h\|_{L^2\mathfrak{L}^k} \lesssim \|\mathfrak{q}_h\|_{H\mathfrak{L}^k} \tag{4.10}$$

Proof. Follows from Theorem 4.1 and the fact that $\mathfrak{d}\mathfrak{d} = 0$ from (2.3). □

5 Mixed-dimensional Auxiliary Space Preconditioner

Based on the (discrete) regular decomposition, we develop robust preconditioners for solving the following abstract mixed-dimensional problem: Find $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, such that

$$(\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^k} + (\mathfrak{d}\mathfrak{q}_h, \mathfrak{d}\tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^{k+1}} = (\mathfrak{f}, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^k} \qquad \forall\, \tilde{\mathfrak{q}}_h \in H_h\mathfrak{L}^k. \tag{5.1}$$

This can be written into a linear system $\mathfrak{A}^k\mathfrak{q}_h = \mathfrak{f}$, where $\mathfrak{A}^k = \mathfrak{I} + \mathfrak{d}^*\mathfrak{d}$ is a symmetric positive definite operator on $H_h\mathfrak{L}^k$, $\mathfrak{I}$ is the identity mapping and $\mathfrak{d}^*$ is the adjoint of $\mathfrak{d}$. Our goal is to derive a preconditioner $\mathfrak{B}$ for the problem (5.1) based on the fictitious or auxiliary space preconditioning theory developed in [16, 24, 29].

5.1 Abstract Theory of Auxiliary Space Preconditioning

We recall the framework of the auxiliary space theory. Assume V is a separable Hilbert space with an inner product $a(\cdot,\cdot)$. We aim to find $u \in V$ that solves

$$a(u, v) = (f, v) \qquad \forall v \in V, \tag{5.2}$$

or equivalently

$$Au = f, \tag{5.3}$$

where $A : V \mapsto V'$ is symmetric positive definite such that $\langle Au, v\rangle = a(u, v)$. Here V' is the dual of V. Using A, the norm induced by $a(\cdot,\cdot)$ can be denoted by $\|\cdot\|_A$ and we also consider another inner product $s(\cdot,\cdot)$ on V, which induces another norm $\|\cdot\|_S$ with S being symmetric positive definite.

For designing auxiliary preconditioners, let W_ℓ, $\ell = 1, 2, \ldots, L$, be auxiliary spaces with inner products $a_\ell(\cdot,\cdot)$ that induces norms $\|\cdot\|_{A_\ell}$, where $A_\ell : W_\ell \mapsto W_\ell'$ are linear operators defined as $\langle A_\ell u_\ell, v_\ell\rangle = a_\ell(u_\ell, v_\ell)$, for $u_\ell, v_\ell \in W_\ell$, $\ell = 1, 2, \ldots, L$. In addition, we assume that there are transfer operators $\Pi_\ell : W_\ell \mapsto V$. Finally, we define the auxiliary product space $\bar{V} = V \times W_1 \times W_2 \times \cdots \times W_L$, and then represent the inner product on $\bar{V}$ as

$$\bar{a}(\bar{v}, \bar{v}) = s(v, v) + \sum_{\ell=1}^{L} a_l(w_\ell, w_\ell) \qquad \forall\, \bar{v} = (v, w_1, \ldots, w_L) \in \bar{V}.$$

Using the fictitious or auxiliary space method, the preconditioner $B : V' \mapsto V$ for the linear problem (5.3) is defined as

$$B = S^{-1} + \sum_{\ell=1}^{L} \Pi_\ell A_\ell^{-1} \Pi_\ell^*, \tag{5.4}$$

where S^{-1} is the so-called smoother operator. The following Lemma from [16], which can be viewed as a special case of the fictitious lemma [24], gives a bound on the condition number $\kappa(BA)$.

Lemma 5.1. *Assume the following conditions hold:*

1. *There exist $c_\ell > 0$, $\ell = 1, \ldots, L$ such that $\|\Pi_\ell w_\ell\|_A \le c_\ell \|w_\ell\|_{A_\ell}$, $\quad \forall\, w_\ell \in W_\ell$.*
2. *There exist $c_s > 0$ such that $\|v\|_A \le c_s \|v\|_S$, $\quad \forall\, v \in V$.*
3. *For every $v \in V$, there exists a decomposition $v = v_0 + \sum_{\ell=1}^{L} w_\ell$, $v_0 \in V$, $w_\ell \in W_\ell$ and $c_0 > 0$ such that*
$$\|v_0\|_S^2 + \sum_{\ell=1}^{L} \|w_\ell\|_{A_\ell}^2 \le c_0 \|v\|_A^2.$$

Then $\kappa(BA) \le c_0^2(c_s^2 + c_1^2 + \cdots + c_L^2)$.

If all the bounds in Lemma 5.1 are independent of discretization parameter h (and any other parameters), then B is a robust preconditioner for A in (5.3). The auxiliary space preconditioner B (5.4) can be viewed as additive version. As mentioned in [16], naturally, we can also apply auxiliary spaces successively to obtain a multiplicative auxiliary space preconditioner, which is also robust under the same conditions, we refer to [17] for details. Note that instead of directly applying operators A_ℓ^{-1}, we can replacing them by their spectrally equivalent approximations, B_ℓ. As long as the constants in the spectral equivalence are independent of physical and discretization parameters, the resulting auxiliary space preconditioners remain robust.

Remark 5.1. *For example, [16] shows that the fixed-dimensional discrete regular decomposition in Theorem 4.2 follows the conditions of Lemma 5.1. Let $V = H_h(\nabla\cdot,\Omega_i)$ and for $u, v \in V$ let the bilinear form $a(u,v) = (u,v) + (\nabla\cdot u, \nabla\cdot v)$ in* (5.2)*. The auxiliary space theory gives the following preconditioner for solving* (5.3)*. Take the auxiliary spaces $W_1 = (H_h(\nabla,\Omega_i))^3$, $W_2 = H_h(\nabla\times,\Omega_i)$, $W_3 = (H_h(\nabla,\Omega_i))^3$ and transfer operators $\Pi_1 = \Pi_h^{\nabla\cdot}$, $\Pi_2 = \nabla\times$, $\Pi_3 = \nabla\times\Pi_h^{\nabla\times}$. With certain choices of smoothers $S^{\nabla\cdot}$ and $S^{\nabla\times}$ on $H_h(\nabla\cdot,\Omega_i)$ and $H_h(\nabla\times,\Omega_i)$ (for example, Jacobi smoother), respectively, we get*

$$B = (S^{\nabla\cdot})^{-1} + \Pi_h^{\nabla\cdot} A_{reg}^{-1} (\Pi_h^{\nabla\cdot})^* + \nabla\times(S^{\nabla\times})^{-1}(\nabla\times)^* + \nabla\times\Pi_h^{\nabla\times} A_{reg}^{-1}(\nabla\times\Pi_h^{\nabla\times})^*, \tag{5.5}$$

where A_{reg} is the linear operator induced by the inner product on $(H_h(\nabla,\Omega_i))^3$. In the following section, we show that the similar preconditioner is feasible in the mixed-dimensional setting using Theorem 4.3.

5.2 Mixed-dimensional Preconditioner

Let us apply the theory in Section 5.1 on the problem (5.1) to develop the auxiliary space preconditioner in the mixed-dimensional setting. Following Theorem 4.3, for any $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, there is a pair $(\mathfrak{a}_h, \mathfrak{f}_h) \in H_h^1\mathfrak{L}^k \times H_h\mathfrak{L}^{k-1}$ and a high-frequency term $\mathfrak{b}_h \in H_h\mathfrak{L}^k$ that allows the following decomposition

$$\mathfrak{q}_h = \mathfrak{b}_h + \Pi_h^k \mathfrak{a}_h + \mathfrak{d}\mathfrak{f}_h.$$

Now, besides the original space $V = H_h\mathfrak{L}^k$, we have two auxiliary spaces $W_1 = H_h^1\mathfrak{L}^k$ and $W_2 = H_h\mathfrak{L}^{k-1}$. Furthermore, we take the transfer operator $\Pi_1 = \Pi_h^k$ restricted to $H_h^1\mathfrak{L}^k$, i.e. $\Pi_h^k : H_h^1\mathfrak{L}^k \mapsto H_h\mathfrak{L}^k$, and $\Pi_2 = \mathfrak{d}$. We write $\mathfrak{A}_{reg}^k$ for the symmetric positive definite linear operator defined by the inner product on the space $H^1\mathfrak{L}^k$, which can be viewed as (vector) Laplacian operators in the mixed-dimensional setting.

For the sake of simplicity, we consider the Jacobi smoother. For a function $\mathfrak{q}_h \in H_h\mathfrak{L}^k$, we have $\mathfrak{q}_h = \sum_{\mathfrak{e}} \mathfrak{q}_h^{\mathfrak{e}}$, where $\mathfrak{q}_h^{\mathfrak{e}} \in \operatorname{span}\{\mathfrak{e}\}$ where $\mathfrak{e}$ denotes a degree of freedom defined on either a node, edge, face or cell of Ω_h. Then the smoothing operator is characterized by the inner product

$$s(\mathfrak{q}_h, \mathfrak{q}_h) = \sum_{\mathfrak{e}} \left((\mathfrak{q}_h^{\mathfrak{e}}, \mathfrak{q}_h^{\mathfrak{e}})_{L^2\mathfrak{L}^k} + (\mathfrak{d}\mathfrak{q}_h^{\mathfrak{e}}, \mathfrak{d}\mathfrak{q}_h^{\mathfrak{e}})_{L^2\mathfrak{L}^{k+1}} \right).$$

This leads to a smoother $\mathfrak{S}^k$, which, in matrix representation, coincides with the diagonal of $\mathfrak{A}^k$.

The auxiliary space preconditioner $\mathfrak{B} : (H_h\mathfrak{L}^k)' \mapsto H_h\mathfrak{L}^k$ for (5.1) takes the following form

$$\mathfrak{B}^k = (\mathfrak{S}^k)^{-1} + \Pi_h^k (\mathfrak{A}_{reg}^k)^{-1} (\Pi_h^k)^* + \mathfrak{d}(\mathfrak{A}^{k-1})^{-1}\mathfrak{d}^*. \tag{5.6}$$

Here, * denotes the adjoint with respect to the $L^2\mathfrak{L}^k$ inner product and is the standard matrix transpose in the matrix representation.

In order to show the bound $\kappa(\mathfrak{B}^k\mathfrak{A}^k) \lesssim 1$, we need to verify the conditions in Lemma 5.1 and the results are summarized in the following theorem.

Theorem 5.1. *Using $\mathfrak{B}^k$ from* (5.6) *as a preconditioner for solving the linear system* (5.1) *leads to a condition number $\kappa(\mathfrak{B}^k\mathfrak{A}^k) \lesssim 1$, where the hidden constant depends only on Ω and shape regularity of the mesh.*

Proof. We verify the three conditions of Lemma 5.1:

1. It follows from the properties of Π_h^k and $\mathfrak{d}$ that

$$\|\Pi_h^k \mathfrak{a}_h\|^2_{H\mathfrak{L}^k} = \|\Pi_h^k \mathfrak{a}_h\|^2_{L^2\mathfrak{L}^k} + \|\mathfrak{d}\Pi_h^k \mathfrak{a}_h\|^2_{L^2\mathfrak{L}^{k+1}} \lesssim \|\mathfrak{a}_h\|^2_{H^1\mathfrak{L}^k} + \|\mathfrak{d}\mathfrak{a}_h\|^2_{L^2\mathfrak{L}^{k+1}} \lesssim \|\mathfrak{a}_h\|^2_{H^1\mathfrak{L}^k}, \tag{5.7}$$

$$\|\mathfrak{d}\mathfrak{f}_h\|_{H\mathfrak{L}^k} = \|\mathfrak{d}\mathfrak{f}_h\|_{L^2\mathfrak{L}^k} \leq \|\mathfrak{f}_h\|_{H\mathfrak{L}^{k-1}}, \tag{5.8}$$

   for any $\mathfrak{a}_h \in H_h^1\mathfrak{L}^k$ and $\mathfrak{f}_h \in H_h\mathfrak{L}^{k-1}$. This verifies the first condition of Lemma 5.1.

2. Since each element has a finite number of neighbors, there is a small constant $c_s > 0$ such that

$$(\mathfrak{q}_h, \mathfrak{q}_h)_{\mathfrak{A}^k} = \|\textstyle\sum_{\mathfrak{e}} q_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^k} + \|\sum_{\mathfrak{e}} \mathfrak{d} q_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^{k+1}} \leq c_s^2 \sum_{\mathfrak{e}} \left(\|q_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^k} + \|\mathfrak{d} q_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^{k+1}}\right) = c_s^2 s(\mathfrak{q}_h, \mathfrak{q}_h), \tag{5.9}$$

   which verifies the second condition of Lemma 5.1.

3. Lastly, we can see from Theorem 4.3 that we only need to show the bound on $\|\mathfrak{b}_h\|_S$,

$$\begin{aligned}
\|\mathfrak{b}_h\|_S^2 = \sum_{\mathfrak{e}} \|b_h^{\mathfrak{e}}\|^2_{H\mathfrak{L}^k} &= \sum_{\mathfrak{e}} \left(\|b_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^k} + \|\mathfrak{d} b_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^{k+1}}\right) \\
\text{(Inverse inequality)} &\lesssim \sum_{\mathfrak{e}} \left(\|b_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^k} + \|h^{-1} b_h^{\mathfrak{e}}\|^2_{L^2\mathfrak{L}^k}\right) \\
(L^2\text{-stability of the bases}) &\lesssim \|\mathfrak{b}_h\|^2_{L^2\mathfrak{L}^k} + \|h^{-1}\mathfrak{b}_h\|^2_{L^2\mathfrak{L}^k} \\
\text{(Theorem 4.3)} &\lesssim \|\mathfrak{q}_h\|^2_{H\mathfrak{L}^k}.
\end{aligned} \tag{5.10}$$

Therefore, by applying Lemma 5.1, we have that $\kappa(\mathfrak{B}^k\mathfrak{A}^k) \lesssim 1$. □

We note that it is possible to choose different smoother, such as Gauss-Seidel smoother. In fact, one could use any $s(\cdot,\cdot)$ that is spectral equivalent to $\|h^{-1}\cdot\|^2_{L^2\mathfrak{L}^k} + \|\cdot\|^2_{L^2\mathfrak{L}^k}$.

Finally, we can integrate the regular decomposition into the preconditioner by utilizing Corollary 4.1 to further expand $\mathfrak{B}^k$. This will be especially useful in Section 6 when designing a preconditioner for a parameter-dependent saddle point problem in practice. Similarly as before, we set $V = H_h\mathfrak{L}^k$, $W_1 = H_h^1\mathfrak{L}^k$, $W_2 = H_h\mathfrak{L}^{k-1}$ and $W_3 = H_h^1\mathfrak{L}^{k-1}$. The transfer operators are then $\Pi_1 = \Pi_h^k$ restricted to $H_h^1\mathfrak{L}^k$, $\Pi_2 = \mathfrak{d}$ and $\Pi_3 = \mathfrak{d}\Pi_h^{k-1}$ restricted to $H_h^1\mathfrak{L}^{k-1}$. Again, we still use Jacobi smoother for the sake of simplicity here. The preconditioner $\mathfrak{B}$ now has the following form

$$\mathfrak{B}^k = (\mathfrak{S}^k)^{-1} + \Pi_h^k(\mathfrak{A}_{reg}^k)^{-1}(\Pi_h^k)^* + \mathfrak{d}(\mathfrak{S}^{k-1})^{-1}\mathfrak{d}^* + \mathfrak{d}\Pi_h^{k-1}(\mathfrak{A}_{reg}^{k-1})^{-1}(\Pi_h^{k-1})^*\mathfrak{d}^*. \tag{5.11}$$

The next corollary shows the bound $\kappa(\mathfrak{B}^k\mathfrak{A}^k) \lesssim 1$ using $\mathfrak{B}^k$ from (5.11).

Corollary 5.1. *Using $\mathfrak{B}^k$ from (5.11) as a preconditioner for solving the linear system (5.1) leads to a condition number $\kappa(\mathfrak{B}^k\mathfrak{A}^k) \lesssim 1$, where the hidden constant depends only on Ω and shape regularity of the mesh.*

Proof. Since this preconditioner results from the one in (5.6) with further decomposing functions in $H_h\mathfrak{L}^{k-1}$, the conditions in Lemma 5.1 follow from (5.7)–(5.10), which gives the desired result. □

Remark 5.2. *We emphasize that instead of directly applying inverses of operators $\mathfrak{A}_{reg}^k$ and $\mathfrak{A}_{reg}^{k-1}$, we can replace them by spectrally equivalent operators, i.e. spectrally equivalent inner products on $H_h^1\mathfrak{L}^k$ and $H_h^1\mathfrak{L}^{k-1}$. Possible choices are multigrid methods and domain decomposition methods.*

Remark 5.3. *For the sake of simplicity, we use model problem (5.3) to derive the mixed-dimensional auxiliary space preconditioner. It is also applicable to the following general problem*

$$\tau(\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^k} + (\mathfrak{d}\mathfrak{q}_h, \mathfrak{d}\tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^{k+1}} = (\mathfrak{f}, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^k} \qquad \forall\, \tilde{\mathfrak{q}}_h \in H_h\mathfrak{L}^k.$$

with $\tau > 0$. In fact, such problem appears in the example presented in Section 6 when the mixed-dimensional permeability is a constant.

6 A Practical Example: Flow in Fractured Porous Media

This section presents a practical example in which the theory from the previous sections comes to use. We consider the setting of flow in fractured porous media in which fractures and intersections are modeled as lower-dimensional manifolds. The goal is to solve for a mass-conservative flow field consisting of a flux and a pressure variable. The flux $\mathfrak{q}$ is considered as a mixed-dimensional $(n-1)$-form whereas the pressure distribution $\mathfrak{p}$ is represented by a mixed-dimensional n-form [8,25]. With respect to the diagram (2.5), this model therefore focuses on the bottom two rows.

We consider the natural case of $n = 3$. Then, the flux is defined as a 3-vector in the three-dimensional surroundings, a 2-vector in the two-dimensional fractures, and a scalar in the one-dimensional intersections between fractures. On the other hand, the pressure is defined as a scalar on all manifolds Ω_i with $i \in I$. In this case, in stead of using $\mathfrak{d}$, we denote $\mathfrak{D}$ as the mixed-dimensional differential, which is an analogue of the operator ∇. Then we represent the complex (2.2) in the same manner as (2.1)

$$H(\mathfrak{D}, \Omega) \xrightarrow{\mathfrak{D}} H(\mathfrak{D}\times, \Omega) \xrightarrow{\mathfrak{D}\times} H(\mathfrak{D}\cdot, \Omega) \xrightarrow{\mathfrak{D}\cdot} L^2(\Omega)$$

The mixed formulation of a fracture flow problem governed by Darcy's law and conservation of mass is then given by: Find $(\mathfrak{q}, \mathfrak{p}) \in H(\mathfrak{D}\cdot, \Omega) \times L^2(\Omega)$ such that

$$(\mathfrak{K}^{-1}\mathfrak{q}, \tilde{\mathfrak{q}})_{L^2\mathfrak{L}^2} - (\mathfrak{D}\cdot\tilde{\mathfrak{q}}, \mathfrak{p})_{L^2\mathfrak{L}^3} = 0, \qquad \forall\, \tilde{\mathfrak{q}} \in H(\mathfrak{D}\cdot, \Omega), \tag{6.1a}$$

$$(\mathfrak{D}\cdot\mathfrak{q}, \tilde{\mathfrak{p}})_{L^2\mathfrak{L}^3} = (\mathfrak{f}, \tilde{\mathfrak{p}})_{L^2\mathfrak{L}^3}, \qquad \forall\, \tilde{\mathfrak{p}} \in L^2\mathfrak{L}^3, \tag{6.1b}$$

where

$$(\mathfrak{K}^{-1}\mathfrak{q}, \tilde{\mathfrak{q}})_{L^2\mathfrak{L}^2} := \sum_{d=1}^{n}\sum_{i\in I^d}(K^{-1}q_i, \tilde{q}_i)_{\Omega_i} + \sum_{j\in I_j^{d-1}}(K_\nu^{-1}\nu_j\cdot q_i, \nu_j\cdot\tilde{q}_i)_{\partial_j\Omega_i} \tag{6.2a}$$

$$(\mathfrak{D}\cdot\mathfrak{q}, \tilde{\mathfrak{p}})_{L^2\mathfrak{L}^3,} := \sum_{d=0}^{n}\sum_{i\in I^d}(\iota_i(\mathfrak{D}\cdot\mathfrak{q}), \tilde{\mathfrak{p}}_i)_{\Omega_i}. \tag{6.2b}$$

Here, $\mathfrak{f}$ is a given source term. $\mathfrak{K}$ is the mixed-dimensional permeability tensor given by a tangential and a normal component, denoted by K and K_ν, respectively.

6.1 Discrete Problem

For the discretization, we follow Section 4 and choose the finite element spaces given by the final two rows in diagram (4.4). Note that this corresponds to the mixed finite element scheme presented and analyzed in [8]. In short, we choose $H_h\mathfrak{L}^2 \times H_h\mathfrak{L}^3$ as in diagram (4.4) and consider the discrete problem: Find $(\mathfrak{q}_h, \mathfrak{p}_h) \in H_h\mathfrak{L}^2 \times H_h\mathfrak{L}^3$ such that

$$(\mathfrak{K}^{-1}\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^2} - (\mathfrak{D}\cdot\tilde{\mathfrak{q}}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} = 0, \qquad \forall\, \tilde{\mathfrak{q}}_h \in H_h\mathfrak{L}^2, \tag{6.3a}$$

$$(\mathfrak{D}\cdot\mathfrak{q}_h, \tilde{\mathfrak{p}}_h)_{L^2\mathfrak{L}^3} = (\mathfrak{f}, \tilde{\mathfrak{p}}_h)_{L^2\mathfrak{L}^3}, \qquad \forall\, \tilde{\mathfrak{p}}_h \in H_h\mathfrak{L}^3. \tag{6.3b}$$

We briefly verify that problem (6.3) is well-posed. For that, we define the weighted norms

$$\|\mathfrak{q}_h\|^2_{H_\alpha\mathfrak{L}^2} := \|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3}, \tag{6.4a}$$

$$\|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3} := \alpha^{-1}\|\mathfrak{p}_h\|^2_{L^2\mathfrak{L}^3}. \tag{6.4b}$$

Here, the scalar α is chosen such that $\alpha \geq \mathfrak{K}_{\min}^{-1}$ with $\mathfrak{K}_{\min} > 0$ being the minimal eigenvalue of $\mathfrak{K}$. In turn, we have

$$\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} \leq \alpha\|\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2}, \qquad \forall\, \mathfrak{q}_h \in H_h\mathfrak{L}^2. \tag{6.5}$$

A key result in the analysis of this problem is that the pair of finite element spaces $H_h\mathfrak{L}^2 \times H_h\mathfrak{L}^3$ satisfies the following inf-sup condition with respect to the weighted norms (6.4).

Lemma 6.1. *There exists a constant $\gamma_B > 0$ independent of the discretization parameter h and the physical parameter $\mathfrak{K}$ such that*

$$\inf_{\mathfrak{p}_h \in H_h\mathfrak{L}^3} \sup_{\mathfrak{q}_h \in H_h\mathfrak{L}^2} \frac{-(\mathfrak{D}\cdot\mathfrak{q}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3}}{\|\mathfrak{q}_h\|_{H_\alpha\mathfrak{L}^2}\|\mathfrak{p}_h\|_{H_\alpha\mathfrak{L}^3}} \geq \gamma_B. \tag{6.6}$$

Proof. For any given $\mathfrak{p}_h \in H_h\mathfrak{L}^3$, according to the inf-sup condition proven in [8] (Lemma 3.2), there exists a $\mathfrak{q}_h \in H_h\mathfrak{L}^2$ such that

$$\begin{aligned} -(\mathfrak{D}\cdot\mathfrak{q}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} &= \|\mathfrak{p}_h\|^2_{L^2\mathfrak{L}^3}, \\ \|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} + \|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3} &\lesssim \|\mathfrak{p}_h\|^2_{L^2\mathfrak{L}^3}. \end{aligned}$$

Using these properties and (6.5), it follows that

$$\begin{aligned} -(\mathfrak{D}\cdot\mathfrak{q}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} &= \|\mathfrak{p}_h\|^2_{L^2\mathfrak{L}^3} \\ &= \left(\alpha^{-\frac{1}{2}}\|\mathfrak{p}_h\|_{L^2\mathfrak{L}^3}\right)\left(\alpha^{\frac{1}{2}}\|\mathfrak{p}_h\|_{L^2\mathfrak{L}^3}\right) \\ &\gtrsim \|\mathfrak{p}_h\|_{H_\alpha\mathfrak{L}^3}\left(\alpha\|\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3}\right)^{\frac{1}{2}} \\ &\geq \|\mathfrak{p}_h\|_{H_\alpha\mathfrak{L}^3}\|\mathfrak{q}_h\|_{H_\alpha\mathfrak{L}^2}. \end{aligned}$$

This completes the proof. □

Based on Lemma 6.1, we can show the well-posedness of problem (6.3) by introducing the following spaces and weighted norms. Let $\mathfrak{X} := H_h\mathfrak{L}^2 \times H_h\mathfrak{L}^3$ and $\mathfrak{X}'$ be the corresponding dual space. Let the energy norm on $\mathfrak{X}$ be given by

$$|||\mathfrak{x}|||^2_{\mathfrak{X}} = |||(\mathfrak{q}_h, \mathfrak{p}_h)|||^2_{\mathfrak{X}} = \|\mathfrak{q}_h\|^2_{H_\alpha\mathfrak{L}^2} + \|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3}, \tag{6.7}$$

which is induced by the inner product $(\cdot,\cdot)_{\mathfrak{X}}$, i.e. $(\mathfrak{x},\mathfrak{x})_{\mathfrak{X}} = |||\mathfrak{x}|||^2_{\mathfrak{X}}$. In addition, let us introduce the following composite bilinear form

$$\mathcal{L}(\mathfrak{x},\mathfrak{y}) := (\mathfrak{K}^{-1}\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^2} - (\mathfrak{D}\cdot\tilde{\mathfrak{q}}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} + (\mathfrak{D}\cdot\mathfrak{q}_h, \tilde{\mathfrak{p}}_h)_{L^2\mathfrak{L}^3}, \tag{6.8}$$

for $\mathfrak{x} = (\mathfrak{q}_h, \mathfrak{p}_h)$ and $\mathfrak{y} = (\tilde{\mathfrak{q}}_h, \tilde{\mathfrak{p}}_h)$. Now we can show the problem (6.3) is well-posed, as presented in the following theorem.

Theorem 6.1. *There exist constants $\beta, \gamma > 0$ independent of discretization parameter h and physical parameter $\mathfrak{K}$ such that*

$$\inf_{\mathfrak{x}\in\mathfrak{X}} \sup_{\mathfrak{y}\in\mathfrak{X}} \frac{\mathcal{L}(\mathfrak{x},\mathfrak{y})}{|||\mathfrak{x}|||_{\mathfrak{X}}|||\mathfrak{y}|||_{\mathfrak{X}}} \geq \gamma \quad \textit{and} \quad |\mathcal{L}(\mathfrak{x},\mathfrak{y})| \leq \beta|||\mathfrak{x}|||_{\mathfrak{X}}|||\mathfrak{y}|||_{\mathfrak{X}}, \qquad \forall\, \mathfrak{x},\mathfrak{y} \in \mathfrak{X}. \tag{6.9}$$

Proof. Let $\mathfrak{x} = (\mathfrak{q}_h, \mathfrak{p}_h) \in \mathfrak{X}$. Due to the inf-sup condition in Lemma 6.1, there exists $\mathfrak{r}_h \in H_h\mathfrak{L}^2$ for this given $\mathfrak{p}_h$ such that

$$-(\mathfrak{D}\cdot\mathfrak{r}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} \geq \gamma_B\|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3}, \tag{6.10a}$$

$$\|\mathfrak{r}_h\|^2_{H_\alpha\mathfrak{L}^2} = \|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3}. \tag{6.10b}$$

Then, choose $\mathfrak{y} = (\tilde{\mathfrak{q}}_h, \tilde{\mathfrak{p}}_h)$ such that $\tilde{\mathfrak{q}}_h = \mathfrak{q}_h + \gamma_B\mathfrak{r}_h$ and $\tilde{\mathfrak{p}}_h = \mathfrak{p}_h + \alpha\mathfrak{D}\cdot\mathfrak{q}_h$, and use (6.10) together with Cauchy-Schwarz inequality, we have

$$\begin{aligned} \mathcal{L}(\mathfrak{x},\mathfrak{y}) &= (\mathfrak{K}^{-1}\mathfrak{q}_h, \mathfrak{q}_h + \gamma_B\mathfrak{r}_h)_{L^2\mathfrak{L}^2} - (\mathfrak{D}\cdot(\mathfrak{q}_h + \gamma_B\mathfrak{r}_h), \mathfrak{p}_h)_{L^2\mathfrak{L}^3} + (\mathfrak{D}\cdot\mathfrak{q}_h, \mathfrak{p}_h + \alpha\mathfrak{D}\cdot\mathfrak{q}_h)_{L^2\mathfrak{L}^3} \\ &= \|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} + \gamma_B(\mathfrak{K}^{-1}\mathfrak{q}_h, \mathfrak{r}_h)_{L^2\mathfrak{L}^2} - \gamma_B(\mathfrak{D}\cdot\mathfrak{r}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^3} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3} \\ &\geq \|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} - \frac{1}{2}\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} - \frac{\gamma_B^2}{2}\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{r}_h\|^2_{L^2\mathfrak{L}^2} + \gamma_B^2\|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3} \\ &\geq \frac{1}{2}\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} - \frac{\gamma_B^2}{2}\|\mathfrak{r}_h\|^2_{H_\alpha\mathfrak{L}^2} + \gamma_B^2\|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3} \\ &= \frac{1}{2}\|\mathfrak{K}^{-\frac{1}{2}}\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^2} + \frac{\gamma_B^2}{2}\|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3} + \alpha\|\mathfrak{D}\cdot\mathfrak{q}_h\|^2_{L^2\mathfrak{L}^3} \\ &\geq \frac{1}{2}\min\{1,\gamma_B^2\}(\|\mathfrak{q}_h\|^2_{H_\alpha\mathfrak{L}^2} + \|\mathfrak{p}_h\|^2_{H_\alpha\mathfrak{L}^3}) \\ &= \frac{1}{2}\min\{1,\gamma_B^2\}|||\mathfrak{x}|||^2_{\mathfrak{X}}. \end{aligned}$$

On the other hand, using continuity of the norms and Cauchy-Schwarz inequality, it is straighforward to verify that $\|\|\mathfrak{y}\|\|_{\mathfrak{X}}^2 \leq \frac{\sqrt{2}}{2}\|\|\mathfrak{x}\|\|_{\mathfrak{X}}^2$, and that gives the first condition in (6.9). The same arguments can be applied to get the second condition on $\mathcal{L}(\cdot,\cdot)$ in (6.9), which concludes the proof. □

6.2 Block Preconditioners based on Auxiliary Space Preconditioning

Let $\langle\cdot,\cdot\rangle$ denote the duality pairing between a function space and its dual. The discrete system (6.3) can be represented by the following block operator form

$$\mathcal{A}\begin{pmatrix}\mathfrak{q}_h\\ \mathfrak{p}_h\end{pmatrix} = \begin{pmatrix}0\\ \mathfrak{f}\end{pmatrix} \quad \text{with} \quad \mathcal{A} := \begin{pmatrix}A_{\mathfrak{q}} & -B^T\\ B & 0\end{pmatrix}, \tag{6.11}$$

where $\langle A_{\mathfrak{q}}\mathfrak{q}_h, \tilde{\mathfrak{q}}_h\rangle := (\mathfrak{K}^{-1}\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^2}$ and $\langle B\mathfrak{q}_h, \tilde{\mathfrak{p}}_h\rangle := (\mathfrak{D}\cdot\mathfrak{q}_h, \tilde{\mathfrak{p}}_h)_{L^2\mathfrak{L}^3}$.

According to Theorem 6.1, $\mathcal{A}$ is an isomorphism with respect to the weighted energy norm (6.7). Following the standard framework [23], the canonical block preconditioner for solving the linear system (6.11) is the Riesz operator $\mathcal{B}: \mathfrak{X}' \mapsto \mathfrak{X}$ corresponding to the inner product $(\cdot,\cdot)_{\mathfrak{X}}$, i.e.,

$$(\mathcal{B}\mathfrak{f}, \mathfrak{x})_{\mathfrak{X}} = \langle \mathfrak{f}, \mathfrak{x}\rangle, \qquad \forall\, \mathfrak{f} \in \mathfrak{X}',\ \mathfrak{x} \in \mathfrak{X}.$$

It follows from Theorem 6.1 that,

$$\kappa(\mathcal{B}\mathcal{A}) = \|\mathcal{B}\mathcal{A}\|_{\mathscr{L}(\mathfrak{X},\mathfrak{X})}\|(\mathcal{B}\mathcal{A})^{-1}\|_{\mathscr{L}(\mathfrak{X},\mathfrak{X})} \leq \frac{\beta}{\gamma}. \tag{6.12}$$

If β and γ are independent of the discretization and physical parameters, then $\mathcal{B}$ is a robust preconditioner for linear system (6.11). Based on the definition of the weighted energy norm (6.7), the preconditioner $\mathcal{B}$ takes the following block diagonal form

$$\mathcal{B} = \begin{pmatrix}A_{\mathfrak{q}} + \alpha B^T B & 0\\ 0 & \alpha^{-1}A_{\mathfrak{p}}\end{pmatrix}^{-1} = \begin{pmatrix}\left(A_{\mathfrak{q}} + \alpha B^T B\right)^{-1} & 0\\ 0 & \alpha A_{\mathfrak{p}}^{-1}\end{pmatrix}, \tag{6.13}$$

where $\langle A_{\mathfrak{p}}\mathfrak{p}_h, \mathfrak{p}_h\rangle := (\mathfrak{p}_h, \mathfrak{p}_h)_{L^2\mathfrak{L}^n}$.

Remark 6.1. *The top block $(A_{\mathfrak{q}} + \alpha B^T B)^{-1}$ in the preconditioner $\mathcal{B}$ corresponds to applying the augmented Lagrangian method to a parameter-dependent problem* (6.11)*. The method is well-known and used in literature [13, 21, 28] for general elliptic problems since it effectively handles the difficulties in convergence of general iterative methods, such as the physical parameter $\mathfrak{K}$ affecting the condition number of the linear system.*

In practice, directly inverting the diagonal blocks in (6.13) might not be feasible. To overcome this difficult, we replace the diagonal blocks by their spectrally equivalent approximation and propose the following block diagonal preconditioner,

$$\mathcal{M}_D = \begin{pmatrix}M_{\mathfrak{q}} & 0\\ 0 & M_{\mathfrak{p}}\end{pmatrix},$$

where

$$\begin{aligned} c_{1,\mathfrak{q}}\langle M_{\mathfrak{q}}\mathfrak{q}_h, \mathfrak{q}_h\rangle &\leq \langle (A_{\mathfrak{q}} + \alpha B^T B)^{-1}\mathfrak{q}_h, \mathfrak{q}_h\rangle \leq c_{2,\mathfrak{q}}\langle M_{\mathfrak{q}}\mathfrak{q}_h, \mathfrak{q}_h\rangle,\\ c_{1,\mathfrak{p}}\langle M_{\mathfrak{p}}\mathfrak{p}_h, \mathfrak{p}_h\rangle &\leq \langle \alpha A_{\mathfrak{p}}^{-1}\mathfrak{p}_h, \mathfrak{p}_h\rangle \leq c_{2,\mathfrak{p}}\langle M_{\mathfrak{p}}\mathfrak{p}_h, \mathfrak{p}_h\rangle, \end{aligned}$$

where $c_{1,\mathfrak{q}}$, $c_{1,\mathfrak{p}}$, $c_{2,\mathfrak{q}}$, and $c_{2,\mathfrak{p}}$ are positive constants independent of discretization and physical parameters. Following [9, 23] and using Theorem 6.1 and (6.12), the condition number of $\mathcal{M}_D\mathcal{A}$ can be directly estimated as

$$\kappa(\mathcal{M}_D\mathcal{A}) \leq \frac{\beta c_2}{\gamma c_1},$$

for $c_2 = \max\{c_{2,\mathfrak{q}}, c_{2,\mathfrak{p}}\}$ and $c_1 = \min\{c_{1,\mathfrak{q}}, c_{1,\mathfrak{p}}\}$. Again, if β, γ, c_1, and c_2 are independent of the discretization and physical parameters, then $\mathcal{M}_D$ is a robust preconditioner as well.

Now we discuss our choices of $M_{\mathfrak{q}}$ and $M_{\mathfrak{p}}$. We start with $M_{\mathfrak{p}}$. Due to the fact that the choice of finite element space for the pressure variable is piecewise constant, it follows that the corresponding mass matrix is diagonal and thus, easily invertible. Therefore, we take $M_{\mathfrak{p}} = \alpha A_{\mathfrak{p}}^{-1}$ and, naturally, $c_{1,\mathfrak{p}} = c_{2,\mathfrak{p}} = 1$.

Regarding $M_{\mathfrak{q}}$, since the first block $A_{\mathfrak{q}} + \alpha B^T B$ corresponds to the problem

$$(\mathfrak{K}^{-1}\mathfrak{q}_h, \tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^2} + \alpha(\mathfrak{D}\cdot\mathfrak{q}_h, \mathfrak{D}\cdot\tilde{\mathfrak{q}}_h)_{L^2\mathfrak{L}^3}. \tag{6.14}$$

It is quite challenging to solve it using traditional methods due to the large kernel of the operator $\mathfrak{D}\cdot$. Therefore, we propose to use the mixed-dimensional auxiliary space preconditioner (5.11), derived in Section 5.2. The form (6.14) can be viewed as a special case of the mixed dimensional problem (5.1) when $n = 3$, $k = 2$, and certain coefficients are added. Directly apply the auxiliary space preconditioner (5.11), we have

$$\mathfrak{B}^2 = (\mathfrak{S}^2)^{-1} + \Pi_h^2(\mathfrak{A}_{reg}^2)^{-1}(\Pi_h^2)^* + (\mathfrak{D}\times)(\mathfrak{S}^1)^{-1}(\mathfrak{D}\times)^* + (\mathfrak{D}\times)(\Pi_h^1)(\mathfrak{A}_{reg}^1)^{-1}(\Pi_h^1)^*(\mathfrak{D}\times)^*.$$

The smoothers $\mathfrak{S}^2$ and $\mathfrak{S}^1$ are chosen to satisfy the second condition in Lemma 5.1. In our implementation, we use symmetric Gauss-Seidel smoothers for both cases. Since the regular space $H_h^1\mathfrak{L}^2$ is given by d_i-tuples of linear Lagrange elements on each $\Omega_{i,h}$ with $i \in I$ and $d_i \geq 1$ and $H_h^1\mathfrak{L}^1$ is defined for $i \in I^3$ (respectively I^2) as a 3-vector field (respectively scalar field) of linear Lagrange elements on $\Omega_{i,h}$, $\mathfrak{A}_{reg}^2$ and $\mathfrak{A}_{reg}^1$ represent the (weighted) inner products on these spaces from (3.6). Moreover, it is often advantageous to further substitute spectrally equivalent operators for $(\mathfrak{A}_{reg}^k)^{-1}$, $k = 2, 1$, denoted by $\mathfrak{B}_{reg}^k$, then the overall auxiliary space preconditioner for solving (6.14) is

$$\mathfrak{B}_{\mathfrak{q}} := (\mathfrak{S}^2)^{-1} + \Pi_h^2\mathfrak{B}_{reg}^2(\Pi_h^2)^* + (\mathfrak{D}\times)(\mathfrak{S}^1)^{-1}(\mathfrak{D}\times)^* + (\mathfrak{D}\times)(\Pi_h^1)\mathfrak{B}_{reg}^1(\Pi_h^1)^*(\mathfrak{D}\times)^*. \tag{6.15}$$

and our choice of $M_{\mathfrak{q}}$ is defined as solving (6.14) by Generalize Minimal Residual (GMRES) method with $\mathfrak{B}_q$ as the preconditioner. In our implementation, $\mathfrak{B}_{ref}^k$, $k = 2, 1$, are defined by one W-cycle unsmoothed aggregation algebraic multigrid method. A theoretical study of their spectrally equivalence properties and thorough comparison of the different available choices is outside the scope of this work and are subjects of our future work.

Lastly, we also consider two block triangular preconditioners

$$\mathcal{M}_L = \begin{pmatrix} M_{\mathfrak{q}}^{-1} & 0 \\ -B & M_{\mathfrak{p}}^{-1} \end{pmatrix}^{-1} \text{ and } \mathcal{M}_U = \begin{pmatrix} M_{\mathfrak{q}}^{-1} & B^T \\ 0 & M_{\mathfrak{p}}^{-1} \end{pmatrix}^{-1},$$

where $\mathcal{M}_L$ serves as a uniform left and $\mathcal{M}_U$ as a uniform right preconditioner for solving (6.11). It can be proven that $\mathcal{M}_L$ and $\mathcal{M}_U$ are so-called field-of-value (FoV) equivalent preconditioners based on the well-posdeness conditions (6.9) and proper inner product induced by $\mathcal{M}_D$. We refer the reader to [1, 2, 9, 22] for a more detailed theoretical analysis on these preconditioners and restrict our focus on their numerical performances in the next section.

7 Numerical Examples

In this section, we propose several numerical tests to confirm the theory derived in previous sections. These tests are designed to emphasize common challenges related to mixed-dimensional problems, such as the geometric complexity and parameter heterogeneity. Also, the problems represent simplified mathematical models of common applications, in this case the model of flow in fractured porous media introduced in Section 6.

In each example, we generate separate simplicial grids on rock and fracture subdomains which combined produce mixed-dimensional geometry Ω. For the sake of simplicity, we assume that Ω is of rectangular type and all the adjacent grids are matching. We want to point out that the analysis presented in this paper allows more flexibility in the geometrical structure.

To solve the system (6.11), we use a Flexible Generalized Minimal Residual (FGMRES) method as an *outer* iterative solver and set the tolerance to be the relative residual less than 10^{-6}. We precondition the outer FGMRES solver with the block preconditioners designed in Section 6.2, i.e., the block diagonal preconditioner $\mathcal{M}_D$ and the block triangular preconditioners $\mathcal{M}_L$ and $\mathcal{M}_U$. As mentioned, the pressure block $\alpha^{-1}A_{\mathfrak{p}}$ is represented as a diagonal matrix using piecewise constant finite elements, thus the inverse is given straightforwardly. On the other hand, the flux block $A_{\mathfrak{q}} + \alpha B^T B$ is approximated by $M_{\mathfrak{q}}$ which is defined by GMRES method preconditioned by the mixed-dimensional auxiliary space preconditioner $\mathfrak{B}_{\mathfrak{q}}$ (6.15). We refer this as the inner solver with a relative residual tolerance set to 10^{-3}. To define $\mathfrak{B}_{\mathfrak{q}}$, we use symmetric Gauss-Seidel method as smoothers $(\mathfrak{S}^2)^{-1}$ and $(\mathfrak{S}^1)^{-1}$, and one application of W-cycle unsmoothed aggregation Algebraic Multigrid method (UA-AMG) as $\mathfrak{B}^2_{reg}$ and $\mathfrak{B}^1_{reg}$.

For obtaining the mixed-dimensional geometry and discretization, we use the PorePy library [18], an open-source simulation tool for fractured and deformable porous media written in Python. The solving methods and preconditioners are implemented in HAZMATH library [3], a finite element solver library written in C. The following numerical examples are performed on a workstation with an 8-core 3GHz Intel Xeon "Sandy Bridge" CPU and 256 GB of RAM.

7.1 Example: Three-dimensional Regular network

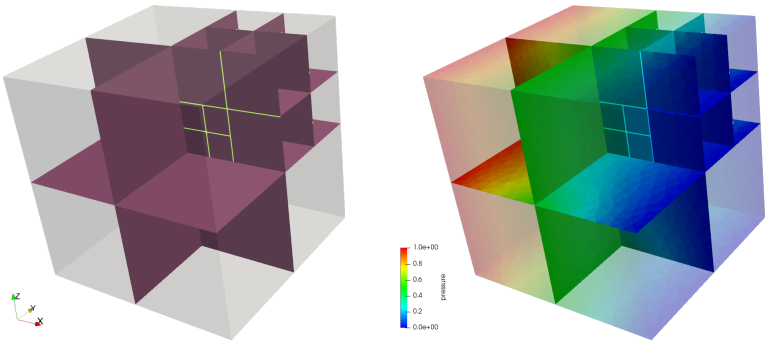


Figure 2: (Left) The three-dimensional unit cube domain in Example 7.1 is decomposed by 9 fracture planes, 9 intersection lines and 1 intersection point. (Right) Pressure solution is presented for the case of a homogeneous permeability tensor $\mathfrak{K} = \boldsymbol{I}$ and a mesh size $h = 1/16$.

This example considers simulations of a 3D problem taken from the benchmark study [6], that is, a three-dimensional Geiger fracture network. The rock domain is a unit cube intersected with a fracture network that consists of nine intersecting planes. The physical parameters are set as following: we take the fracture aperture to be 10^{-2} and the mixed-dimensional permeability tensor is homogeneous $\mathfrak{K} = \boldsymbol{I}$. Within $\mathfrak{K}$, we take into account that due to the reduced model scaling, the tangential K and the normal component K_ν represent the effective values of the permeability field. See [6] for more details. Furthermore, in the heterogeneous case, we consider splitting the tangential permeability into the rock matrix permeability K_m and fracture permeability K_f to allow for different flow patterns within the fracture network, either conducting or blocking the flow in the tangential direction. Also, we consider higher or lower normal permeability K_ν that conducts or blocks the flow over the interface between the rock and the fractures. At the boundary, we impose pressure boundary conditions with unitary pressure drop from $x = 0$ to $x = 1$ boundary planes. The boundary conditions are applied to both the rock matrix and the fracture network. A graphical illustration of the geometry and the numerical solution is given in Figure 2.

Our goal is to investigate the robustness of the block preconditioners in Section 5.2. with respect to discretization parameter h and physical parameter $\mathfrak{K}$. We also vary the scaling parameter α to

study the influence on the convergence rate of the solver and how it changes with the heterogeneous permeability field. We compute and compare number of iterations of the outer and inner solver, as well as the elapsed process (CPU) time of the solver with regards to the number of degrees of freedom.

		$\mathcal{M}_D$			$\mathcal{M}_L$			$\mathcal{M}_U$		
h	N_{dof}	N_{it}	T_{cpu}	rate	N_{it}	T_{cpu}	rate	N_{it}	T_{cpu}	rate
1/4	7173	12 (5)	0.331	–	20 (5)	0.402	–	20 (4)	0.351	–
1/8	17172	11 (6)	0.580	0.643	19 (5)	0.617	0.492	19 (5)	0.553	0.523
1/16	89731	11 (6)	3.229	1.039	19 (7)	4.265	1.169	20 (5)	3.716	1.152
1/32	518291	11 (8)	31.569	1.300	17 (8)	39.499	1.269	18 (7)	37.431	1.317
1/64	3375415	11 (11)	356.098	1.293	17 (11)	482.206	1.335	18 (9)	436.261	1.311

Table 1: Performance of the outer FGMRES solver using preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ in Section 7.1 with regards to mesh refinement. For each preconditioner, we report number of outer (average inner) iterations N_{it} needed to reach the prescribed tolerance and overall elapsed CPU time T_{cpu}. Last column presents the exponential rate of T_{cpu} of outer solver with regards to total degrees of freedom N_{dof}. The permeability tensor is homogeneous and set to $\mathfrak{K} = \boldsymbol{I}$ and the scaling parameter is set to $\alpha = 1$.

The following tables consider the homogeneous permeability case with $\mathfrak{K} = \boldsymbol{I}$ and $\alpha = 1$. In Table 1, we present the results to study the robustness of the preconditioners with respect to the mesh refinement, where each row stands for a mesh twice finer than the previous one. For each preconditioner $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ we give the number of iterations N_{it} of the outer FGMRES solver followed by average number of iterations of the inner GMRES solver in brackets, as well as the CPU time T_{cpu} of the solving process and the exponential rate of the CPU time with regards to the number of degrees of freedom N_{dof}. We clearly see that all preconditioners show that the number of iterations of the outer solver stays stable when refining the mesh, while there is a slight increase of iterations in the inner solver, which is due to our choice of $\mathfrak{B}^2_{reg}$ and $\mathfrak{B}^1_{reg}$. Therefore, we can conclude that preconditioners are robust with regards to the mesh size h, but it suggests a different choice of the inner solver. As mentioned before, the inner solver performance depends on the choices of the spectrally equivalent approximations $\mathfrak{B}^2_{reg}$ and $\mathfrak{B}^1_{reg}$ of operators $(\mathfrak{A}^2_{reg})^{-1}$ and $(\mathfrak{A}^1_{reg})^{-1}$, respectively. These operators are represented in the nodal basis giving a Laplacian-type structure and thus, we have chosen UA-AMG as the approximation method. However, a further analysis that this UA-AMG approximation is actually spectrally equivalent is needed. Although the operators $\mathfrak{A}^2_{reg}$ and $\mathfrak{A}^1_{reg}$ act as a vector-Laplacian on each subdomain, they are still mixed-dimensional, and the off-diagonal coupling between the subdomains is still present which possibly diminishes the preferable structure for AMG methods. Moreover, this suboptimal behavior can be seen in the exponential rates of the CPU time T_{cpu} of the total solving process with regards to the total number of degrees of freedom N_{dof}. We expect T_{cpu} to scale as $\mathcal{O}(N_{dof})$, giving a rate ≈ 1, but all preconditioners show rate closer to 1.3. This is also visible in Figure 3 where the increase in T_{cpu} fairly follows, but does not match the linear rate line. Even with a suboptimal process time performance, we still believe the preconditioners to be working well on the given problem setup, and consider the investigating proper spectrally equivalent approximations of operators $(\mathfrak{A}^2_{reg})^{-1}$ and $(\mathfrak{A}^1_{reg})^{-1}$ in future research.

While still taking the permeability tensor to be homogeneous and unitary, we set the mesh size to $h = 1/32$ and study the performance of the preconditioners with a range of values of the parameter α. Although the theory suggests taking any $\alpha \geq \mathfrak{K}^{-1}_{min}$, we consider instead $\alpha \geq \max\{1, \mathfrak{K}^{-1}_{min}\}$ to achieve reasonable convergence of the underlying augmented Lagrangian method. Table 2 shows the results of the overall outer (and average inner) number of iterations for both diagonal and triangular preconditioners. As expected, the performance of block preconditioners improves with higher values of α since, according to the theory of the augmented Lagrangian method [13], the iterative method should converge faster in those cases. On the other hand, increasing α gives more weight on the mixed-dimensional divergence part of the inner product (6.14), which makes the problem at each inner iteration nearly singular [21, 28]. This may slightly deteriorate the performance of the inner GMRES method, that mostly affects the UA-AMG method

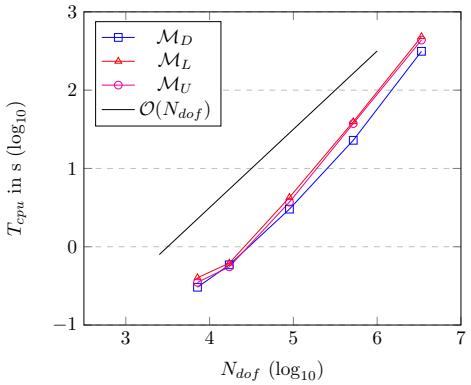


Figure 3: CPU time T_{cpu} of FGMRES solver with block preconditioners compared to total number of degrees of freedom N_{dof} of the linear system in Section 7.1. The values of T_{cpu} and N_{dof} are taken from Table 1. We mark $\mathcal{O}(N_{dof})$ complexity with a black continuous line.

α	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
10^0	11 (8)	17 (8)	18 (7)
10^1	6 (9)	9 (9)	10 (8)
10^2	5 (10)	7 (10)	7 (8)
10^3	4 (12)	5 (11)	7 (9)
10^4	4 (13)	4 (13)	6 (10)

Table 2: Performance of the outer FGMRES solver using preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ in Section 7.1 with regards to varying the scaling parameter α. For each preconditioner, we report number of outer (average inner) iterations needed to reach the prescribed tolerance. The permeability tensor is homogeneous and set to $\mathfrak{K} = \boldsymbol{I}$ and mesh size is set to $h = 1/32$.

within it. Nevertheless, we find a good balance to performance of both the outer and inner solver to be around $\alpha = \max\{1, 100\mathfrak{K}_{min}^{-1}\}$. This can be observed in the study on the heterogeneous permeability field in Table 3. Here, we set the tangential rock component of the permeability to be $K_m = \boldsymbol{I}$, while the tangential fracture component K_f and the normal fracture component K_ν in conjunction assume different values, from low to high permeable case. The results show similar behavior as in Table 2: we get a lower number of outer iterations for $\alpha \gg \max\{1, \mathfrak{K}_{min}^{-1}\}$, but in turn the inner number of iteration increases. Therefore, in this example, we can conclude that taking $\alpha = \max\{1, 100\mathfrak{K}_{min}^{-1}\}$ gives the optimal performance of the preconditioned iterative method.

	$K_f = K_\nu$				
α	10^{-4}	10^{-2}	10^{0}	10^{2}	10^{4}
10^{0}	–	–	11 (5)	5 (5)	5 (13)
10^{2}	–	11 (5)	5 (6)	4 (9)	4 (14)
10^{4}	11 (5)	5 (6)	4 (15)	4 (22)	4 (41)

Table 3: Performance of the outer FGMRES solver using preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ in Section 7.1 with regards to varying the scaling parameter α and the lowest eigenvalue of the permeability tensor $\mathfrak{K}_{min}$. The variations in the eigenvalue spectrum come from the heterogeneity of the fractured porous medium: the tangential rock component of the permeability is $K_m = \boldsymbol{I}$, while we vary the tangential fracture component K_f and the normal fracture component K_ν. For each preconditioner, we report number of outer (average inner) iterations needed to reach the prescribed tolerance. The mesh size is set to $h = 1/16$.

7.2 Example: Two-dimensional Complex Network

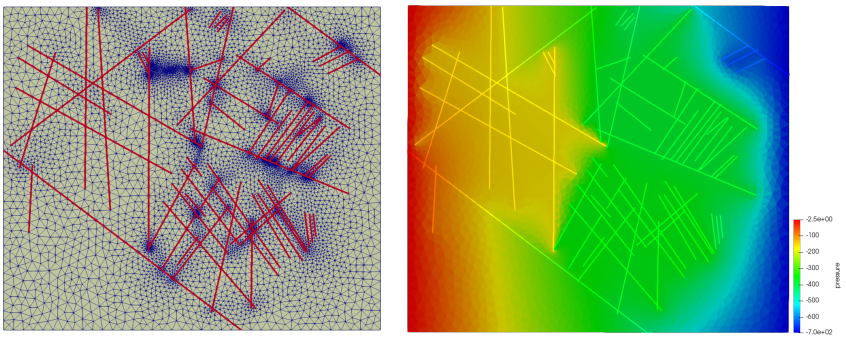


Figure 4: (Left) Graphical representation of the two-dimensional domain and fracture network geometry of Example 7.2. (Right) Pressure solution profile.

We provide another example, chosen from the benchmark study [12], which contains a fracture network from an interpreted outcrop in the Sotra island, near Bergen, Norway. The network includes 63 fractures, all with different length. The porous medium domain spatial dimensions are 700 m $\times$ 600 m with uniform matrix permeability $K_m = \boldsymbol{I}$ m^2. All the fractures have the same scalar tangential and normal permeability $K_n = K_f = 10^5\boldsymbol{I}$ m^2 and aperture $\varepsilon = 10^{-2}$ m. The permeability tensors K_n and K_f are considered to be the effective values, meaning that we incorporate the aperture scaling with ε within the permeability values due to the reduced fracture modeling. See [8,12] for the detailed description of the scalings. The pressure boundary conditions are imposed on all boundaries, with a linear unitary pressure drop from the left to the right boundary. Throughout all the tests, we use a fixed mesh grid with a typical mesh size $h = 18.75$

m and total of 44765 degrees of freedom. See Figure 4 for an illustration of the domain, the mesh and the numerical solution of this problem.

This more realistic case of a fracture network is chosen to demonstrate the robustness of our auxiliary preconditioners, even with a larger number of fractures in the system and a complex fracture network configuration. Large-scale simulations often require handling those features of fractured porous media, as they often appear in geological rock formations in the subsurface and can significantly influence the stability of the any given solving method. In this case, the sharp tips and very acute intersections of fractures may decrease the shape regularity of the mesh, but also increase the condition number of the system and the number of unknowns, as seen in this example and Figure 4. Therefore, we aim to show that our preconditioners still show a good performance under these challenging conditions.

α	$\mathcal{M}_D$	$\mathcal{M}_L$	$\mathcal{M}_U$
10^2	40 (13)	78 (10)	79 (9)
10^3	15 (9)	24 (9)	25 (8)
10^4	8 (5)	10 (5)	11 (5)
10^5	5 (4)	8 (5)	4 (5)
10^6	6 (28)	7 (11)	12 (4)

Table 4: Performance of the outer FGMRES solver using preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ in Section 7.2 with regards to varying the scaling parameter α. For each preconditioner, we report number of outer (average inner) iterations needed to reach the prescribed tolerance. The permeability tensors are set to $K_m = \boldsymbol{I}$, $K_n = K_f = 10^5\boldsymbol{I}$ and the mesh size is set to $h = 18.75$.

We first consider different values of the parameter α, with results given in Table 4. As before, the performance of the diagonal $\mathcal{M}_D$ and triangular preconditioners $\mathcal{M}_L$ and $\mathcal{M}_U$ improves with larger values of α, reaching relatively optimal value at $\alpha = 10^5$ for all three preconditioners. This is different from the previous example in Section 7.1 where the best results are given when $\alpha = \max\{1, 100\mathfrak{K}_{min}^{-1}\}$, considering that in this case we have $\mathfrak{K}_{min}^{-1} = 10^{-5}$. However, there are many differences in the problem settings of these two examples that need to be taken into consideration. First, according to Theorem 5.1 ,the performance of the mixed-dimensional auxiliary space preconditioners can depend on the mixed-dimensional domain Ω and the regularity of the corresponding mesh. In comparison to the example in Section 7.1, the ambient domain in this example is two-dimensional, the domain is more rectangular-type and, due to the complex fracture network configuration, the mesh is less regular. Therefore, we expect a different behavior of both the outer FGMRES and inner GMRES solver in this example. Particularly, this can be seen in Table 4, where the number of outer and inner iterations reduces for larger values of the scaling parameter α, though it started with a large number of iterations in all preconditioners for $\alpha = 10^2$, and for $\alpha = 10^6$ it get slightly larger again. We remind that although larger values of parameter α should improve the performance of the block preconditioners, the divergence part of the inner product (6.14) now dominates, which makes it harder for the inner solver to convergence because of the problem becomes more nearly singular [21, 28].

It is not only the case that the fracture network is more complex, we also have many more fractures included in the domain. This factor should not affect the performance of the preconditioners, which we aim to show in the next set of numerical tests. In the following, we only test the block diagonal preconditioner $\mathcal{M}_D$ since it shows overall best behavior in comparison to the block triangular ones, this particularly evident from Table 4. We also set the scaling parameter $\alpha = 10^5$.

We consider different numbers of fractures included in the original fracture network of 63 fractures in Figure 4. To this end, we randomly select and gradually add more fractures to the network, starting from 1 fracture, to 5, 10, 20, 40 and ultimately all 63 fractures included. We repeat the process four times, creating four different cases, each having either 1, 5, 10, 20 or 40 fractures. See Figure 5 for an illustration of pressure solutions to all four cases, each with 20 randomly selected fractures. The reason to constructing four cases is to eliminate bias in selecting fractures in specific order. We report in Table 5, for all four cases, the number of degrees of freedom N_{dof} and the number of outer (average inner) iterations N_{it} of the FGMRES (GMRES) method preconditioned with the diagonal preconditioner $\mathcal{M}_D$. It is clear that the preconditioned outer

	$\mathcal{M}_D$							
	Case 1		Case 2		Case 3		Case 4	
N_{fracs}	N_{dof}	N_{it}	N_{dof}	N_{it}	N_{dof}	N_{it}	N_{dof}	N_{it}
1	8241	6 (3)	8101	7 (2)	8891	6 (3)	8561	7 (3)
5	17661	7 (3)	10838	6 (3)	9300	6 (3)	11751	6 (3)
10	15809	6 (3)	14437	7 (3)	9180	6 (3)	11998	6 (3)
20	23083	7 (3)	19147	6 (4)	13659	7 (3)	17341	6 (4)
40	31295	5 (4)	25980	6 (4)	29032	7 (4)	27654	6 (4)
63	44765	5 (4)	44765	5 (4)	44765	5 (4)	44765	5 (4)

Table 5: Performance of the outer FGMRES solver using preconditioners $\mathcal{M}_D$, $\mathcal{M}_L$ and $\mathcal{M}_U$ in Section 7.2 with regards to varying number of fractures N_{fracs} in the fracture network. For each preconditioner, we report number of outer (average inner) iterations N_{it} needed to reach the prescribed tolerance. The permeability tensors are set to $K_m = \boldsymbol{I}$, $K_n = K_f = 10^5\boldsymbol{I}$, the mesh size is set to $h = 18.75$ and the scaling parameter $\alpha = 10^5$.

iterative method does not depend on the number of fractures in the fracture network, in all the cases. The same can be seen in the inner solver showing a relatively even number of iterations. Therefore, the robustness of the preconditioner $\mathcal{M}_D$ with regards to the number of fractures in the fracture network is shown, which is consistent with the analysis in the previous sections.

8 Conclusion

In this work, we have derived nodal auxiliary space preconditioners for discretizations of mixed-dimensional partial differential equations. In order to do so, we have extended the stable regular decomposition, both in continuous and discrete setting, to mixed-dimensional geometries. The resulting decomposition differs from the fixed-dimensional case in the way that we do not consider directly the regular inverse, but we establish the regular decomposition hierarchically by combining the regular decompositions on each sub-manifold of the mixed-dimensional domain. Based on this and the auxiliary space preconditioning framework, we propose robust preconditioners to solving mixed-dimensional elliptic problems. We demonstrate how these preconditioners are derived and implemented with an example of mixed-dimensional model of flow in fractured porous media. The robustness of the preconditioners is also verified of two benchmark numerical experiments of fractured porous media. From the numerical experiments, we also see the need of a robust method for solving Laplacian problem in the mixed-dimensional setting in order to further improve the robustness and effectiveness of the proposed preconditioners. This is the topic for our future work.

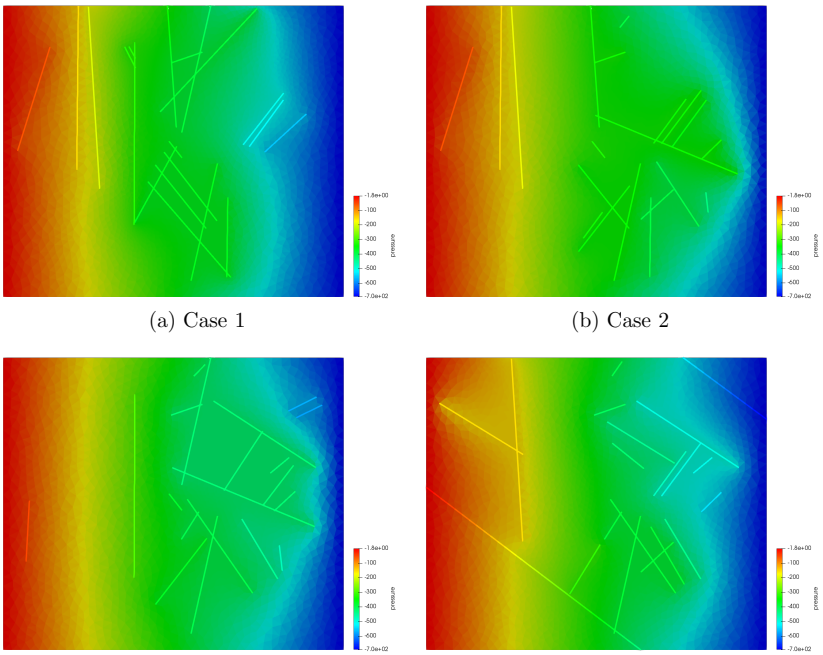
9 Acknowledgments

The first author acknowledges the financial support from the TheMSES project funded by Norwegian Research Council grant 250223. The second author thanks the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) for supporting this work by funding SFB 1313, Project Number 327154368. The work of the third author is partially supported by the National Science Foundation under grant DMS-1620063. A special thanks is extended to Jan M. Nordbotten for valuable comments and discussions on the presented work.

References

[1] J. H. Adler, F. J. Gaspar, X. Hu, P. Ohm, C. Rodrigo, and L. T. Zikatanov, *Robust preconditioners for a new stabilized discretization of the poroelastic equations*, arXiv:1905.10353 [math.NA], (2019).

[2] J. H. Adler, F. J. Gaspar, X. Hu, C. Rodrigo, and L. T. Zikatanov, *Robust block preconditioners for biot's model*, in Domain Decomposition Methods in Science and Engineer-



(a) Case 1 (b) Case 2

(c) Case 3 (d) Case 4

Figure 5: Graphical representation of the pressure solution in Example 7.2 with only 20 fractures included. Each Case 1–4 corresponds to selecting randomly 20 fractures from the original set of 63 fractures, without changing the original positioning of the selected fractures within the domain.

ing XXIV, P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, and O. B. Widlund, eds., Cham, 2018, Springer International Publishing, pp. 3–16.

[3] J. H. Adler, X. Hu, and L. T. Zikatanov, *HAZMATH: A simple finite element, graph and solver library.*

[4] D. N. Arnold, R. S. Falk, and R. Winther, *Finite element exterior calculus, homological techniques, and applications*, Acta numerica, 15 (2006), pp. 1–155.

[5] J. Bear, *Hydraulics of groundwater*, Courier Corporation, 2012.

[6] I. Berre, W. Boon, B. Flemisch, A. Fumagalli, D. Gläser, E. Keilegavlen, A. Scotti, I. Stefansson, and A. Tatomir, *Call for participation: Verification benchmarks for single-phase flow in three-dimensional fractured porous media*, arXiv preprint arXiv:1809.06926, (2018).

[7] W. M. Boon, J. M. Nordbotten, and J. E. Vatne, *Functional analysis and exterior calculus on mixed-dimensional geometries*, tech. rep., arXiv:1710.00556v3 [math.AP], 2018.

[8] W. M. Boon, J. M. Nordbotten, and I. Yotov, *Robust discretization of flow in fractured porous media*, SIAM J. Numer. Anal., 56 (2018), pp. 2203–2233.

[9] A. Budiša and X. Hu, *Block preconditioners for mixed-dimensional discretization of flow in fractured porous media*, arXiv preprint arXiv:1905.13513, (2019).

[10] P. G. Ciarlet, *Mathematical Elasticity: Volume II: Theory of Plates*, vol. 27, Elsevier, 1997.

[11] R. Falk and R. Winther, *Local bounded cochain projections*, Mathematics of Computation, 83 (2014), pp. 2631–2656.

[12] 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 (2018), pp. 239–258.

[13] M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Applications to the Numerical Solution of Boundary-Value Problems*, vol. 15 of Studies in Mathematics and Its Applications, Elsevier, 2000.

[14] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numerica, 11 (2002), pp. 237–339.

[15] R. Hiptmair and C. Pechstein, *Regular decompositions of vector fields – continuous, discrete and structure-preserving*, SAM Research Report No. 2019-18, (2019).

[16] R. Hiptmair and J. Xu, *Nodal auxiliary space preconditioning in H(curl) and H(div) spaces*, SIAM Journal on Numerical Analysis, 45 (2007), pp. 2483–2509.

[17] X. Hu, S. Wu, X.-H. Wu, J. Xu, C.-S. Zhang, S. Zhang, and L. Zikatanov, *Combined preconditioning with applications in reservoir simulation*, Multiscale Modeling & Simulation, 11 (2013), pp. 507–521.

[18] 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*, arXiv:1908.09869 [math.NA], (2019).

[19] T. V. Kolev and P. S. Vassilevski, *Parallel auxiliary space AMG for H(curl) problems*, Journal of Computational Mathematics, 27 (2009), pp. 604–623.

[20] T. V. Kolev and P. S. Vassilevski, *Parallel Auxiliary Space AMG Solver for H(div) Problems*, SIAM Journal on Scientific Computing, 34 (2012), pp. A3079–A3098.

[21] Y.-J. Lee, J. Wu, J. Xu, and L. T. Zikatanov, *Robust subspace correction methods for nearly singular systems*, Mathematical Models and Methods in Applied Sciences, 17 (2007), pp. 1937–1963.

[22] D. Loghin and A. J. Wathen, *Analysis of preconditioners for saddle-point problems*, SIAM J. Sci. Comput., 25 (2004), pp. 2029–2049.

[23] K.-A. Mardal and R. Winther, *Preconditioning discretizations of systems of partial differential equations*, Numer. Linear Algebra Appl., 18 (2011), pp. 1–40.

[24] S. Nepomnyaschikh, *Decomposition and fictitious domains methods for elliptic boundary value problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992, pp. 62–72.

[25] J. M. Nordbotten and W. M. Boon, *Modeling, structure and discretization of hierarchical mixed-dimensional partial differential equations*, in Domain Decomposition Methods in Science and Engineering XXIV, P. E. Bjørstad, S. C. Brenner, L. Halpern, H. H. Kim, R. Kornhuber, T. Rahman, and O. B. Widlund, eds., Springer International Publishing, 2018, pp. 87–101.

[26] J. M. Nordbotten and M. A. Celia, *Geological storage of CO2: modeling approaches for large-scale simulation*, John Wiley & Sons, 2011.

[27] M. Spivak, *Calculus on manifolds: a modern approach to classical theorems of advanced calculus*, CRC Press, 2018.

[28] R. S. Tuminaro, J. Xu, and Y. Zhu, *Auxiliary space preconditioners for mixed finite element methods*, in Domain Decomposition Methods in Science and Engineering XVIII, M. Bercovier, M. J. Gander, R. Kornhuber, and O. Widlund, eds., Berlin, Heidelberg, 2009, Springer Berlin Heidelberg, pp. 99–109.

[29] J. Xu, *The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids*, Computing, 56 (1996), pp. 215–235.

Paper C

A Multiscale Flux Basis for Mortar Mixed Discretizations of Reduced Darcy-Forchheimer Fracture Models

E. Ahmed, A. Fumagalli, A. Budiša

Computer Methods in Applied Mechanics and Engineering 354, (2019), p. 16–36.

doi: 10.1016/j.cma.2019.05.034

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 354 (2019) 16–36
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A multiscale flux basis for mortar mixed discretizations of reduced Darcy–Forchheimer fracture models

Elyes Ahmed, Alessio Fumagalli, Ana Budiša*

Department of Mathematics, University of Bergen, P. O. Box 7800, N-5020 Bergen, Norway

Received 29 October 2018; received in revised form 20 May 2019; accepted 22 May 2019
Available online 27 May 2019

Highlights

- Multiscale flux basis is used to solve a Darcy–Forchheimer reduced fracture model.
- Pre-computation of the basis is completely agnostic to the fracture network model.
- Numerical results show notable subdomain solves reduction in the non-linear solver.

Abstract

In this paper, a multiscale flux basis algorithm is developed to efficiently solve a flow problem in fractured porous media. Here, we take into account a mixed-dimensional setting of the discrete fracture matrix model, where the fracture network is represented as lower-dimensional object. We assume the linear Darcy model in the rock matrix and the non-linear Forchheimer model in the fractures. In our formulation, we are able to reformulate the matrix–fracture problem to only the fracture network problem and, therefore, significantly reduce the computational cost. The resulting problem is then a non-linear interface problem that can be solved using a fixed-point or Newton–Krylov methods, which in each iteration require several solves of Robin problems in the surrounding rock matrices. To achieve this, the flux exchange (a linear Robin-to-Neumann co-dimensional mapping) between the porous medium and the fracture network is done *offline* by pre-computing a multiscale flux basis that consists of the flux response from each degree of freedom (DOF) on the fracture network. This delivers a *conserve* for the basis that handles the solutions in the rock matrices for each degree of freedom in the fractures pressure space. Then, any Robin sub-domain problems are replaced by linear combinations of the multiscale flux basis during the interface iteration. The proposed approach is, thus, agnostic to the physical model in the fracture network. Numerical experiments demonstrate the computational gains of pre-computing the flux exchange between the porous medium and the fracture network against standard non-linear domain decomposition approaches.

Keywords: Reduced fracture models; Darcy–Forchheimer's laws; Multiscale flux basis; Mixed finite element; Non-linear domain decomposition; Newton–Krylov method

* Corresponding author.
E-mail addresses: elyes.ahmed@uib.no (E. Ahmed), alessio.fumagalli@uib.no (A. Fumagalli), ana.budisa@uib.no (A. Budiša).

https://doi.org/10.1016/j.cma.2019.05.034

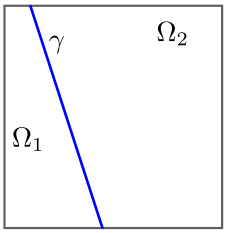


Fig. 1. Graphical example of problem (1.1)–(1.3).

1. Introduction

Using the techniques of domain decomposition [1], a first reduced model has been proposed for flow in a porous medium with a fracture, in which the flow in the fracture is governed by the Darcy–Forchheimer's law while that in the surrounding matrix is governed by Darcy's law.

We consider here the generalized model given in [2], for which we let Ω to be a bounded domain in $\mathbb{R}^d$, $d = 2, 3$, with boundary $\Gamma := \partial\Omega$, and we let $\gamma \subset \Omega$ be a $(d-1)$-dimensional surface that divide Ω into two sub-domains: $\Omega = \Omega_1 \cup \Omega_2 \cup \gamma$, where $\gamma := \partial\Omega_1 \cap \partial\Omega_2$ and $\Gamma_i := \partial\Omega_i \cap \partial\Omega$, $i = 1, 2$. The reduced model problem as presented in [2] is as follows:

$$\boldsymbol{K}_i^{-1}\boldsymbol{u}_i + \nabla p_i = 0 \quad \text{in } \Omega_i, \tag{1.1a}$$

$$\nabla \cdot \boldsymbol{u}_i = f_i \quad \text{in } \Omega_i, \tag{1.1b}$$

$$p_i = 0 \quad \text{in } \Gamma_i, \tag{1.1c}$$

for $i = 1, 2$, together with

$$(\boldsymbol{K}_\gamma^{-1} + \beta_\gamma \boldsymbol{I}|\boldsymbol{u}_\gamma|)\boldsymbol{u}_\gamma = -\nabla_\tau p_\gamma \quad \text{in } \gamma, \tag{1.2a}$$

$$\nabla_\tau \cdot \boldsymbol{u}_\gamma = f_\gamma + (\boldsymbol{u}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{u}_2 \cdot \boldsymbol{n}_2) \quad \text{in } \gamma, \tag{1.2b}$$

$$p_\gamma = 0 \quad \text{in } \partial\gamma, \tag{1.2c}$$

and subject to the following interface conditions

$$-\boldsymbol{u}_i \cdot \boldsymbol{n}_i + \alpha_\gamma p_i = \alpha_\gamma p_\gamma \quad \text{on } \gamma, \tag{1.3}$$

for $i = 1, 2$. Here, ∇_τ denotes the $(d-1)$-dimensional gradient operator in the plane of γ, the coefficient $\boldsymbol{K}_i$, $i = 1, 2$, is the hydraulic conductivity tensor in the sub-domain Ω_i, and $\boldsymbol{K}_\gamma$ is the hydraulic conductivity tensor in the fracture, $\boldsymbol{I} \in \mathbb{R}^{d\times d}$ is the identity matrix, $\boldsymbol{n}_i$ is the outward unit normal vector to $\partial\Omega_i$, and β_γ is a non-negative scalar known as the Forchheimer coefficient. In (1.3), the coefficient α_γ is a function proportional to the normal component of the permeability of the physical fracture and inversely proportional to the fracture width/aperture. We refer to [3] for a more detailed model description. For illustration purposes, we give a simple graphical example of a fractured porous medium in Fig. 1.

The system (1.1)–(1.3) can be seen as a domain decomposition problem, with non-standard and non-local boundary conditions between the sub-domains Ω_i, $i = 1, 2$. Eqs. (1.1) are the mass conservation equation and the Darcy's law equation in the sub-domain Ω_i while Eqs. (1.2) are the lower-dimensional mass conservation and the Darcy–Forchheimer equation in the fracture of co-dimension 1. The last Eq. (1.3) can be seen as a Robin boundary condition for the sub-domain Ω_i with a dependence on the pressure on the fracture γ. Clearly, if $\beta_\gamma = 0$, then (1.2) is reduced to a linear Darcy flow in the fracture. The homogeneous Dirichlet boundary conditions (1.1c) and (1.2c) are considered merely for simplicity. The functions $f_i \in L^2(\Omega_i)$, $i = 1, 2$ and $f_\gamma \in L^2(\gamma)$ are source terms in the matrix and in the fracture, respectively.

The mixed-dimensional problem (1.1)–(1.3) is an alternative to the possibility of using a very fine grid in the physical fracture and a necessarily much coarser grid away from the fracture. This idea was developed in [4] for highly permeable fractures and in [5] for fractures that may be highly permeable or nearly impermeable. We also refer to [6–8] for similar models. For all of the above models, where the linear Darcy's law is used

as the constitutive law for flow in the fractures as well as in the surrounding domains, there are interactions between fractures and surrounding domains. This coupling is ensured using Robin type conditions as in [9], delivering discontinuous normal velocity and pressure across the fractures. Particularly, for fractures with large enough permeability, Darcy's law is replaced by Darcy–Forchheimer's law as established in [2], which complicates the coupling with the surrounding medium.

Several numerical schemes have been developed for fracture models, such as a cell-centred finite volume scheme in [10], an extended finite element method in [11], a mimetic finite difference [12] and a block-centred finite difference method in [13]. The aforementioned numerical approaches solve coupled fracture models directly. However, different equations defined in different regions are varied in type, such as coupling linear and non-linear systems, and often interface conditions involve new variables in different domains, which results in very complex algebraic structures. Particularly, several papers deal with the analysis and implementation of mixed methods applied to the above model problem in the linear case, on conforming and non-conforming grids [9,14–17]. In [1], the model problem (1.1)–(1.3) was solved using domain decomposition techniques based on mixed finite element methods (see [4] for the linear counterpart).

The purpose of this paper is to propose an efficient domain decomposition method to solve (1.1)–(1.3) based on the multiscale mortar mixed finite element method (MMMFEM) [18]. The method reformulates (1.1)–(1.3) into an interface problem by eliminating the sub-domain variables. The resulting interface problem is a superposition of a non-linear operator handling the flow on the fracture and a linear operator presenting the flux contribution from the sub-domains. When applying the MMMFEM, an *outer–inner iterative algorithm* like, the Newton-GMRes (or any Krylov solver) method or fixed-point-GMRes method, is used to solve the interface problem. As an example, if a fixed-point method (outer) is adopted, the linearized interface equation for the interface update can be solved with a domain decomposition algorithm (inner), in which at each iteration sub-domain solves, together with inter-processor communication, are required. The main issue of this outer–inner algorithm is that it leads to an excessive calculation from the sub-domains, as the dominant computational cost is measured by the number of sub-domain solves.

The new implementation recasts this algorithm by distinguishing the linear and non-linear contributions in the overall calculation and employing the multiscale flux basis functions from [18] for the linear part of the problem, before the non-linear interface iterations begin. The fact that the non-linearity in (1.1)–(1.3) is only within the fracture, we can adopt the notion that sub-domain problems can be expressed as a superposition of multiscale basis functions. In our terminology the mortar variable considered in [18] becomes the fracture pressure, these multiscale flux basis with respect to the fracture pressure can be computed by solving a fixed number of Robin sub-domain problems, that is equal to the number of fracture pressure degrees of freedom per sub-domain. Furthermore, this is done in parallel without any inter-processor communication.

An inexpensive linear combination of the multiscale flux basis functions then circumvents the need to solve any sub-domain problems in the inner domain decomposition iterations. This procedure can be enhanced by applying interface preconditioners as in [9,19,20] and by using a posteriori error estimates of [21] to adaptively refine the mesh grids. This calculation made in an *offline* step typically spares numerous unnecessary sub-domain solves. Precisely, in the original implementations, the number of sub-domain solves is approximately equal to $\sum_{k=1}^{N_{\text{lin}}} N_{\text{dd}}^{i}$, where N_{lin} is the number of iterations of the linearization procedure, and N_{dd}^{k} denotes the number of domain decomposition iterations (GMRes or any Krylov solver). For the new implementation, the number of sub-domains solves will be reduced if $\sum_{k=1}^{N_{\text{lin}}} N_{\text{dd}}^{k}$ exceeds the maximum number of fracture pressure degrees of freedom on any sub-domain.

This step of *freezing* the contributions on the flow from the rock matrices can be *easily coded, cheaply evaluated*, and *efficiently used* in *practical simulations*, i.e, it permits reusing the same basis functions to extend (1.1)–(1.3) by simulating various linear and non-linear models for flow in the fracture, such as generalized Forchheimer's laws:

$$(\boldsymbol{K}_{\gamma}^{-1} + \beta_{\gamma}\boldsymbol{I}|\boldsymbol{u}_{\gamma}| + \zeta_{\gamma}\boldsymbol{I}|\boldsymbol{u}_{\gamma}|^{2})\boldsymbol{u}_{\gamma} = -\nabla_{\tau} p_{\gamma},$$
$$(\boldsymbol{K}_{\gamma}^{-1} e^{\zeta p_{\gamma}} + \beta_{\gamma}\boldsymbol{I}|\boldsymbol{u}_{\gamma}|)\boldsymbol{u}_{\gamma} = -\nabla_{\tau} p_{\gamma},$$

as well as exploring the fracture and barrier cases and comparing in a cheap way various non-linear solvers to (1.1)–(1.3). Crucially, the present approach can naturally be integrated into discrete fracture networks (DFNs) models [16,17,22,23], which in contrast to discrete fracture models (DFMs), do not consider the flow in the surrounding sub-domains, but handle both a large number of fractures and a complex interconnecting network of these fractures.

For the present setting, we allow for the discretization of (1.1)–(1.3) by different numerical methods applied separately in the surrounding sub-domains and in the fracture. We allow for the cases where the grids of the porous sub-domains do not match along the fracture, where different mortar grid elements are used. We also investigate the case where the permeability in the fracture $\boldsymbol{K}_\gamma$ is much lower than the permeability in the surrounding matrix $\boldsymbol{K}$.

The library PorePy [24] has been used and extended to cover the numerical schemes and examples introduced in this article. The main contribution to the library is the implementation of the multiscale and domain decomposition frameworks. Even if we focus on lowest-order Raviart–Thomas–Néd élec finite elements, our implementation is agnostic with respect to the numerical scheme. The example presented are also available in the GitHub repository.

This paper is organized as follows: Firstly, the variational formulation of the problem and the MMMFEM approximation are given in Section 2. Therefore, the reduction of the original problem into non-linear interface problem is introduced. The linearization–domain-decomposition procedures are formulated in Section 3. Section 4 describes the implementation based on the multiscale flux basis. We show that structurally the same implementation can be extended for more complex intersecting fractures model. Finally, we showcase the performance of our method on several numerical examples in Section 5 and draw the conclusions in Section 6.

2. Non-linear domain decomposition method

As explained earlier, it is natural to solve the mixed-dimensional problem (1.1)–(1.3) using domain decomposition techniques. To this aim, we introduce the weak spaces in each sub-domain Ω_i, $i = 1, 2$,

$$\boldsymbol{V}_i := \boldsymbol{H}(\mathrm{div}, \Omega_i), \qquad M_i := L^2(\Omega_i),$$

and define their global versions by

$$\boldsymbol{V} := \bigoplus_{i=1}^{2} \boldsymbol{V}_i, \qquad M := \bigoplus_{i=1}^{2} M_i.$$

Equivalently, we introduce the weak spaces on the fracture γ, i.e,

$$\boldsymbol{V}_\gamma := \boldsymbol{H}(\mathrm{div}_\tau, \gamma), \qquad M_\gamma := L^2(\gamma).$$

Following [1,9], a mixed-dimensional weak form of (1.1)–(1.3) asks for $(\boldsymbol{u}, p) \in \boldsymbol{V} \times M$ and $(\boldsymbol{u}_\gamma, p_\gamma) \in \boldsymbol{V}_\gamma \times M_\gamma$ such that, for each $i \in \{1, 2\}$,

$$\begin{aligned}
&(\boldsymbol{K}^{-1}\boldsymbol{u}, \boldsymbol{v})_{\Omega_i} + \alpha_\gamma^{-1}\langle \boldsymbol{u}\cdot\boldsymbol{n}_i, \boldsymbol{v}\cdot\boldsymbol{n}_i\rangle_\gamma = (p, \nabla\cdot\boldsymbol{v})_{\Omega_i} - \langle p_\gamma, \boldsymbol{v}\cdot\boldsymbol{n}_i\rangle_\gamma && \forall \boldsymbol{v} \in \boldsymbol{V},\\
&(\nabla\cdot\boldsymbol{u}, q)_{\Omega_i} = (f, q)_{\Omega_i} && \forall q \in M, && \text{(2.1a)}\\
&\langle \mathcal{K}^{-1}(\boldsymbol{u}_\gamma)\boldsymbol{u}_\gamma, \boldsymbol{v}_\gamma\rangle_\gamma = \langle p_\gamma, \nabla_\tau\cdot\boldsymbol{v}_\gamma\rangle_\gamma && \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_\gamma, && \text{(2.1b)}\\
&\langle \nabla_\tau\cdot\boldsymbol{u}_\gamma, q_\gamma\rangle_\gamma = \langle f_\gamma + [\![\boldsymbol{u}\cdot\boldsymbol{n}]\!], q_\gamma\rangle_\gamma && \forall q_\gamma \in M_\gamma, && \text{(2.1c)}
\end{aligned}$$

where we introduced the functions $\boldsymbol{K}$ and f in $\Omega_1 \cup \Omega_2$ such that $\boldsymbol{K}_i = \boldsymbol{K}|_{\Omega_i}$, and $f_i = f|_{\Omega_i}$, $i = 1, 2$. The jump $[\![\cdot]\!]$ is defined by

$$[\![\boldsymbol{u}\cdot\boldsymbol{n}]\!] := \boldsymbol{u}_1\cdot\boldsymbol{n}_1 + \boldsymbol{u}_2\cdot\boldsymbol{n}_2,$$

with $\boldsymbol{n}_i$ the outer unit normal vector of Ω_i on γ, for $i = 1, 2$. Finally, the non-linear term is defined as

$$\mathcal{K}^{-1}(\boldsymbol{u}_\gamma) := \boldsymbol{K}_\gamma^{-1} + \beta_\gamma \boldsymbol{I}|\boldsymbol{u}_\gamma|,$$

The reader is referred to [2] for proof of the existence and uniqueness of a solution to the variational formulation (2.1).

2.1. The discrete problem

Let $\mathcal{T}_{h,i}$ be a partition of the sub-domain Ω_i into either d-dimensional simplicial and/or rectangular elements. We also let $\mathcal{T}_{h,\gamma}$ to be a partition of the fracture γ into $(d-1)$-dimensional simplicial and/or rectangular elements. Note that, for both partitions, general elements can be treated via sub-meshes, see [21] and the references therein.

Moreover, we assume that each partition is conforming within each sub-domain as well as in the fracture. The meshes $\mathcal{T}_{h,i}$, $i = 1, 2$, are allowed to be non-conforming on the fracture-interface γ, but also different from $\mathcal{T}_{h,\gamma}$. We then set $\mathcal{T}_h := \cup_{i=1}^{2} \mathcal{T}_{h,i}$ and denote by h the maximal element diameter in $\mathcal{T}_h$. For the scalar unknowns, we introduce the approximation spaces $M_h := M_{h,1} \times M_{h,2}$ and $M_{h,\gamma}$, where $M_{h,i}$, $i = 1, 2$, respectively $M_{h,\gamma}$, is the space of piecewise constant functions associated with $\mathcal{T}_{h,i}$, $i = 1, 2$, respectively $\mathcal{T}_{h,\gamma}$. For the vector unknowns, we introduce the approximation spaces $\boldsymbol{V}_h := \boldsymbol{V}_{h,1} \times \boldsymbol{V}_{h,2}$ and $\boldsymbol{V}_{h,\gamma}$, where $\boldsymbol{V}_{h,i}$, $i = 1, 2$ and $\boldsymbol{V}_{h,\gamma}$, are the lowest-order Raviart–Thomas–Néd élec finite elements spaces associated with $\mathcal{T}_{h,i}$, $i = 1, 2$ and $\mathcal{T}_{h,\gamma}$, respectively. Clearly, in contrast to what is done in [18,25], we use the same order of the polynomials for the interface-pressure and the normal traces of the sub-domain velocities on the interface.

The discrete mixed-dimensional finite element approximation of (2.1) is as follows: find $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times M_h$ and $(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma}$ such that, for each $i \in \{1, 2\}$,

$$
\begin{aligned}
&(\boldsymbol{K}^{-1}\boldsymbol{u}_h, \boldsymbol{v})_{\Omega_i} + \alpha_\gamma^{-1}\langle \boldsymbol{u}_h \cdot \boldsymbol{n}_i, \boldsymbol{v} \cdot \boldsymbol{n}_i\rangle_\gamma = (p_h, \nabla \cdot \boldsymbol{v})_{\Omega_i} - \langle p_{h,\gamma}, \boldsymbol{v} \cdot \boldsymbol{n}_i\rangle_\gamma \quad && \forall \boldsymbol{v} \in \boldsymbol{V}_h, \\
&(\nabla \cdot \boldsymbol{u}_h, q)_{\Omega_i} = (f, q)_{\Omega_i} && \forall q \in M_h, && \text{(2.2a)} \\
&\langle \mathcal{K}^{-1}(\boldsymbol{u}_{h,\gamma})\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_\gamma = \langle p_{h,\gamma}, \nabla_\tau \cdot \boldsymbol{v}_\gamma\rangle_\gamma && \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_{h,\gamma}, && \text{(2.2b)} \\
&\langle \nabla_\tau \cdot \boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_\gamma = \langle f_\gamma + [\![\boldsymbol{u}_h \cdot \boldsymbol{n}]\!], q_\gamma\rangle_\gamma && \forall q_\gamma \in M_{h,\gamma}. && \text{(2.2c)}
\end{aligned}
$$

The next step in formulating a multiscale flux basis algorithm to solve (2.2) is to adopt domain decomposition techniques to reduce the global mixed-dimensional problem to an interface problem posed only on the fracture [19].

2.2. Reduction to interface problem

We introduce the discrete (linear) Robin-to-Neumann operator $\mathcal{S}_i^{\mathrm{RtN}}$, $i = 1, 2$:

$$
\mathcal{S}_i^{\mathrm{RtN}} : M_{h,\gamma} \times M_i \to M_{h,\gamma}, \qquad \mathcal{S}_i^{\mathrm{RtN}}(\lambda_{h,\gamma}, f) := -\boldsymbol{u}_h(\lambda_{h,\gamma}, f) \cdot \boldsymbol{n}_i,
$$

where $(\boldsymbol{u}_h, p_h) \in \boldsymbol{V}_h \times M_h$ is the solution of the sub-domain problems with source term f, homogeneous Dirichlet boundary condition on $\partial\Omega$, and λ as a Robin boundary condition along the fracture γ, i.e, for $i = 1, 2$,

$$
\begin{aligned}
&(\boldsymbol{K}^{-1}\boldsymbol{u}_h, \boldsymbol{v})_{\Omega_i} + \alpha_\gamma^{-1}\langle \boldsymbol{u}_h \cdot \boldsymbol{n}_i, \boldsymbol{v} \cdot \boldsymbol{n}_i\rangle_{\Omega_i} = (p_h, \nabla \cdot \boldsymbol{v})_{\Omega_i} - \langle \lambda_{h,\gamma}, \boldsymbol{v} \cdot \boldsymbol{n}_i\rangle_\gamma \quad && \forall \boldsymbol{v} \in \boldsymbol{V}_{h,i}, \\
&(\nabla \cdot \boldsymbol{u}_h, q)_{\Omega_i} = (f, \boldsymbol{v})_{\Omega_i} && \forall q \in M_{h,i}. && \text{(2.3a)}
\end{aligned}
$$

Then we set

$$
\mathcal{S}^{\mathrm{RtN}}(\lambda_{h,\gamma}, f) := \sum_{i=1}^{2} \mathcal{S}_i^{\mathrm{RtN}}(\lambda_{h,\gamma}, f_i).
$$

With these notations, we can see that solving (2.2) is equivalent to solving the following non-linear mixed interface problem: find $(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma}$ such that,

$$
\begin{aligned}
&\langle \mathcal{K}^{-1}(\boldsymbol{u}_{h,\gamma})\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_\gamma - \langle p_{h,\gamma}, \nabla_\tau \cdot \boldsymbol{v}_\gamma\rangle_\gamma = 0 && \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_{h,\gamma}, && \text{(2.4a)} \\
&\langle \nabla_\tau \cdot \boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_\gamma + \langle \mathcal{S}^{\mathrm{RtN}}(p_{h,\gamma}, f), q_\gamma\rangle_\gamma = \langle f_\gamma, q_\gamma\rangle_\gamma && \forall q_\gamma \in M_{h,\gamma}, && \text{(2.4b)}
\end{aligned}
$$

or equivalently

$$
\begin{aligned}
&\langle \mathcal{K}^{-1}(\boldsymbol{u}_{h,\gamma})\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_\gamma - \langle p_{h,\gamma}, \nabla_\tau \cdot \boldsymbol{v}_\gamma\rangle_\gamma = 0 && \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_{h,\gamma}, && \text{(2.5a)} \\
&\langle \nabla_\tau \cdot \boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_\gamma + \langle \mathcal{S}_\gamma(p_{h,\gamma}), q_\gamma\rangle_\gamma = \langle f_\gamma + g_\gamma, q_\gamma\rangle_\gamma && \forall q_\gamma \in M_{h,\gamma}, && \text{(2.5b)}
\end{aligned}
$$

where we have set

$$
\mathcal{S}_\gamma(p_{h,\gamma}) := \mathcal{S}^{\mathrm{RtN}}(p_{h,\gamma}, 0) \quad \text{and} \quad g_\gamma := -\mathcal{S}^{\mathrm{RtN}}(0, f). \tag{2.6}
$$

The above distinction is classical in domain decomposition techniques in which we split the sub-domain problems into two families of local problems on each Ω_i: one is with zero source and specified Robin value on the fracture-interface, and the other is with zero Robin value on the fracture-interface and specified source. In compact form, the mixed interface Darcy–Forchheimer problem (2.5) can be rewritten as

$$
\begin{bmatrix} \mathcal{K}^{-1}(\cdot) & B_\gamma^{\mathrm{T}} \\ B_\gamma & \mathcal{S}_\gamma \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_{h,\gamma} \\ p_{h,\gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ g_\gamma + f_\gamma \end{bmatrix}. \tag{2.7}
$$

This system is a non-linear mixed interface problem [26] that can be solved iteratively by using fixed point iterations or via a Newton–Krylov method. To present the two approaches, let us first consider the linear context, i.e, suppose the operator $\mathcal{K}^{-1}(\cdot)$ is linear. Then (2.7) is the system associated to the linear mixed Darcy problem on the fracture that can be solved using a Krylov type method, such as GMRes or MINRes. Given an initial guess $\boldsymbol{w}_{h,\gamma}^{(0)} := [\boldsymbol{u}_{h,\gamma}^{(0)}, p_{h,\gamma}^{(0)}]^\top$, the GMRes algorithm computes

$$\boldsymbol{w}_{h,\gamma}^{(m)} := \underset{\boldsymbol{v} \in \boldsymbol{w}_{h,\gamma}^{(0)} + \mathcal{K}_m(\mathcal{A}_\gamma, \boldsymbol{r}_\gamma^{(0)})}{\arg\min} \|\boldsymbol{b}_\gamma - \mathcal{A}_\gamma \boldsymbol{v}\|_2 \quad \text{for} \quad m \geq 1, \tag{2.8}$$

as an approximate solution to (2.7), where $\mathcal{A}_\gamma$ is the associated stiffness matrix of the linear system, $\boldsymbol{b}_\gamma$ is the right-hand side, and $\mathcal{K}_m(\mathcal{A}_\gamma, \boldsymbol{r}_\gamma^{(0)})$ is the m-dimensional Krylov subspace generated by the initial residual $\boldsymbol{r}_\gamma^{(0)} := \boldsymbol{b}_\gamma - \mathcal{A}_\gamma \boldsymbol{w}_{h,\gamma}^{(0)}$, i.e,

$$\mathcal{K}_m(\mathcal{A}_\gamma, \boldsymbol{r}_\gamma^{(0)}) := \text{span}(\boldsymbol{r}_\gamma^{(0)}, \mathcal{A}_\gamma \boldsymbol{r}_\gamma^{(0)}, \ldots, \mathcal{A}_\gamma^{(m-1)} \boldsymbol{r}_\gamma^{(0)}).$$

Clearly, each GMRes iteration needs to evaluate the action of the Robin-to-Neumann type operator $\mathcal{S}_\gamma$ via (2.6), representing physically the contributions on the flow from the rock matrices, i.e, to solve one Robin sub-domain problem per sub-domain. Thus the GMRes algorithm is implemented in the matrix-free context [18,27,28].

One can easily observe that the evaluation of $\mathcal{S}_\gamma$ dominates the total computational costs in (2.8). In practice, this step is done in parallel and involves inter-processor communication across the fracture-interface [27]. To present the evaluating algorithm of $\mathcal{S}_\gamma$, we let $\mathcal{D}_{h,i}^\top : \boldsymbol{V}_{h,i} \cdot \boldsymbol{n}_i|_\gamma \to M_{h,\gamma}$ be the L^2-orthogonal projection from the normal trace of the velocity space onto the mortar space normal trace of the velocity space in sub-domain Ω_i, $i = 1, 2$, onto the pressure space on the fracture $M_{h,\gamma}$. We then summarize the evaluation of the interface operator by the following steps:

Algorithm 2.1 (*Evaluating the Action of $\mathcal{S}_\gamma$*)**.**

1. Enter an interface data $\varphi_{h,\gamma}$.
2. **For** $i = 1 : 2$

   (a) Project mortar data onto sub-domain boundary, i.e,

   $$\varphi_{h,\gamma} \xrightarrow{\mathcal{D}_{h,i}} \lambda_{h,\gamma}.$$

   (b) Solve the sub-domain problem (2.3) with Robin boundary condition $\lambda_{h,\gamma}$ and with $f = 0$.
   (c) Project the resulting flux onto the mortar space $M_{h,\gamma}$, i.e,

   $$-\boldsymbol{u}_h(\lambda_{h,\gamma}, 0) \cdot \boldsymbol{n}_i \xrightarrow{\mathcal{D}_{h,i}^\top} -\mathcal{D}_{h,i}^\top \boldsymbol{u}_h(\lambda_{h,\gamma}, 0) \cdot \boldsymbol{n}_i.$$

   EndFor
3. Compute the flow contribution from the sub-domains to the fracture given by the flux jump across the fracture, i.e,

$$\mathcal{S}_\gamma(\varphi_{h,\gamma}) = \sum_{i=1}^{2} -\mathcal{D}_{h,i}^\top \boldsymbol{u}_h(\lambda_{h,\gamma}, 0) \cdot \boldsymbol{n}_i.$$

3. Non-linear interface iterations

In this section, we form two linearization–domain-decomposition algorithms to solve the mixed interface Darcy–Forchheimer problem (2.5). For the linearization (outer) of (2.5), a first algorithm based on a fixed-point method is presented along with a second one based on Newton-GMRes method [29,30]. For the solver of the inner systems (domain decomposition systems), both methods uses the GMRes method (2.8) to solve the reduced mixed interface problems. Note that the two approaches have competitive performance for such non-linear model problems and they lead to different applications of the multiscale flux basis functions of Section 4.

3.1. Method 1: fixed-point-GMRes

We consider first a standard fixed-point approach to solve the interface Darcy–Forchheimer problem (2.5) (see [30]). Given an initial value $\boldsymbol{u}_{h,\gamma}^{(0)}$, being the solution of a linear Darcy, for $k = 1, 2, \ldots$, until convergence, find $(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma}$ such that,

$$\langle \boldsymbol{\mathcal{K}}^{-1}(\boldsymbol{u}_{h,\gamma}^{(k-1)})\boldsymbol{u}_{h,\gamma}^{(k)}, \boldsymbol{v}_\gamma\rangle_\gamma - \langle p_{h,\gamma}^{(k)}, \nabla_\tau \cdot \boldsymbol{v}_\gamma\rangle_\gamma = 0 \qquad \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_{h,\gamma}, \tag{3.1a}$$

$$\langle \nabla_\tau \cdot \boldsymbol{u}_{h,\gamma}^{(k)}, q_\gamma\rangle_\gamma + \langle \mathcal{S}_\gamma(p_{h,\gamma}^{(k)}), q_\gamma\rangle_\gamma = \langle f_\gamma + g_\gamma, q_\gamma\rangle_\gamma \quad \forall q_\gamma \in M_{h,\gamma}. \tag{3.1b}$$

This process is linear and can be solved using GMRes method (2.8), where each iteration needs to set up the action of the Robin-to-Neumann operator $\mathcal{S}_\gamma$ using Algorithm 2.1. The above fixed-point-GMRes algorithm is iterated until a fixed-point residual tolerance reaches some prescribed value.

The result of this procedure is then used to generate the solution in the sub-domains via

$$\boldsymbol{u}_h|_{\Omega_i} = \boldsymbol{u}_h(p_{h,\gamma}^{(\infty)}, 0)|_{\Omega_i} + \boldsymbol{u}_h(0, f_i), \tag{3.2a}$$

$$p_h|_{\Omega_i} = p_h(p_{h,\gamma}^{(\infty)}, 0)|_{\Omega_i} + p_h(0, f_i), \tag{3.2b}$$

for $i = 1, 2$, requiring two additional sub-domain solves, and where $p_{h,\gamma}^{(\infty)}$ indicates the fracture pressure at convergence.

Remark 3.1 (*An Alternative to* (3.1))**.** A well-known drawback of GMRes algorithm for solving the interface-fracture problem (3.1) is that the number of iterations depends essentially on the number of sub-domain solves. A preconditioner is then necessary to reduce the iterations number to a reasonable level. To this aim, it is possible to reformulate (3.1) into a primal problem: at the iteration $k \geq 1$, by solving for the sole scalar unknown $p_{h,\gamma}^{(k)}$, such that

$$-\nabla_\tau \cdot [-\boldsymbol{\mathcal{K}}(p_{h,\gamma}^{(k-1)})\nabla_\tau p_{h,\gamma}^{(k)}] + \mathcal{S}_\gamma(p_{h,\gamma}^{(k)}) = g_\gamma + f_\gamma \quad \text{on } \gamma, \tag{3.3a}$$

which can be discretized with a cell-centred finite volume method, leading to a symmetric and positive definite system that can be solved with a CG method. The CG method can be equipped with a preconditioner being the inverse of the discrete counterpart of the operator $-\nabla_\tau \cdot [-\boldsymbol{\mathcal{K}}(p_{h,\gamma}^{(k-1)})\nabla_\tau]$ (see [19,31] for more details).

Remark 3.2 (*The Total Computational Costs*)**.** The total computational costs in the inner–outer iterative approach (3.1) is dominated by the number of sub-domain solves required. Precisely, the total number of sub-domain solves is given by $\sum_{k=1}^{N_{\text{lin}}} N_{\text{dd}}^k$, where N_{lin} is the number of iterations of the fixed-point procedure as outer-loop algorithm, and N_{dd}^k denotes the number of inner loop domain decomposition iterations (GMRes) at the fixed-point iteration $k \geq 1$.

3.2. Method 2: Newton-GMRes

In the second approach, we propose Newton's method to solve the interface Darcy–Forchheimer problem (2.5). For simplicity of notation, we introduce the following

$$\boldsymbol{\mathcal{K}}^{-1,(k)} := \boldsymbol{\mathcal{K}}^{-1}(\boldsymbol{u}_{h,\gamma}^{(k)}) = \boldsymbol{K}_\gamma^{-1} + \beta_\gamma \boldsymbol{I}|\boldsymbol{u}_{h,\gamma}^{(k)}| \qquad \text{and} \qquad \boldsymbol{\mathcal{K}}_\partial^{-1,(k)} := \frac{\partial \boldsymbol{\mathcal{K}}^{-1,(k)}}{\partial \boldsymbol{u}_{h,\gamma}^{(k)}} = \beta_\gamma \frac{\boldsymbol{u}_{h,\gamma}^{(k)}}{|\boldsymbol{u}_{h,\gamma}^{(k)}|}.$$

The non-linear variational form (2.4) may be rewritten in the following canonical form: find $\boldsymbol{u}_{h,\gamma} \in \boldsymbol{V}_{h,\gamma}$ and $p_{h,\gamma} \in M_{h,\gamma}$, such that

$$\mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] = 0, \quad \forall (\boldsymbol{v}_\gamma q_\gamma) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma},$$

where $\mathcal{F}_\gamma$ is the residual expression from the mixed system given as follows:

$$\begin{aligned}\mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] := {} & \langle \boldsymbol{\mathcal{K}}^{-1}(\boldsymbol{u}_{h,\gamma})\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_\gamma + \langle \mathcal{S}_\gamma(p_{h,\gamma}), q_\gamma\rangle_\gamma \\ & - \langle p_{h,\gamma}, \nabla_\tau \cdot \boldsymbol{v}_\gamma\rangle_\gamma + \langle \nabla_\tau \cdot \boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_\gamma - \langle f_\gamma + g_\gamma, q_\gamma\rangle_\gamma.\end{aligned}$$

In the next step, we calculate the Jacobian given by $\mathcal{J}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}); (\delta\boldsymbol{u}_{h,\gamma}, \delta p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right]$ by taking the Gâteaux variation of the residual $\mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right]$ at $\boldsymbol{u}_{h,\gamma} = \boldsymbol{u}_{h,\gamma}^{(k)}$ and $p_{h,\gamma} = p_{h,\gamma}^{(k)}$ in the directions of $\delta\boldsymbol{u}_{h,\gamma}$ and $\delta p_{h,\gamma}$, respectively. This can be formally obtained by computing

$$\mathcal{J}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}); (\delta\boldsymbol{u}_{h,\gamma}, \delta p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] := \left[\frac{\mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)} + \epsilon\delta\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}^{(k)} + \epsilon\delta p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] - \mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}), (\boldsymbol{v}_\gamma, q_\gamma)\right]}{\epsilon}\right]_{\epsilon\to 0}.$$

This definition yields

$$\mathcal{J}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}); (\delta\boldsymbol{u}_{h,\gamma}, \delta p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] = \langle(\boldsymbol{\mathcal{K}}^{-1,(k)} + \boldsymbol{\mathcal{K}}_\partial^{-1,(k)} \otimes \boldsymbol{u}_{h,\gamma}^{(k)})\delta\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_\gamma + \langle\mathcal{S}_\gamma(\delta p_{h,\gamma}), q_\gamma\rangle_\gamma - \langle\delta p_{h,\gamma}, \nabla_\tau\cdot\boldsymbol{v}_\gamma\rangle_\gamma + \langle\nabla_\tau\cdot\delta\boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_\gamma,$$

where $\otimes$ denotes the standard tensor product. At each Newton iteration, we solve the following linear variational problem: find $(\delta\boldsymbol{u}_{h,\gamma}, \delta p_{h,\gamma}) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma}$, such that

$$\mathcal{J}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}); (\delta\boldsymbol{u}_{h,\gamma}, \delta p_{h,\gamma}), (\boldsymbol{v}_\gamma, q_\gamma)\right] = -\mathcal{F}_\gamma\left[(\boldsymbol{u}_{h,\gamma}^{(k)}, p_{h,\gamma}^{(k)}), (\boldsymbol{v}_\gamma, q_\gamma)\right], \quad \forall(\boldsymbol{v}_\gamma, q_\gamma) \in M_{h,\gamma} \times \boldsymbol{V}_{h,\gamma}. \tag{3.4}$$

In compact form, the linear system for the Newton step $k \geq 0$ has the following mixed structure

$$\begin{bmatrix}\mathcal{J}_\gamma^k & B_\gamma^{\mathrm{T}} \\ B_\gamma & \mathcal{S}_\gamma\end{bmatrix}\begin{bmatrix}\delta\boldsymbol{u}_{h,\gamma} \\ \delta p_{h,\gamma}\end{bmatrix} = \begin{bmatrix}\mathcal{R}_\gamma^{\boldsymbol{u},k} \\ \mathcal{R}_\gamma^{p,k}\end{bmatrix}, \tag{3.5}$$

where $\mathcal{R}_\gamma^{\boldsymbol{u},k}$ and $\mathcal{R}_\gamma^{p,k}$ are the residuals at the current Newton step formed from (3.4).

The interface system (3.5) is then solved with the GMRes iterations (2.8). On each GMRes iteration, we need to evaluate the action of the Robin-to-Neumann operator $\mathcal{S}_\gamma$ using Algorithm 2.1. The solution of the interface problem is therefore obtained in an iterative fashion using the following update equations until the Newton residual reaches some prescribed tolerance:

$$\boldsymbol{u}_{h,\gamma}^{(k+1)} = \boldsymbol{u}_{h,\gamma}^{(k)} + \delta\boldsymbol{u}_{h,\gamma} \qquad \text{and} \qquad p_{h,\gamma}^{(k+1)} = p_{h,\gamma}^{(k)} + \delta p_{h,\gamma}.$$

The result of this iterative approach is then used to infer the solution in the sub-domains using (3.2), which needs two additional sub-domain solves.

Remark 3.3 (*An Alternative to* (3.4))**.** For the mixed Jacobian problem in the fracture (3.4), it is possible to adopt the idea introduced in Remark 3.2 to reduce the computational cost by reformulating (3.4) into a cell-centred finite volume problem with the pressure step $\delta p_{h,\gamma}$ as the sole variable. The resulting system is also symmetric definite and positive and can be solved with the CG method equipped with a local preconditioner.

4. Outer–inner interface iterations with multiscale flux basis

As noticed previously, the dominant computational cost in the above linearization–domain-decomposition procedures comes from the sub-domain solves to evaluate the action of $\mathcal{S}_\gamma$ using Algorithm 2.1 (step 2(b)). We recall that the number of sub-domain solves required by each method is approximately equal to $\sum_{k=1}^{N_{\text{lin}}} N_{\text{dd}}^i$, where N_{lin} is the number of iterations of the linearization procedure, and N_{dd}^k denotes the number of domain decomposition iterations (GMRes or any Krylov solver). Even though all sub-domain solves can be computed in parallel, this still be very costly; first, as the non-linear interface solver may converge very slowly and, second, that at each linearization iteration the condition number of the linearized interface problem ((3.1) for Method 1 and (3.4) for Method 2) is large due to a highly refined mesh.

One way to reduce the computational costs, is to employ the multiscale flux basis, following [18]. The motivation of these techniques in this work stems from eliminating the dependency between the total number of solves and the employed outer–inner procedure on the interface-fracture. This is easily achieved by pre-computing and storing the flux sub-domain responses, called *multiscale flux basis*, associated with each fracture pressure degree of freedom on each sub-domain.

The multiscale flux basis requires solving a fixed number of linear sub-domain solves and permits retrieving the action of $\mathcal{S}_\gamma$ on $M_{h,\gamma}$ by simply taking a linear combination of multiscale flux basis functions. As a result, the number of sub-domains solves is now independent of the used linearization procedure as well as of the used solver for the inner domain decomposition systems. In practice, the number of sub-domains solves will be reduced if $\sum_{k=1}^{N_{\text{lin}}} N_{\text{dd}}^k$ exceeds the maximum number of fracture pressure degrees of freedom on any sub-domain.

4.1. Multiscale flux basis

Following [18], we define $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$ to be the set of basis functions on the interface pressure space $M_{h,\gamma}$, where $\mathcal{N}_{h,\gamma}$ is the number of pressure degrees of freedom on sub-domain γ. As a result, on the fracture-interface, we have

$$p_{h,\gamma} \coloneqq \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} p_{h,\gamma}^\ell \, \Phi_{h,\gamma}^\ell.$$

We compute the multiscale flux basis functions corresponding to $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$ using the following algorithm:

Algorithm 4.1 (*Assembly of the Multiscale Flux Basis*)**.**

1. Enter the basis $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$. Set $\ell = 0$.
2. **Do**
   (a) Increase $\ell \coloneqq \ell + 1$.
   (b) Project $\Phi_{h,\gamma}^\ell$ on the sub-domain boundary, i.e,
   $$\Phi_{h,\gamma}^\ell \xrightarrow{\mathcal{D}_{h,i}} \lambda_{h,i}^\ell.$$
   (c) Solve problem (2.3) in each sub-domain Ω_i with Robin boundary condition $\lambda_{h,i}^\ell$ and with $f = 0$.
   (d) Project the boundary flux onto the mortar space on the fracture, i.e,
   $$-\boldsymbol{u}_h(\lambda_{h,i}^\ell, 0) \cdot \boldsymbol{n}_i \xrightarrow{\mathcal{D}_{h,i}^\top} \Psi_{h,\gamma,i}^\ell$$
   While $\ell \leq \mathcal{N}_{h,\gamma}$.
3. Form the multiscale flux basis for sub-domain Ω_i, i.e,
   $$\left\{ \Psi_{h,\gamma,i}^1, \Psi_{h,\gamma,i}^2, \ldots, \Psi_{h,\gamma,i}^{\mathcal{N}_{h,\gamma}} \right\} \subset M_{h,\gamma}.$$

Once the multiscale flux basis functions are constructed for each sub-domain, the action of interface operator $\mathcal{S}_i^{\text{RtN}}$, and then also the action of $\mathcal{S}_\gamma$ via (2.6), is replaced by a linear combination of the multiscale flux basis functions $\Psi_{h,\gamma,i}^\ell$. Specifically, for an interface datum $\varphi_{h,\gamma} \in M_{h,\gamma}$, we have $\varphi_{h,\gamma} \coloneqq \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \varphi_{h,\gamma}^\ell \, \Phi_{h,\gamma}^\ell$, and for $i = 1, 2$,

$$\mathcal{S}_i^{\text{RtN}}(\varphi_{h,\gamma}, 0) \coloneqq \mathcal{S}_i^{\text{RtN}}\Big(\sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \varphi_{h,\gamma}^\ell \, \Phi_{h,\gamma}^\ell, 0\Big) = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \varphi_{h,\gamma}^\ell \mathcal{S}_i^{\text{RtN}}(\Phi_{h,\gamma}^\ell, 0) = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \varphi_{h,\gamma}^\ell \, \Psi_{h,\gamma,i}^\ell.$$

Remark 4.2 (*DOFs on γ*)**.** We observe that each fracture pressure basis function $\Phi_{h,\gamma}^\ell$ on the fracture-interface corresponds to exactly two different multiscale flux basis functions, one for Ω_1 and one for Ω_2. For the case of a fractures network, say $\gamma \coloneqq \cup_{i \neq j} \gamma_{ij}$, where γ_{ij} is the fracture between the sub-domain Ω_i and Ω_j, the previous basis reconstruction is then applied independently on each fracture.

4.2. Application on intersecting fractures model: solving the DFNs system

In this part, we first introduce and describe the case of intersecting fractures, and then we provide our amendments to the previous algorithms.

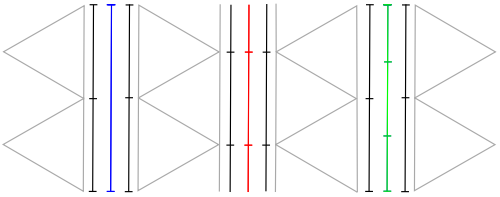


Fig. 2. Representation of three possible fracture mesh configurations: on the left coarser, on the centre conforming, and on the right finer. The triangles are represented in grey.

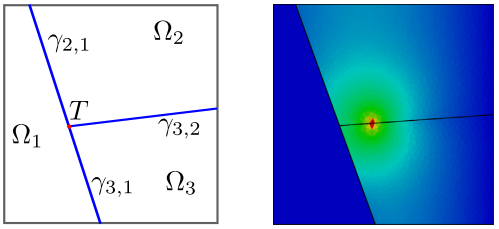


Fig. 3. On the left, graphical example of problem (1.1)–(1.3) along with (4.2) in case of intersecting fractures. On the right, example of construction of a multiscale flux basis.

4.2.1. Mathematical model

For the sake of simplicity, we consider the Darcy–Forchheimer model in a two-dimensional geological domain made up with three sub-domains Ω_i, $i = 1, 2, 3$, physically subdivided by fractures $\gamma_{i,j}$, $1 \leq i < j \leq 3$. The rock matrix is now defined as $\overline{\Omega} := \sum_{i=1}^{3} \overline{\Omega_i}$, $\Omega_i \cap \Omega_j = \emptyset$, where a single fracture is $\gamma_{i,j} := \partial\Omega_i \cap \partial\Omega_j$, all fractures that touch sub-domain Ω_i are $\gamma_i := \partial\Omega_i \setminus \partial\Omega$. Also, $T := \partial\gamma_{1,2} \cap \partial\gamma_{2,3} = \partial\gamma_{2,3} \cap \partial\gamma_{1,3} = \partial\gamma_{1,3} \cap \partial\gamma_{1,2}$ corresponds to the intersection point of the fractures $\gamma_{i,j}$ and $\Gamma_i := \partial\Omega_i \cap \partial\Omega$ the boundary of each sub-domain Ω_i. We impose the Darcy model (1.1) in each sub-domain Ω_i and the Darcy–Forchheimer model (1.2) in each fracture $\gamma_{i,j}$, with unknowns denoted by $(\boldsymbol{u}_{\gamma_{i,j}}, p_{\gamma_{i,j}})$. See Fig. 3 (left) as an example.

They are coupled using the Robin boundary conditions given by

$$-\boldsymbol{u}_i \cdot \boldsymbol{n}_i + \alpha_{i,j} p_i = \alpha_{i,j} p_{\gamma_{i,j}} \quad \text{on } \gamma_{i,j}, \tag{4.1}$$

for $1 \leq i < j \leq 3$, where the coefficient $\alpha_{i,j}$ can now be different in each fracture. To close the system, we need to impose transmission conditions between the fractures at the $(d-2)$-dimensional interface T. On the intersection T, we set, for $1 \leq i < j \leq 3$,

$$-\boldsymbol{u}_{\gamma_{i,j}} \cdot \boldsymbol{n}_{i,j} + \alpha_{\gamma_{i,j}} p_{\gamma_{i,j}} = \alpha_{\gamma_{i,j}} p_T \quad \text{on } T, \tag{4.2a}$$

$$\sum_{1 \leq i < j \leq 3} \boldsymbol{u}_{\gamma_{i,j}} \cdot \boldsymbol{n}_{i,j} = 0 \quad \text{on } T, \tag{4.2b}$$

where $\alpha_{\gamma_{i,j}}$ is a strictly positive constant, and $\boldsymbol{n}_{i,j}$ is the outer unit normal vector to $\partial\gamma_{i,j}$.

For the partition of the sub-domain Ω_i, $1 \leq i \leq 3$, and the fractures $\gamma_{i,j}$, $1 \leq i < j \leq 3$, we extend the notation introduced in Section 2.1. We let $\mathcal{T}_{h,i}$ be a partition of the sub-domain Ω_i into 2-dimensional simplicial elements and let $\mathcal{T}_{h,\gamma_{i,j}}$ to be a partition of the fracture $\gamma_{i,j}$ into 1-dimensional simplicial elements. Again, the meshes $\mathcal{T}_{h,i}$, $1 \leq i \leq 3$, are allowed to be non-conforming on the fractures $\gamma_{i,j}$, $1 \leq i < j \leq 3$, but also different from those used in $\gamma_{i,j}$, $1 \leq i < j \leq 3$ (see Fig. 2 for more details). We also extend the same notation for the approximation spaces in the sub-domains and in the fractures, and additionally we let $M_{h,T}$ be the space endowed with constant functions on T.

4.2.2. Domain decomposition formulation

The extension of the reduced interface problem (2.5) to the present intersecting fractures setting is as follows: find the triplet $(\boldsymbol{u}_{h,\gamma}, p_{h,\gamma}, p_{h,T}) \in \boldsymbol{V}_{h,\gamma} \times M_{h,\gamma} \times M_{h,T}$ such that, for each $1 \leq i < j \leq 3$,

$$\begin{aligned}
&\langle \mathcal{K}^{-1}(\boldsymbol{u}_{h,\gamma})\boldsymbol{u}_{h,\gamma}, \boldsymbol{v}_\gamma\rangle_{\gamma_{i,j}} + \alpha^{-1}_{\gamma_{i,j}}\langle \boldsymbol{u}_{h,\gamma}\cdot\boldsymbol{n}_{i,j}, \boldsymbol{v}_\gamma\cdot\boldsymbol{n}_{i,j}\rangle_T - \langle p_{h,\gamma}, \nabla_\tau\cdot\boldsymbol{v}_\gamma\rangle_{\gamma_{i,j}} \\
&\qquad = -\langle p_{h,T}, \boldsymbol{u}_{\gamma_{i,j}}\cdot\boldsymbol{n}_{i,j}\rangle_T \qquad \forall \boldsymbol{v}_\gamma \in \boldsymbol{V}_{h,\gamma_{i,j}}, \qquad (4.3a)\\
&\langle \nabla_\tau\cdot\boldsymbol{u}_{h,\gamma}, q_\gamma\rangle_{\gamma_{i,j}} + \langle \mathcal{S}_{\gamma_{i,j}}(p_{h,\gamma}), q_\gamma\rangle_{\gamma_{i,j}} = \langle f_{\gamma_{i,j}} + g_{\gamma_{i,j}}, q_\gamma\rangle_{\gamma_{i,j}} \qquad \forall q_\gamma \in M_{h,\gamma_{i,j}}, \qquad (4.3b)\\
&\sum_{1\leq i<j\leq 3}\langle \boldsymbol{u}_{h,\gamma_{i,j}}\cdot\boldsymbol{n}_{i,j}, q_T\rangle_T = 0 \qquad \forall q_T \in M_{h,T}. \qquad (4.3c)
\end{aligned}$$

On each fracture, the Robin-to-Neumann operator $\mathcal{S}_{\gamma_{i,j}}$ and the linear functional $g_{\gamma_{i,j}}$, $1 \leq i < j \leq 3$, are now given by

$$\begin{aligned}
\mathcal{S}_{\gamma_{i,j}}(p_{h,\gamma}) &:= \sum_{l\in(i,j)} \mathcal{S}^{\mathrm{RtN}}_{\gamma l}(p_{h,\gamma}, 0) = -\sum_{l\in(i,j)} \boldsymbol{u}_{h,l}(p_{h,\gamma}, 0)\cdot\boldsymbol{n}_l|_{\gamma_l},\\
g_{\gamma_{i,j}} &:= \sum_{l\in(i,j)} \mathcal{S}^{\mathrm{RtN}}_{\gamma l}(0, f_l) = \sum_{l\in(i,j)} \boldsymbol{u}_{h,l}(0, f_l)\cdot\boldsymbol{n}_l|_{\gamma_i}.
\end{aligned}$$

The above problem can be seen as a DFNs system on the set of fractures, and as a domain decomposition problem between the 1-dimensional fractures $\gamma_{i,j}$, $1 \leq i < j \leq 3$, cf. [16,17,22] for more details.

4.2.3. Iterative procedure

We propose to solve the non-linear domain decomposition problem (4.3) using the fixed-point approach in Section 3.1. This iterative process is now equipped with the multiscale flux basis of Section 4 to lessen the interface iterations. To this aim, we introduce

$$\mathcal{S}_\gamma(p_{h,\gamma}) := \sum_{0\leq i<j\leq 3} \mathcal{S}_{\gamma_{i,j}}(p_{h,\gamma}) \quad \text{and} \quad g_\gamma := \sum_{1\leq i<j\leq 3} g_{\gamma_{i,j}},$$

and let

$$\mathcal{S}_T(p_{h,T}) := \sum_{1\leq i<j\leq 3} \boldsymbol{u}_{h,\gamma_{i,j}}\cdot\boldsymbol{n}_{i,j}|_T.$$

Applying the fixed-point algorithm on the set of interface Darcy–Forchheimer equations (4.3) can be interpreted as follows: at the iteration $k \geq 1$, we solve

$$\begin{bmatrix} \mathcal{K}_\gamma^{-1,(k)} & B_\gamma^{\mathrm{T}} & \mathcal{S}_T^{\mathrm{T}} \\ B_\gamma & \mathcal{S}_\gamma & 0 \\ \mathcal{S}_T & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^k_{h,\gamma} \\ p^k_{h,\gamma} \\ p^k_{h,T} \end{bmatrix} = \begin{bmatrix} 0 \\ f_\gamma + g_\gamma \\ 0 \end{bmatrix}, \qquad (4.4)$$

using GMRes method until a fixed tolerance is reached. Again, the evaluation of $\mathcal{S}_\gamma$ in each interface GMRes iteration dominates the total computational costs of this outer–inner procedure. Note that each inner iteration also requires the evaluation of the Dirichlet-to-Neumann operator $\mathcal{S}_T$, which requires solves in the fractures. The complete algorithm when equipped with multiscale flux basis is now given by the following algorithm.

Algorithm 4.3 (*Fixed-point Algorithm with Multiscale Flux Basis for Fracture Network Model*)**.**

1. Enter the source terms and the permeabilities in the fractures and the rock matrices.
2. Choose the meshes $\mathcal{T}_{h,i}$, $1 \leq i \leq 3$, and $\mathcal{T}_{h,\gamma_{i,j}}$, $1 \leq i < j \leq 3$.
3. Calculate the right-hand-sides $g_{\gamma_{i,j}}$, $1 \leq i < j \leq 3$, by solving the Darcy sub-domain problem in Ω_i with source term f_i and zero Robin value on the fracture-interface γ_i. Then, compute the resulting jump across all sub-domain interfaces.
4. In the sub-domain Ω_i, $1 \leq i \leq 3$, let $\mathcal{N}_{h,\gamma_i}$ be the number of degrees of freedom in the space M_{h,γ_i}. Define the basis $(\Phi^\ell_{h,\gamma_i})_{\ell=1}^{\mathcal{N}_{h,\gamma_i}}$. Set $i = 0$.
   Do {Assembly of the multiscale flux basis}

(a) Increase $i := i + 1$.

(b) Compute the multiscale flux basis functions $(\Psi^{\ell}_{h,\gamma_i})_{\ell=1}^{\mathcal{N}_{h,\gamma_i}}$ corresponding to $(\Phi^{\ell}_{h,\gamma_i})_{\ell=1}^{\mathcal{N}_{h,\gamma_i}}$ using Algorithm 4.1, i.e.,

$$\Psi^{\ell}_{h,\gamma_i} := \mathcal{S}^{\mathrm{RtN}}_{\gamma_l}(\Phi^{\ell}_{h,\gamma_i}, 0), \quad \ell = 1, \dots, \mathcal{N}_{h,\gamma_i}.$$

While $i \leq 3$.

5. Given an initial guess $\boldsymbol{u}^{(0)}_{h,\gamma_{i,j}}$, $1 \leq i < j \leq 3$. Set $k = 0$.
Do {Fixed-point iterations}

(a) Increase $k := k + 1$.

(b) Solve the linear system on the fractures (4.4) using GMRes method (2.8), where in every iteration the operator action $\mathcal{S}_{\gamma}$ on any $\varphi_{h,\gamma} \in M_{h,\gamma}$ is computed with the following steps:

i. Use a linear combination of the multiscale flux basis to compute the action of $\mathcal{S}^{\mathrm{RtN}}_{\gamma_i}$ by

$$\mathcal{S}^{\mathrm{RtN}}_{\gamma_i}(\varphi_{h,\gamma,i}, 0) = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma,i}} \varphi^{\ell}_{h,\gamma,i} \Psi^{\ell}_{h,\gamma,i}.$$

ii. Compute the jump across all the fractures:

$$\mathcal{S}_{\gamma}(\varphi_{h,\gamma}) = \sum_{0 \leq i < j \leq 3} \sum_{l \in (i,j)} \mathcal{S}^{\mathrm{RtN}}_{\gamma_l}(\varphi_{h,\gamma}, 0).$$

$$\textbf{While } \frac{\|(p^{k,\infty}_{h,\gamma}, \boldsymbol{u}^{k,\infty}_{h,\gamma}) - (p^{k-1,\infty}_{h,\gamma}, \boldsymbol{u}^{k-1,\infty}_{h,\gamma})\|_{\infty}}{\|(p^{k-1,\infty}_{h,\gamma}, \boldsymbol{u}^{k-1,\infty}_{h,\gamma})\|_{\infty}} \geq \varepsilon_{tol}. \tag{4.5}$$

5. Numerical examples

In this section, we validate the model and analysis presented in the previous parts by means of numerical test cases. We have chosen three examples designed to show how the proposed linearization–domain-decomposition approaches equipped with multiscale flux basis behaves versus the standard ones in various physical and geometrical situations. To compare these approaches, the main criteria considers the number of solutions of the higher-dimensional sub-problems since it constitutes the major computational cost. We consider solving the problem in the network of fractures as negligible. Since each of the higher-dimensional sub-problem is linear and will be solved many times, we consider an LU-factorization of the system matrix and a forward–backward substitution algorithm to compute the numerical solution. It results in a computational cost reduced to $\mathcal{O}(n^2)$ flops each time, where n is the size of the matrix. For bigger systems, an iterative scheme is preferable.

We use the PorePy [24] library, which is a simulation tool for fractured and deformable porous media written in Python. PorePy uses SciPy [32] as default sparse linear algebra. All the examples are reported in the GitHub repository of PorePy, we want to stress again that even if we focus on lowest-order Raviart–Thomas–Néd élec finite elements, our implementation is agnostic with respect to the numerical scheme.

For the multiscale flux basis scheme presented in Section 4, for a fixed rock matrix grid and normal fracture permeability it is possible to compute once all the basis functions. The results in the next parts should be read under this important property of the method, thus in many cases only a pure fracture network will be solved at a negligible computational cost. The multiscale basis functions are computed and stored in an *offline* phase prior the simulation (called *online*).

Unless otherwise noted, the tolerance for the relative residual in the inner GMRes algorithm is taken to be 10^{-6}. The same tolerance is chosen for the outer Newton/fixed-point algorithm. We consider an LU-factorization of the fracture network matrix [33,34] as the preconditioner of the GMRes method. To illustrate the efficiency of the proposed new implementation while preserving the accuracy of the MMMFEM, in Section 5.4 we distinguish two cases: for high-permeable fractures, the mortar mesh is assumed to be conforming with the fine mesh in the rock matrix, while for low-permeable fractures, a coarse scale of the mortar mesh is considered.

In the examples, we use the abbreviation MS when the linearization–domain-decomposition approach is equipped with multiscale flux basis techniques, and DD for the corresponding classical approach.

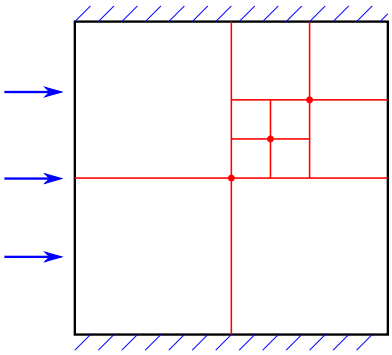


Fig. 4. Graphical representation of the domain and fracture network geometry (resulting into 10 sub-domains) common for all test cases.

Table 1
Definition of the cases for the examples.

	$\boldsymbol{K}_\gamma$	α_γ
case (i)	$10^4 \epsilon \boldsymbol{I}$	$10^4/\epsilon$
case (ii)	$10^{-4} \epsilon \boldsymbol{I}$	$10^{-4}/\epsilon$
case (iii)	$10^4 \epsilon \boldsymbol{I}$	$10^{-4}/\epsilon$
case (iv)	$10^{-4} \epsilon \boldsymbol{I}$	$10^4/\epsilon$

Remark 5.1 (*Fracture Aperture*)**.** Even if not explicitly considered in the previous parts of the work, we introduce the fracture aperture ϵ as a constant parameter. This choice is based on the fact that geometries and (some) data of the forthcoming examples are taken from the literature.

In 5.1 we describe the geometry and some data of the problem considered. Few subsections follow with an increase level of challenge: linear case in 5.2, Forchheimer model in 5.3, Forchheimer model with heterogeneous parameters in 5.4, to conclude with a generalized Forchheimer model in 5.5.

5.1. Problem setting

To validate the performance of the two proposed algorithms, we consider the first problem presented in the benchmark study [35]. The unit square domain Ω, depicted in Fig. 4, has unitary permeability of the rock matrix and it is divided into 10 sub-domains by a set of fractures with fixed aperture ϵ equal to 10^{-4}. At the boundary, we impose zero flux condition on the top and bottom, unitary pressure on the right, and flux equal to -1 on the left. The boundary conditions are applied to both the rock matrix and the fracture network.

Contrary to what has been done in the benchmark paper, we consider four different scenarios for the fracture permeabilities, by having high or low values in the tangential and normal parts. Thus, we have the *case (i)* with high permeable fractures, *case (ii)* has low permeable fractures, while *cases (iii)* and *(iv)* have mixed high and low permeability in normal and tangential directions. See Table 1 for a summary of the fracture permeability in each case. *Case (i)* and *(ii)* have the same permeabilities used in the benchmark paper [35].

In the following examples, we consider the maximal number of rock matrix solves to be 10^4, and we mark with ∞ if this is exceeded.

5.2. Darcy model: $\beta_\gamma = 0$

The first example considers the Forchheimer coefficient set to zero, thus the problem becoming linear. The results for different level of discretization are reported in Table 2. We indicate by *level 1* a grid with a total of 110 triangles and 26 mortar edges, *level 2* with 1544 triangles and 84 mortar edges, and *level 3* with 3906 triangles and 138 mortar edges.

Table 2
Total number of the higher-dimensional problem solves for the case study of example in Section 5.2. For each level of refinement cases marked in † share the same multiscale flux basis, which can be constructed only once. The same is valid for §.

	level 1		*level 2*		*level 3*	
	MS	DD	MS	DD	MS	DD
case (i)	$28^{\dagger}$	10	$86^{\dagger}$	11	$140^{\dagger}$	11
case (ii)	$28^{\S}$	81	$86^{\S}$	112	$140^{\S}$	189
case (iii)	$28^{\S}$	22	$86^{\S}$	28	$140^{\S}$	29
case (iv)	$28^{\dagger}$	82	$86^{\dagger}$	61	$140^{\dagger}$	86

Table 3
Total number of the higher-dimensional problem solves for the case study of example in Section 5.2. For each level of refinement we change the convergence tolerance for the domain decomposition method.

Tolerance	*level 1*		*level 2*		*level 3*	
	10^{-4}	10^{-8}	10^{-4}	10^{-8}	10^{-4}	10^{-8}
case (i)	8	11	9	12	9	12
case (ii)	42	105	82	∞	150	∞
case (iii)	21	30	22	36	22	36
case (iv)	42	122	50	∞	70	∞

Table 2 shows the results of this example for the physical considerations of Table 1. We notice that for high permeable fractures (*case (i)* and *(iii)*), the standard domain decomposition method performs better than our method with multiscale flux basis, while the opposite occurs for low permeable fractures. A possible explanation is related to the ratio between normal and tangential permeability. The normal permeability determines how strong the flux exchange is between the rock matrix and the fractures (thus, the communications at each DD iteration), while for small values of the tangential permeability the fractures are more influenced by the surrounding rock matrices. The opposite occurs in the case of high tangential permeability. Additionally, the choice of the preconditioner for DD slightly goes in favour of high permeable fractures due to the dominating role of the fracture flow in the system. We also recall that the number of higher-dimensional problem solves does not depend on the number of outer–inner interface iterations, but only on the number of local mortar degrees of freedom on the fractures network. A further important result in this experiments, is that *case (i)* and *(iv)* share the same value of α_γ, thus the multiscale flux basis are computed only once per level of refinement. The same applies to *case (ii)* and *(iii)*. As a result, the developed method is globally more efficient than the classical approach. That is, the results in Table 2 show a reduction of the number of the higher-dimensional problem solves from 195 to 56 for *level 1*, from 212 to 186 for *level 2*, and from 312 to 280 for *level 3*. Note that the two methods produce the same solution for all the cases, within the same relative convergence tolerance. The numerical solution for all cases is reported in Fig. 5.

The next series of numerical experiments aims at assessing the stability of the domain decomposition approach with respect to GMRes tolerance. The multiscale flux basis approach provides the extra flexibility to do such analysis with negligible costs, by reusing the stored multiscale flux basis used for the results of Table 2 but now with different tolerance for GMRes. Further, this set of test cases aims assessing how the overall gain for an entire simulation in terms of number of higher-dimensional problem solves can be appreciated or depreciated with more or less stringent stopping criteria for GMRes; this is a preparatory step to address the complete approaches of Section 3 for the full non-linear problem, which requires several solves of linear Darcy problems, for which one should formulate the stopping criteria very carefully. In Table 2, we have considered the relative residual to be below 10^{-6}, while in Table 3 we present the results in the case of 10^{-4} and 10^{-8}. Based on the results of Table 3, we can conclude that even with less stringent criterion, a considerable gain in terms of number of higher-dimensional problem solves can be achieved. We also see that all the results are free of oscillations and neither the fracture, barrier, or the very different tangential and normal permeabilities pose any problems for the domain decomposition approach. Based on the above results and in what follows we consider 10^{-6} as tolerance for the GMRes algorithm.

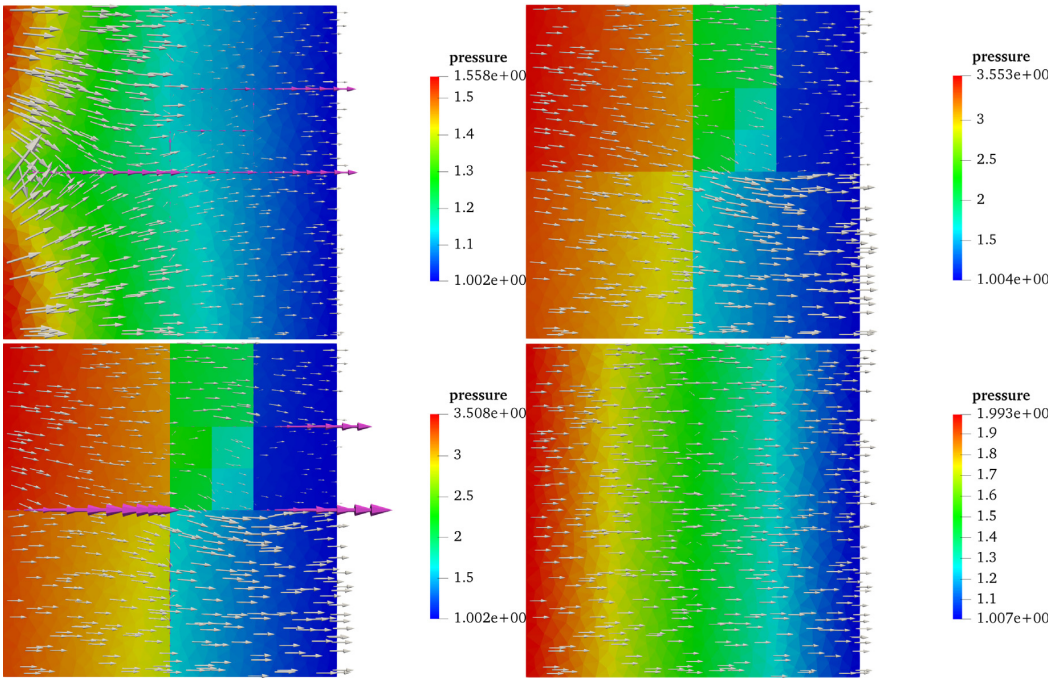


Fig. 5. Pressure and velocity solutions for the four cases: on the top-left *case (i)*, on the top-right *case (ii)*, on the bottom-left *case (iii)*, and on the bottom-right *case (iv)*. In all the cases, the velocity is represented by arrows (purple for the fractures) proportional to its magnitude. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Total number of the higher-dimensional problem solves required by Method 1 for the case study in Section 5.3. The number of the fixed-point iterations are in brackets. Within each case the construction of the multiscale flux basis is done only once, we mark by † (respectively §) common computations.

β_γ	*case (i)*		*case (iii)*	
	MS	DD	MS	DD
1	$86^\dagger$ (2)	33 (2)	$86^\S$ (1)	56 (1)
10^2	$86^\dagger$ (3)	44 (3)	$86^\S$ (2)	84 (2)
10^4	$86^\dagger$ (8)	99 (8)	$86^\S$ (3)	115 (3)
10^6	$86^\dagger$ (94)	1424 (94)	$86^\S$ (11)	457 (11)

5.3. Forchheimer model

In this second example we consider *case (i)* and *(iii)* for the fracture permeabilities since the Forchheimer model requires high permeable fractures. In this problem, we fix the computational grid *level 2* of Table 2 and we change the value of β_γ in order to compare the performances of Method 1 and Method 2 with and without multiscale flux basis. The Forchheimer coefficient here varies as $\{1, 10^2, 10^4, 10^6\}$. These values are reasonable since in our model we do not explicitly scale β_γ by the aperture, as done in [2,3]. Therefore, the last two values are more realistic. The stopping criteria for both methods is based on the relative residual criteria (4.5) with a threshold fixed as 10^{-6}. The initial guess is taken by solving the linear Darcy by taking β_γ equal to zero.

For Method 1, the number of higher-dimensional problem solves is reported in Table 4. As expected, Method 1 equipped with multiscale flux basis (MS) performs all the higher-dimensional problem solves in the offline phase,

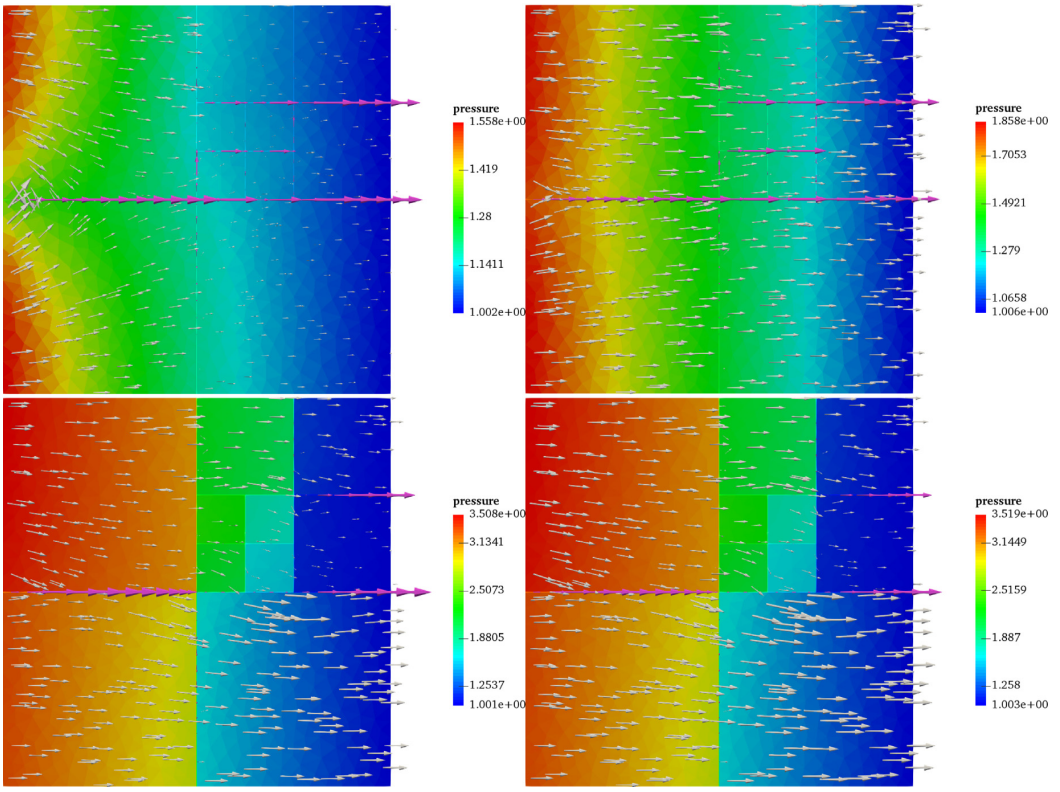


Fig. 6. Pressure and velocity solutions for different configurations of example presented in Section 5.3. On the top *case (i)* and on the bottom for *case (iii)*. On the left, we consider value of the Forchheimer coefficient equal to $\beta_\gamma = 1$ and on the right a high value $\beta_\gamma = 10^6$. In all the cases, the velocity is represented by arrows (purple for the fractures) proportional to its magnitude. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

thus the outer–inner interface iterations for the resulting fracture network problem do not influence the total computational costs. On the contrary, the computational costs of the classical approach (DD) is influenced by the non-linearity, by varying β_γ, as well as by the ratio of the normal and tangential permeabilities, by varying $\boldsymbol{K}_\gamma$ and α_γ. Particularly, the total gain of the new approach is more significant when the non-linear effects becomes more important (by increasing the value of β_γ). Furthermore, for the entire simulation of each case of Table 4, the multiscale flux basis are computed only once. As a conclusion, the entire simulation of *case (i)* required for Method 1 1600 higher-dimensional problem solves, while for Method 1 with multiscale flux basis, this number is reduced by 95%. For *case (iii)*, we reduce the computational costs by 88%.

The numerical solution for two values of β_γ is reported in Fig. 6 for both cases. Despite the different values of β_γ, we notice that the graphical results are very similar in the case of low α_γ. While for high value of α_γ, the resulting apparent permeability given by $\boldsymbol{K}_\gamma(1 + \boldsymbol{K}_\gamma^{-1}\beta_\gamma|\boldsymbol{u}_\gamma|)^{-1}$ decreases (for a fixed $|\boldsymbol{u}_\gamma|$) and the fractures are less prone to be the main path for the flow. Also as stated previously, since we do not explicitly scale β_γ by the aperture, values of $\beta_\gamma > 10^4$ are more likely for real applications.

For Method 2, involving Newton's method for the linearization step, the number of higher-dimensional problem solves is reported in Table 5. As expected, Method 2 is more efficient than Method 1 in terms of the number of higher-dimensional problem solves required, regardless of using multiscale flux basis in the domain decomposition algorithm. Again, the number of solves for the classical approach(DD) depends on the used parameters. This table demonstrates (as shown with Method 1) that as the value of β_γ is increased, there is a point after which Method 2

Table 5
Total number of the higher-dimensional problem solves required by Method 2 for the case study in Section 5.3. The number of the Newton iterations is in brackets. Within each case the construction of the multiscale flux basis is done only once, we mark by † (respectively §) common computations.

β_γ	*case (i)*		*case (iii)*	
	MS	DD	MS	DD
1	$86^{\dagger}$ (2)	20 (2)	$86^{\S}$ (1)	38 (1)
10^2	$86^{\dagger}$ (2)	20 (2)	$86^{\S}$ (2)	71 (2)
10^4	$86^{\dagger}$ (3)	31 (3)	$86^{\S}$ (2)	71 (2)
10^6	$86^{\dagger}$ (7)	128 (7)	$86^{\S}$ (4)	266 (6)

Table 6
Total number of the higher-dimensional problem solves required by Method 1 for the case study in Section 5.4. The number of the fixed-point iterations are in brackets. Within each case the construction of the multiscale basis is done only once, we mark by † common computations. A coarse scale of the mortar mesh is used on every low-permeable fracture.

β_γ	MS	DD
1	$62^{\dagger}$ (2)	63 (2)
10^2	$62^{\dagger}$ (3)	84 (3)
10^4	$62^{\dagger}$ (8)	189 (8)
10^6	$62^{\dagger}$ (64)	2372 (64)

with multiscale flux basis is more efficient than without multiscale flux basis. In that case, the gain in the number of solves becomes more significant when decreasing the value of α_γ. Note that, in practice, the simulations for Method 2 with multiscale flux basis are performed with negligible computational costs as we reused the flux basis inherited from Method 1. This point together with the fact that the total number of solves required by the entire simulation of *case (i)* is now reduced by 57% as well as that of *case (i)* is reduced by 80% showcase the performance of Method 2 with multiscale flux basis.

To sum up, equipping Method 1 and 2 with multiscale flux basis leads to powerful tools to solve complex fracture network with important savings in terms of the number of higher-dimensional problem solves. Note that, as known, one limitation of Method 2 involving Newton method is that a good initial value is usually required to obtain a solution. A good combination of both methods can also be used, in which one can perform first some fixed-point iterations and then switch to Newton method. Concerning the computational costs, let us point out that the fixed-point algorithm of Method 1 requires at each iteration the assembly of the matrix corresponding to the linearization of the Darcy–Forchheimer equations and the solution of a linear system. The Newton method in Method 2 is slightly more expensive since one has to assemble two matrices at each iteration and to update the right-hand side.

5.4. Heterogeneous Forchheimer model

In this example we assign high permeability to the two largest fractures (one horizontal and one vertical) while to the others low permeability. For the highly permeable fractures we adopt the physical parameters of *case (i)*, while for those with lower permeabilities, the physical parameters corresponding to *case (ii)* together with zero Forchheimer coefficient. In this case, we want to test the applicability of Method 1 with and without multiscale flux basis on highly heterogeneous setting for both the permeability and the flow models. We then consider *level 2* for the computation and, subsequently, use a coarse scale of the local grids of the low-permeable fractures, i.e, the number of coarse scale elements being the half of the original number of elements from the fine scale (on the subdomains); this results in 60 mortar elements instead of 84.

As usually, we compare the method with and without multiscale flux basis in terms of the number of higher-dimensional problem solves. The results are represented in Table 6. In the present setting, we can see that the classical approach is outperformed with the approach equipped with multiscale flux basis, particularly, the total

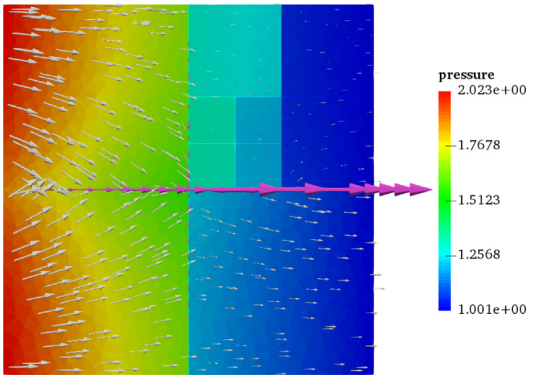


Fig. 7. Pressure and velocity solutions for example presented in Section 5.4 for $\beta = 10^2$. The velocity is represented by arrows (purple for the fractures) proportional to its magnitude. A coarse scale on the mortar mesh is used on every low-permeable fracture. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

computational costs is drastically reduced when the non-linear effects become more important. The entire simulation of Table 6 required 2708 higher-dimensional problem solves for the classical approach while the same approach equipped with multiscale flux basis required 62 solves. The overall gain is then of 94% which can also be appreciated for level 3. Similar conclusions as above can be drawn for Method 2, namely in terms of reduction of the solves (not shown). An example of solution is given in Fig. 7.

5.5. Generalized Forchheimer model

As stated previously, another advantage distinguishes our approach is that it can integrate easily more complex problems. Here, we apply our procedure to a more general model describing the pressure–flow relation in the fractures. Precisely, for larger fracture flow velocities, the drag forces (in the Forchheimer model proportional to the velocity norm) require to consider an additional term proportional to the fluid viscosity. Considering the Barus formula [36], we have an exponential relation between the fluid viscosity and the pressure. We consider problem (2.1) where the non-linear term is as follows

$$\mathcal{K}^{-1}(\boldsymbol{u}_\gamma, p_\gamma) := \boldsymbol{K}_\gamma^{-1} e^{\zeta p_\gamma} + \beta_\gamma \boldsymbol{I} |\boldsymbol{u}_\gamma|,$$

where ζ being a model parameter. Thus, the non-linear effects are now dependent on both the pressure and the velocity. For a more detailed discussion we refer to [37]. For the present setting, the fracture permeabilities are set as in *case (i)* and *(iii)* of Table 1.

For the discretization of the mixed geometry, we consider *level 2*. We use Method 1 with and without multiscale basis functions. Also, it was not necessary to recompute the basis functions, since we can reuse the stored multiscale flux basis from the previous test case and solve then only on the fracture network the above more complex Darcy–Forchheimer model. The total number the higher-dimensional problem solves for $\beta_\gamma = 20$ and $\zeta \in (0.5, 5, 7.5)$ is reported in Table 7. As expected, for such a strong non-linearity, the results show that a considerable gain in terms of higher-dimensional problem solves can be achieved. Particularly, for large values of ζ the classical approach becomes uncompetitive to the new approach. In Fig. 8 we report the solution for $\zeta = 5$.

6. Conclusions

In this work, we have presented a strategy to speed up the computation of a Darcy–Forchheimer model for flow and pressure in fractured porous media by means of multiscale flux basis, that represents the inter-dimensional flux exchange. The scheme transforms a computationally expensive discrete fracture model to a more affordable discrete fracture network, where in the latter only a co-dimensional problem is solved. The multiscale flux basis is computed in an *offline* stage of the simulation and, despite the particular choice done in this paper, are completely

Table 7
Total number of the higher-dimensional problem solves required by Method 1 for the case study in Section 5.5. The number of the fixed-point iterations is in brackets. Within each case the construction of the multiscale basis is done only once, we mark by † (respectively §) common computations.

ζ	*case (i)*		*case (iii)*	
	MS	DD	MS	DD
0.5	$86^{\dagger}$ (5)	71 (5)	$86^{\S}$ (4)	176 (4)
5	$86^{\dagger}$ (4)	648 (4)	$86^{\S}$ (6)	∞
7.5	$86^{\dagger}$ (3)	5317 (3)	$86^{\S}$ (4)	∞

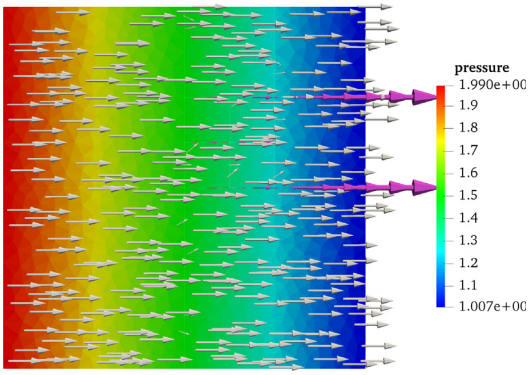


Fig. 8. Pressure and velocity solutions for example presented in Section 5.5 for $\zeta = 5$. The velocity is represented by arrows (purple for the fractures) proportional to its magnitude. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

agnostic to the model in the fracture network. The numerical results show the speed-up gain compared to a more classical linearization–domain-decomposition approaches, where solves in both the matrix and the fracture network are required along the entire outer–inner iterative method. Crucially, an important number of the outer–inner interface iterations may be spared.

With the proposed approach we are able to predict the computational effort needed to solve the problem since it is directly related to the number of mortar grids in the fracture network. Furthermore, the multiscale flux basis can be reused when the fracture network geometry, rock matrix properties, and normal permeability are fixed. Theoretical findings and numerical results show the validity of the proposed approach and of its aforementioned properties.

Even if not explicitly considered in this work, it is possible to further increase the efficiency of the proposed scheme by the following two steps. First, compute a multiscale flux basis only in the related connected part of the rock matrix. Second, use an adaptive stopping criteria for the inner–outer iterative method based on a posteriori error estimates. These enhancements are a part of future work along with the extension in three-dimensions.

Acknowledgements

We acknowledge the financial support from the Research Council of Norway for the TheMSES project (project no. 250223) and the ANIGMA project (project no. 244129/E20) through the ENERGIX program. The authors also warmly thank Eirik Keilegavlen for valuable comments and discussions on this topic.

References

[1] N. Frih, J.E. Roberts, A. Saada, Modeling fractures as interfaces: a model for forchheimer fractures, Comput. Geosci. 12 (1) (2008) 91–104, http://dx.doi.org/10.1007/s10596-007-9062-x.

[2] P. Knabner, J.E. Roberts, Mathematical analysis of a discrete fracture model coupling darcy flow in the matrix with darcy-forchheimer flow in the fracture, ESAIM Math. Model. Numer. Anal. 48 (2014) 1451–1472, http://dx.doi.org/10.1051/m2an/2014003, http://www.esaim-m2an.org/article_S0764583X1400003X.

[3] N. Frih, J.E. Roberts, A. Saada, Modeling fractures as interfaces: a model for forchheimer fractures, Comput. Geosci. 12 (1) (2008) 91–104, http://dx.doi.org/10.1007/s10596-007-9062-x.
[4] C. Alboin, J. Roberts, C. Serres, Domain decomposition for flow in porous media with fractures, in: Proceeding of the 11th International Conference on Domain Decomposition Methods in Greenwich, 1999.
[5] P. Angot, F. Boyer, F. Hubert, Asymptotic and numerical modelling of flows in fractured porous media, M2AN Math. Model. Numer. Anal. 43 (2) (2009) 239–275, http://dx.doi.org/10.1051/m2an/2008052.
[6] M. Lesinigo, C. D'Angelo, A. Quarteroni, A multiscale Darcy-Brinkman model for fluid flow in fractured porous media, Numer. Math. 117 (4) (2011) 717–752, http://dx.doi.org/10.1007/s00211-010-0343-2.
[7] F. Morales, R.E. Showalter, The narrow fracture approximation by channeled flow, J. Math. Anal. Appl. 365 (1) (2010) 320–331, http://dx.doi.org/10.1016/j.jmaa.2009.10.042.
[8] I. Faille, A. Fumagalli, J. Jaffré, J.E. Roberts, Model reduction and discretization using hybrid finite volumes for flow in porous media containing faults, Comput. Geosci. 20 (2) (2016) 317–339, http://dx.doi.org/10.1007/s10596-016-9558-3.
[9] V. Martin, J. Jaffré, J.E. Roberts, Modeling fractures and barriers as interfaces for flow in porous media, SIAM J. Sci. Comput. 26 (5) (2005) 1667–1691, http://dx.doi.org/10.1137/S1064827503429363.
[10] H. Hægland, A. Assteerawatt, H. Dahle, G. Eigestad, R. Helmig, Comparison of cell- and vertex-centered discretization methods for flow in a two-dimensional discrete-fracture–matrix system, Adv. Water Resour. 32 (12) (2009) 1740–1755, http://dx.doi.org/10.1016/j.advwatres.2009.09.006, http://www.sciencedirect.com/science/article/pii/S030917080900150X.
[11] M. Del Pra, A. Fumagalli, A. Scotti, Well posedness of fully coupled fracture/bulk Darcy flow with XFEM, SIAM J. Numer. Anal. 55 (2) (2017) 785–811, http://dx.doi.org/10.1137/15M1022574.
[12] P.F. Antonietti, L. Formaggia, A. Scotti, M. Verani, N. Verzott, Mimetic finite difference approximation of flows in fractured porous media, ESAIM Math. Model. Numer. Anal. 50 (3) (2016) 809–832, http://dx.doi.org/10.1051/m2an/2015087.
[13] W. Liu, Z. Sun, A block-centered finite difference method for reduced fracture model in karst aquifer system, Comput. Math. Appl. 74 (6) (2017) 1455–1470, http://dx.doi.org/10.1016/j.camwa.2017.06.028.
[14] C. D'Angelo, A. Scotti, A mixed finite element method for darcy flow in fractured porous media with non-matching grids, ESAIM Math. Model. Numer. Anal. 46 (2) (2012) 465–489, http://dx.doi.org/10.1051/m2an/2011148.
[15] N. Frih, V. Martin, J.E. Roberts, A. Saâda, Modeling fractures as interfaces with nonmatching grids, Comput. Geosci. 16 (4) (2012) 1043–1060, http://dx.doi.org/10.1007/s10596-012-9302-6.
[16] G. Pichot, J. Erhel, J.R. de Dreuzy, A mixed hybrid mortar method for solving flow in discrete fracture networks, Appl. Anal. 89 (10) (2010) 1629–1643, http://dx.doi.org/10.1080/00036811.2010.495333.
[17] G. Pichot, J. Erhel, J.-R. de Dreuzy, A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks, SIAM J. Sci. Comput. 34 (1) (2012) B86–B105, http://dx.doi.org/10.1137/100804383.
[18] B. Ganis, I. Yotov, Implementation of a mortar mixed finite element method using a multiscale flux basis, Comput. Methods Appl. Mech. Engrg. 198 (49) (2009) 3989–3998, http://dx.doi.org/10.1016/j.cma.2009.09.009, http://www.sciencedirect.com/science/article/pii/S0045782509003077.
[19] L. Amir, M. Kern, J.E. Roberts, V. Martin, Décomposition de domaine pour un milieu poreux fracturé: un modèle en 3D avec fractures, Rev. Africaine Rech. Inf. Math. Appl. 5 (2006) 11–25.
[20] T.-T.-P. Hoang, C. Japhet, M. Kern, J.E. Roberts, Space-time domain decomposition for reduced fracture models in mixed formulation, SIAM J. Numer. Anal. 54 (1) (2016) 288–316, http://dx.doi.org/10.1137/15M1009651.
[21] G.V. Pencheva, M. Vohralík, M.F. Wheeler, T. Wildey, Robust a posteriori error control and adaptivity for multiscale, multinumerics, and mortar coupling, SIAM J. Numer. Anal. 51 (1) (2013) 526–554, http://dx.doi.org/10.1137/110839047.
[22] A. Fumagalli, E. Keilegavlen, Dual virtual element method for discrete fractures networks, SIAM J. Sci. Comput. 40 (1) (2018) B228–B258, http://dx.doi.org/10.1137/16M1098231.
[23] L. Formaggia, A. Fumagalli, A. Scotti, P. Ruffo, A reduced model for darcy's problem in networks of fractures, ESAIM Math. Model. Numer. Anal. 48 (4) (2014) 1089–1116, http://dx.doi.org/10.1051/m2an/2013132.
[24] E. Keilegavlen, A. Fumagalli, R. Berge, I. Stefansson, I. Berre, PorePy: An Open Source Simulation Tool for Flow and Transport in Deformable Fractured Rocks, 2017, arXiv:1712.00460 [cs.CE]. URL https://arxiv.org/abs/1712.00460.
[25] Y. Efendiev, T.Y. Hou, in: M. Brodsky (Ed.), Multiscale Finite Element Methods: Theory and Applications, in: Surveys and Tutorials in the Applied Mathematical Sciences, Springer-Verlag, 2009.
[26] R. Eymard, T. Gallouët, R. Herbin, $\mathcal{RT}_k$ mixed finite elements for some nonlinear problems, Math. Comput. Simulation 118 (2015) 186–197, http://dx.doi.org/10.1016/j.matcom.2014.11.013.
[27] B. Ganis, G. Pencheva, M.F. Wheeler, T. Wildey, I. Yotov, A frozen jacobian multiscale mortar preconditioner for nonlinear interface operators, Multiscale Model. Simul. 10 (3) (2012) 853–873, http://dx.doi.org/10.1137/110826643.
[28] I. Yotov, Interface solvers and preconditioners of domain decomposition type for multiphase flow in multiblock porous media, in: Scientific Computing and Applications (Kananaskis, AB, 2000), in: Adv. Comput. Theory Pract., vol. 7, Nova Sci. Publ., Huntington, NY, 2001, pp. 157–167.
[29] H. López, B. Molina, J.J. Salas, Comparison between different numerical discretizations for a Darcy-Forchheimer model, Electron. Trans. Numer. Anal. 34 (2008/09) 187–203.
[30] H. Pan, H. Rui, Mixed element method for two-dimensional Darcy-Forchheimer model, J. Sci. Comput. 52 (3) (2012) 563–587, http://dx.doi.org/10.1007/s10915-011-9558-3.
[31] E. Ahmed, J. Jaffré, J.E. Roberts, A reduced fracture model for two-phase flow with different rock types, Math. Comput. Simulation 137 (2017) 49–70, http://dx.doi.org/10.1016/j.matcom.2016.10.005.
[32] E. Jones, T. Oliphant, P. Peterson, et al., SciPy: Open Source Scientific Tools for Python, 2001–, [Online]. http://www.scipy.org/.

[33] M. Benzi, Preconditioning techniques for large linear systems: a survey, J. Comput. Phys. 182 (2) (2002) 418–477, http://dx.doi.org/10.1006/jcph.2002.7176.
[34] J.W. Demmel, N.J. Higham, R.S. Schreiber, Stability of block *LU* factorization, Numer. Linear Algebra Appl. (2) (1995) 173–190, http://dx.doi.org/10.1002/nla.1680020208.
[35] B. Flemisch, I. Berre, W. Boon, A. Fumagalli, N. Schwenck, A. Scotti, I. Stefansson, A. Tatomir, Benchmarks for single-phase flow in fractured porous media, Adv. Water Resour. 111 (2018) 239–258, http://dx.doi.org/10.1016/j.advwatres.2017.10.036, https://www.sciencedirect.com/science/article/pii/S0309170817300143.
[36] C. Barus, Isothermals, isopiestics and isometrics relative to viscosity, Amer. J. Sci. 45 (1893) 87–96.
[37] S. Srinivasan, A generalized darcy–dupuit–forchheimer model with pressure-dependent drag coefficient for flow through porous media under large pressure gradients, Transp. Porous Media 111 (3) (2016) 741–750, http://dx.doi.org/10.1007/s11242-016-0625-y.

Paper D

Robust Linear Domain Decomposition Schemes for Reduced Non-linear Fracture Flow Models

E. Ahmed, A. Fumagalli, A. Budiša, E. Keilegavlen, J. M. Nordbotten, A. F. Radu

SIAM Journal on Numerical Analysis. In review.

arXiv:1906.05831 [math.NA]

Robust linear domain decomposition schemes for reduced non-linear fracture flow models

Elyes Ahmed[†] Alessio Fumagalli[*] Ana Budiša[†] Eirik Keilegavlen[†]
Jan M. Nordbotten[†‡] Florin A. Radu[†]

June 13, 2019

Abstract

In this work, we consider compressible single-phase flow problems in a porous media containing a fracture. In the latter, a non-linear pressure-velocity relation is prescribed. Using a non-overlapping domain decomposition procedure, we reformulate the global problem into a non-linear interface problem. We then introduce two new algorithms that are able to efficiently handle the non-linearity and the coupling between the fracture and the matrix, both based on linearization by the so-called L-scheme. The first algorithm, named MoLDD, uses the L-scheme to resolve the non-linearity, requiring at each iteration to solve the dimensional coupling via a domain decomposition approach. The second algorithm, called ItLDD, uses a sequential approach in which the dimensional coupling is part of the linearization iterations. For both algorithms, the computations are reduced only to the fracture by pre-computing, in an offline phase, a multiscale flux basis (the linear Robin-to-Neumann co-dimensional map), that represent the flux exchange between the fracture and the matrix. We present extensive theoretical findings and in particular, the stability and the convergence of both schemes are obtained, where user-given parameters are optimized to minimise the number of iterations. Examples on two important fracture models are computed with the library PorePy and agree with the developed theory.

Key words: Porous medium; reduced fracture models; generalized Forchheimer's laws; mortar mixed finite element; multiscale flux basis; non-linear interface problem; non-overlapping domain decomposition; L-scheme.

1 Introduction

Fractures are ubiquitous in porous media and strongly affect the flow and transport. Several energy and environmental applications including carbon sequestration, geothermal energy, and ground-water contamination involve flow and transport problems in a porous medium containing fractures. Typically, fractures are thin and long formations that correspond to a fast pathway along which medium properties, such as permeability or porosity, differ from the adjacent formations (the rocks) (see [5, 19, 38, 43]). Since it appears to be the cornerstone of many complex fracture models, we consider here (on the fractures) non-Darcy flow generalized Forchheimer's law [37].

1.1 Model problem

Let Ω be a bounded domain in $\mathbb{R}^d$, $d \in \{2, 3\}$, with boundary $\Gamma := \partial\Omega$. Furthermore, let T be the final time simulation and $I := (0, T)$. Suppose that $\gamma \subset \Omega$ is a $(d-1)$-dimensional surface that divides Ω into

[†]Department of Mathematics, University of Bergen, P. O. Box 7800, N-5020 Bergen, Norway. elyes.ahmed@uib.no, ana.budisa@uib.no, eirik.keilegavlen@uib.no, jan.nordbotten@uib.no, florin.radu@uib.no,

[*]Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, 10129 Torino, Italy. alessio.fumagalli@polito.it,

[‡]Department of Civil and Environmental Engineering, Princeton University, Princeton, N. J., USA.

two subdomains: $\Omega = \Omega_1 \cup \Omega_2 \cup \gamma$, where $\gamma := \partial\Omega_1 \cap \partial\Omega_2$ and $\Gamma_i := \partial\Omega_i \cap \partial\Omega$, $i \in \{1,2\}$. Assume that the flow in $I \times \Omega_i$, $i \in \{1,2\}$, is described by the system of equations

$$\mathbf{K}_i^{-1}\mathbf{u}_i + \nabla p_i = \mathbf{0} \quad \text{in } I \times \Omega_i, \tag{1.1a}$$
$$\partial_t p_i + \nabla \cdot \mathbf{u}_i = f_i \quad \text{in } I \times \Omega_i, \tag{1.1b}$$
$$p_i = 0 \quad \text{in } I \times \Gamma_i, \tag{1.1c}$$
$$p_i(\cdot, 0) = p_i^0 \quad \text{in } \Omega_i, \tag{1.1d}$$

and in $I \times \gamma$, by the following equations

$$\xi(\mathbf{u}_\gamma) + \mathbf{K}_\gamma^{-1}\mathbf{u}_\gamma + \nabla_\tau p_\gamma = \mathbf{0} \quad \text{in } I \times \gamma, \tag{1.2a}$$
$$\partial_t p_\gamma + \nabla_\tau \cdot \mathbf{u}_\gamma = f_\gamma + (\mathbf{u}_1 \cdot \mathbf{n}_1 + \mathbf{u}_2 \cdot \mathbf{n}_2) \quad \text{in } I \times \gamma, \tag{1.2b}$$
$$p_\gamma = 0 \quad \text{in } I \times \partial\gamma, \tag{1.2c}$$
$$p_\gamma(\cdot, 0) = p_\gamma^0 \quad \text{in } \gamma, \tag{1.2d}$$

where the transmission conditions

$$-\mathbf{u}_i \cdot \mathbf{n}_i + \alpha_\gamma p_i = \alpha_\gamma p_\gamma \quad \text{on } I \times \gamma, \tag{1.3}$$

for $i \in \{1,2\}$, are prescribed. Here, ∇_τ denotes the $(d-1)$-dimensional gradient operator in the plane of γ, $\mathbf{K}_\gamma$ is the hydraulic conductivity tensor in the fracture, $\mathbf{K}_i$ is the hydraulic conductivity tensor in the subdomain Ω_i and $\mathbf{n}_i$ is the outward unit normal vector to $\partial\Omega_i$, $i \in \{1,2\}$. The function ξ is a nonlinear function extending the classical Forchheimer flow to more general laws. In (1.3), the coefficient α_γ is proportional to the normal component of the permeability of the physical fracture and inversely proportional to the fracture width/aperture. The functions f_γ and f_i, $i \in \{1,2\}$, are source terms in the fracture and in the matrix, respectively. For simplicity we have imposed a homogeneous Dirichlet condition on the boundary $\partial\Omega$. Finally, p_γ^0 and p_i^0, $i \in \{1,2\}$, are initial conditions.

The system (1.1)–(1.3) is a mixed-dimensional model for flow in fractured porous media: the equations (1.1b)-(1.1a) are the mass conservation equation and the Darcy's law equation in the subdomain Ω_i while equations (1.2b)-(1.2a) are the lower-dimensional mass conservation and a non-Darcy flow generalized Forchheimer's law, in the fracture of co-dimension 1. Together, these equations form a non-standard transmission problem where the fracture system sees the surrounding matrix system through the source term $(\mathbf{u}_1 \cdot \mathbf{n}_1 + \mathbf{u}_2 \cdot \mathbf{n}_2)$ in (1.2b), while the rock matrix system communicates to the fracture through Robin type boundary conditions (1.3). Note that the restriction to only one fracture is made for the ease of presentation, but the model and the analysis below can be extended straightforwardly to more fractures, see for example [1, 43].

1.2 Assumptions on the data and weak formulation

Let $D \subseteq \Omega$. For $s \geq 0$, $||\cdot||_{s,D}$ stands for the usual Sobolev norm on $H^s(D)$. If $s = 0$, $||\cdot||_D$ is simply the L^2 norm and $(\cdot,\cdot)_D$ stands for the L^2 scalar product. We define the weak spaces in Ω_i for $i \in \{1,2\}$ as

$$\mathbf{V}_i := \left\{\mathbf{v} \in \mathbf{H}(\mathrm{div}, \Omega_i) : \mathbf{v} \cdot \mathbf{n}_i \in L^2(\gamma)\right\} \quad \text{and} \quad M_i := L^2(\Omega_i),$$

where we have implicitly considered the trace operator of $\mathbf{v} \cdot \mathbf{n}_i$. Moreover, we introduce their global versions by $\mathbf{V} := \bigoplus_{i=1}^2 \mathbf{V}_i$ and $M := \bigoplus_{i=1}^2 M_i$. The mixed spaces on the fracture γ, are $\mathbf{V}_\gamma := \mathbf{H}(\mathrm{div}_\tau, \gamma)$ and $M_\gamma := L^2(\gamma)$. For simplicity of notation, we introduce the jump $[\![\cdot]\!]$ given by $[\![\mathbf{u} \cdot \mathbf{n}]\!] := \mathbf{u}_1 \cdot \mathbf{n}_1 + \mathbf{u}_2 \cdot \mathbf{n}_2$. and the functions $\mathbf{K}$ and f in $\Omega_1 \cup \Omega_2$ such that $\mathbf{K}_i := \mathbf{K}|_{\Omega_i}$, and $f_i := f|_{\Omega_i}$, $i \in \{1,2\}$. Throughout the paper, we assume that the following assumptions hold true:

(A1) $\xi : \mathbb{R} \to \mathbb{R}$ is C^1, strictly increasing and Lipschitz continuous, i.e., there exist $\xi_m > 0$ and L_ξ such that $\xi_m \leq \xi'(\mathbf{u}) \leq L_\xi < \infty$. Otherwise, we require bounded flux for the differential problem (1.1)–(1.3), i.e, $\mathbf{u} \in [L^\infty(\Omega)]^d$, when ξ is simply an increasing function ($\xi' \geq 0$), and we let $L_\xi := \sup_{|\mathbf{u}| \leq C_\xi} \xi'(\mathbf{u})$, where $C_\xi := \sup_{\mathbf{x} \in \overline{\Omega}} |\mathbf{u}(\mathbf{x})|$.

(**A2**) $\mathbf{K} : \mathbb{R}^d \to \mathbb{R}^d$ is assumed to be constant in time and bounded; there exist $c_{\mathbf{K}} > 0$ and $C_{\mathbf{K}}$ such that $\zeta^{\mathrm{T}}\mathbf{K}^{-1}(\mathbf{x})\zeta \geq c_{\mathbf{K}}|\zeta|^2$ and $|\mathbf{K}^{-1}(\mathbf{x})\zeta| \leq C_{\mathbf{K}}|\zeta|$ for a.e. $\mathbf{x} \in \Omega_1 \cup \Omega_2$, $\forall \zeta \in \mathbb{R}^d$.

(**A3**) $\mathbf{K}_\gamma : \mathbb{R}^{d-1} \to \mathbb{R}^{d-1}$ is assumed to be constant in time and bounded; there exist $c_{\mathbf{K},\gamma} > 0$ and $C_{\mathbf{K},\gamma}$ such that $\zeta^{\mathrm{T}}\mathbf{K}_\gamma^{-1}(\mathbf{x})\zeta \geq c_{\mathbf{K},\gamma}|\zeta|^2$ and $|\mathbf{K}_\gamma^{-1}(\mathbf{x})\zeta| \leq C_{\mathbf{K},\gamma}|\zeta|$ for a.e. $\mathbf{x} \in \gamma$, $\forall \zeta \in \mathbb{R}^{d-1}$.

(**A4**) The Robin parameter α_γ is a strictly positive constant.

(**A5**) The initial conditions are such that $p_i^0 \in L^2(\Omega_i)$, $i \in \{1,2\}$, and $p_\gamma^0 \in L^2(\gamma)$. The source terms are such that $f_i \in L^2(0,T;L^2(\Omega_i))$, $i \in \{1,2\}$, and $f_\gamma \in L^2(0,T;L^2(\gamma))$. For simplicity we further assume that f and f_γ are piecewise constant in time with respect to the temporal mesh introduced in Section 2.1.

Remark 1.1 (On assumptions). *The Lipschitz-continuity of ξ is not true when the function ξ (therefore the flux) is unbounded, as it is the case for generalized Forchheimer's law. However, for bounded flux $\mathbf{u}$, this can be verified. Otherwise, this assumption can be recovered by truncating the original function ξ. Obviously, the solution of the truncated problem will not in general solve the original one. See, for example [44].*

We introduce the bilinear forms $a_i : \mathbf{V}_i \times \mathbf{V}_i \to \mathbb{R}$, $b_i : \mathbf{V}_i \times M_i \to \mathbb{R}$ and $c_i : M_i \times M_i \to \mathbb{R}$, $i \in \{1,2\}$,

$$a_i(\mathbf{u},\mathbf{v}) := (\mathbf{K}^{-1}\mathbf{u},\mathbf{v})_{\Omega_i} + \frac{1}{\alpha_\gamma}(\mathbf{u}\cdot\mathbf{n}_i, \mathbf{v}\cdot\mathbf{n}_i)_\gamma, \quad b_i(\mathbf{u},q) := (\nabla\cdot\mathbf{u},q)_{\Omega_i}, \quad c_i(p,q) := (p,q)_{\Omega_i}. \tag{1.4}$$

On the fracture, we define the bilinear forms $a_\gamma : \mathbf{V}_\gamma \times \mathbf{V}_\gamma \to \mathbb{R}$, $b_\gamma : \mathbf{V}_\gamma \times M_\gamma \to \mathbb{R}$ and $c_\gamma : M_\gamma \times M_\gamma \to \mathbb{R}$,

$$a_\gamma(\mathbf{u},\mathbf{v}) := (\mathbf{K}_\gamma^{-1}\mathbf{u},\mathbf{v})_\gamma, \quad b_\gamma(\mathbf{u},\mu) := (\nabla_\tau\cdot\mathbf{u},\mu)_\gamma, \quad c_\gamma(\lambda,\mu) := (\lambda,\mu)_\gamma. \tag{1.5}$$

With the above notations, a weak solution of (1.1)–(1.3) is given in the following.

Definition 1.2 (Mixed-dimensional weak solution). *Assume that* (**A1**)–(**A5**) *hold true. We say that* $(\mathbf{u},p) \in L^2(0,T;\mathbf{V})\times H^1(0,T;M)$ *and* $(\mathbf{u}_\gamma,p_\gamma) \in L^2(0,T;\mathbf{V}_\gamma)\times H^1(0,T;M_\gamma)$ *form a weak solution of* (1.1)–(1.3) *if it satisfies, for each* $i \in \{1,2\}$,

$$a_i(\mathbf{u},\mathbf{v}) - b_i(\mathbf{v},p) + (p_\gamma, \mathbf{v}\cdot\mathbf{n}_i)_\gamma = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_i, \tag{1.6a}$$

$$c_i(\partial_t p, q) + b_i(\mathbf{u},q) = (f,q)_{\Omega_i} \qquad \forall q \in M_i, \tag{1.6b}$$

$$(\xi(\mathbf{u}_\gamma),\mathbf{v})_\gamma + a_\gamma(\mathbf{u}_\gamma,\mathbf{v}) - b_\gamma(\mathbf{v},p_\gamma) = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_\gamma, \tag{1.6c}$$

$$c_\gamma(\partial_t p_\gamma,\mu) + b_\gamma(\mathbf{u}_\gamma,\mu) - (\llbracket\mathbf{u}\cdot\mathbf{n}\rrbracket,\mu)_\gamma = (f_\gamma,\mu)_\gamma \quad \forall \mu \in M_\gamma, \tag{1.6d}$$

together with the initial conditions (1.1d), *and* (1.2d) *in a weak sense.*

In this paper we assume that a weak solution by Definition 1.2 exists. For the static model, and when ξ stems from the classical the Forchheimer's law, the existence and uniqueness of a weak solution was shown in [38]. That of the linear case, i.e., $\xi := 0$, was studied in [35]. Through the paper, we will also consider the case of continuous pressure across γ by letting $\alpha_\gamma \to \infty$ in (1.3). For this case, we will use Definition 1.2 for the week formulation keeping in mind that in (1.4) a_i is simply $a_i(\mathbf{u},\mathbf{v}) := (\mathbf{K}^{-1}\mathbf{u},\mathbf{v})_{\Omega_i}$ and $\mathbf{V}_i := \mathbf{H}(\mathrm{div},\Omega_i)$ for $i \in \{1,2\}$.

1.3 Goal and positioning of the paper

The mixed-dimensional problem (1.1)–(1.3) is an alternative to the possibility to use fine grids of the spatial discretization in the (physical) fracture and thus reduces the computational cost. This idea was introduced in [5] for highly permeable fractures and in [6] for fractures that may be much more permeable than the surrounding medium or nearly impermeable [26, 29, 48]. Particularly, for "fast-path" fractures, Darcy's law is replaced by the classical Darcy-Forchheimer's law as established in [38]. We also refer to [3,16,39,40,54] for extensions to other flow models. Here, we extend the model in [1,38] to unsteady non-Darcy flow generalized Forchheimer's law. The work can be extended straightforwardly to viscosity models for generalized Newtonian fluids, including the Power law, the Cross model and the Carreau model [24,25].

Considerable research efforts have been conducted to mixed-dimensional fracture models. Several numerical schemes for steady models have been proposed, such as a cell-centered finite volume scheme in [34],

an extended finite element method in [21], a mimetic finite difference [8] and a block-centred finite difference method in [42]. We also mention several contributions on the application of mixed methods, on conforming and non-conforming grids [14,20,27,43]; see [13] for detailed account of major contribution on fracture models and discretization approaches. The aforementioned numerical approaches solve coupled fracture-matrix models monolithically, which leads to a large system, particularly if mixed finite element (MFE) methods are adopted [42]. This is especially the case when incorporating different equations varied in type, such as coupling linear and non-linear systems, and where often interface conditions involve additional variables. Domain decomposition (DD) is an elegant tool for modeling such a multi-physics problem and can provide an effective tool for reducing computational complexity and performing parallel calculations. See [22,47] for a general introduction of the subject. In [28], the authors combine domain decomposition techniques with mixed finite element methods for the reduced Darcy-Forchheimer fracture model (see [5,35] for the linear case).

In this paper, we propose efficient DD methods to solve (1.6) combining the mortar mixed finite element method (MMFEM) [10, 32, 55] with non-overlapping domain decomposition [1, 5, 45] and the L-scheme method [41,46]. Our method first reformulates (1.6) into an interface problem by eliminating the subdomain variables. The resulting problem posed only on the fracture is a *superposition* of a *non-linear local* flow operator within the fracture and a *linear non-local* one handling the flux contribution from the subdomains (Robin-to-Neumann type operator). After approximating this problem with the MMFEM in space and the backward Euler scheme in time, we obtain a non-linear system to solve at each time step. A first algorithm is then built with the L-scheme employed as a linearization procedure; a robust quasi-Newton method with a parameter $L > 0$ mimicking the Jacobian from the Newton method [41, 46]. At each iteration of the L-scheme, an *inner iterative algorithm*, such as GMRes or any Krylov solver, is used to solve the linear interface problem [30]. The action of the interface operator requires solving subdomain problems with Robin boundary condition on the fracture. This algorithm referred to henceforth as the Monolithic LDD-scheme (MoLDD) is *Jacobian-free* and subdomain solves are done in parallel. This LDD scheme will be shown to be *unconditionally stable*. Stability and condition number estimates of the inner DD system are obtained as well as contraction estimates and rates of convergence for the outer scheme. While MoLDD offers an elegant outer-inner approach to solve the interface-fracture problem, there is a computational overhead associated with its non-local part (DD), see *e.g.* [1, 30]. Precisely, the dominant computational cost in this approach is measured by the number of subdomain solves; increasing the non-linearity or DD strength and refining the grids both lead to an increase in the number of iterations and the number of subdomain solves.

More recently, the L-scheme has gained attention as an efficient solver to treat simultaneously non-linear and coupling effects in complex problems. See for example [49] for an application of the L-scheme on a non-linear DD problem and [15] on a non-linear coupling one. Building further on this idea, we propose a second algorithm, in which the DD step is part of the linearization iterations (see [4,17] for related works). In other words, the L-scheme is now synchronizing linearization and domain decomposition through one-loop algorithm. This approach referred as the Iterative LDD-scheme (ItLDD) differs from the one commonly used when dealing with non-linear interface problems in the context of DD [2, 12]. At each iteration it has the cost of the sequential approach, yet it converges to the fully monolithic approach. This approach reduces the computational costs as no inner DD solver is required and only a modest number of subdomain solves is required at each iteration, which still done in parallel. This algorithm increases *local to non-local cooperation* and saves time if one process is dominating the whole problem.

The second contribution of this paper concerns the robust and efficient implementation of the LDD schemes above. Precisely, the dominant computational costs in these schemes comes from the subdomain solves. To reduce this computational cost, we make use of the multiscale flux basis framework from [32]. The fact that the non-linearity in (1.6) is only within the local operator on the fracture, we can adopt the notion that the linear non-local contribution from the rock subdomains can be expressed as a superposition of multiscale basis functions [1, 31, 32]. These multiscale flux basis consists of the flux (or velocity trace) response from each fracture pressure degrees of freedom. They are computed by solving a *fixed* number of *steady* Robin subdomain problems, which is equal to the number of fracture pressure degrees of freedom per subdomain. An inexpensive linear combination of the multiscale flux basis functions then replaces the subdomain solves in any inner/outer iteration of the algorithms. This step of *freezing* the contributions on the flow from the rock matrices can be *easily coded, cheaply evaluated*, and *efficiently used* in *all the algorithms*. That is, it permits reusing the same basis functions to compare MoLDD with ItLDD as well as to simulate various linear and non-linear models for flow in the fracture by varying ξ and finally exploring

high and low permeable fractures. This is in total *conformity* with the spirit of *reduced basis* [11, 52, 53]. Crucially, if a fixed time step is used, our multiscale flux basis applied to a non-linear time-dependent problem are constructed only once in the offline phase. This should be kept in mind also for our numerical results reported in the last section. Numerical results are computed with the library PorePy [36].

1.4 Outline of the paper

This paper is organized as follows: Firstly, the approximation of problem (1.6) using the MMFEM in space and a backward Euler scheme in time is given in Section 2. Also, the reduction of this mixed-dimensional scheme into a non-linear interface one is introduced. The LDD-schemes are formulated in Section 3. In Section 4 and Section 5, the analysis of the schemes is presented. Section 6 describes the implementation based on the multiscale flux basis framework. Finally, we showcase the performance of our methods on several numerical examples in Section 7 and draw the conclusions in Section 8.

2 The DD formulation

As explained earlier, it is natural to solve the mixed-dimensional problem (1.6) using domain decomposition techniques, especially as these methods make it possible to take different time grids in the subdomains and in the fracture.

2.1 Discretisation in space and time

We introduce in this section the partitions of Ω and $(0, T)$, basic notation, and the mortar mixed finite element discretization of the mixed-dimensional problem (1.6) .

Let $\mathcal{T}_{h,i}$ be a partition of the subdomain Ω_i into either d-dimensional simplicial or rectangular elements. Moreover, we assume that these meshes are such that $\mathcal{T}_h = \cup_{i=1}^2 \mathcal{T}_{h,i}$ forms a conforming finite element mesh on Ω. We also let $\mathcal{T}_{h,\gamma}$ be either a partition of the fracture γ induced by $\mathcal{T}_h$ or slightly coarser. Denote h as the maximal mesh size of both $\mathcal{T}_h$ and $\mathcal{T}_{h,\gamma}$.

For an integer $N \geq 0$, let $(\tau^n)_{0 \leq n \leq N}$ denotes a sequence of positive real numbers corresponding to the discrete time steps such that $T = \sum_{n=1}^{N} \tau^n$. Let $t^0 := 0$, and $t^n := \sum_{j=1}^{n} \tau^j,\ 1 \leq n \leq N$, be the discrete times. Let $I^n := (t^{n-1}, t^n],\ 1 \leq n \leq N$.

2.1.1 Finite-dimensional spaces and projection operators

For the approximation of scalar unknowns, we introduce the approximation spaces $M_h := M_{h,1} \times M_{h,2}$ and $M_{h,\gamma}$, where $M_{h,i}$, $i \in \{1, 2\}$, and $M_{h,\gamma}$ are the spaces of piecewise constant functions associated with $\mathcal{T}_{h,i}$, $i \in \{1, 2\}$ and $\mathcal{T}_{h,\gamma}$, respectively. For the vector unknowns, we introduce the approximation spaces $\mathbf{V}_h := \mathbf{V}_{h,1} \times \mathbf{V}_{h,2}$ and $\mathbf{V}_{h,\gamma}$, where $\mathbf{V}_{h,i}$, $i \in \{1, 2\}$ and $\mathbf{V}_{h,\gamma}$, are the lowest-order Raviart-Thomas-Nédélec finite elements spaces associated with $\mathcal{T}_{h,i}$, $i \in \{1, 2\}$ and $\mathcal{T}_{h,\gamma}$, respectively. Thus, $\mathbf{V}_h \times M_h \subset \mathbf{V} \times M$ and $\mathbf{V}_{h,\gamma} \times M_{h,\gamma} \subset \mathbf{V}_\gamma \times M_\gamma$. For all of the above spaces,

$$\nabla \cdot \mathbf{V}_h = M_h, \quad \text{and} \quad \nabla_\tau \cdot \mathbf{V}_{h,\gamma} = M_{h,\gamma}, \tag{2.1}$$

and there exists a projection $\tilde{\Pi}_i : \mathbf{H}^{1/2+\epsilon}(\Omega_i) \cap \mathbf{V}_i \to \mathbf{V}_{h,i}$, $i \in \{1, 2\}$, (for any $\epsilon > 0$) see *e.g.* [31], satisfying among other properties that for any $\mathbf{u} \in \mathbf{H}^{1/2+\epsilon}(\Omega_i) \cap \mathbf{V}_i$

$$(\nabla \cdot (\mathbf{u} - \tilde{\Pi}_i \mathbf{u}), q)_{\Omega_i} = 0 \qquad \forall q \in M_{h,i}, \tag{2.2}$$

$$((\mathbf{u} - \tilde{\Pi}_i \mathbf{u}) \cdot \mathbf{n}_i, \mathbf{v} \cdot \mathbf{n}_i)_{\partial\Omega_i} = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}. \tag{2.3}$$

We also note that if $\mathbf{u} \in \mathbf{H}^\epsilon(\Omega_i) \cap \mathbf{V}_i$, $0 < \epsilon < 1$, $\tilde{\Pi}_i \mathbf{u}$ is well-defined [51] and

$$||\tilde{\Pi}_i \mathbf{u}||_{\Omega_i} \lesssim ||\mathbf{u}||_{\epsilon,\Omega_i} + ||\nabla \cdot \mathbf{u}||_{\Omega_i}. \tag{2.4}$$

We introduce $\mathcal{Q}_{h,i}$ the L^2-projection onto $\mathbf{V}_{h,i} \cdot \mathbf{n}_i$ and denote $\mathcal{Q}_{h,i}^{\mathrm{T}} : \mathbf{V}_{h,i} \cdot \mathbf{n}_i \to M_{h,\gamma}$ as the L^2-projection from the normal velocity trace on the subdomains onto the mortar space $M_{h,\gamma}$. Thus, for all $\lambda \in M_{h,\gamma}$ the condition

$$||\lambda||_\gamma \lesssim ||\mathcal{Q}_{h,1}\lambda||_\gamma + ||\mathcal{Q}_{h,2}\lambda||_\gamma, \tag{2.5}$$

can easily be verified if the mesh on the fracture $\mathcal{T}_{h,\gamma}$ matches the one resulting from the surrounding subdomains, or if $\mathcal{T}_{h,\gamma}$ is chosen slightly coarser [9, 14]. Note that (2.1) can be satisfied by choosing any of the usual MFE pairs. That of the condition (2.5) can be satisfied even if the space $M_{h,\gamma}$ is (not much) richer than the space of normal traces on γ of elements of $\mathbf{V}_h$ [31, 32].

2.1.2 The discrete scheme

The fully discrete scheme of the mixed-dimensional formulation (1.6) based on the MMFEM in space and the backward Euler scheme in time is defined through the following.

Definition 2.1 (The mixed-dimensional scheme). *At each time step $n \geq 1$, assuming $(p_{h,\gamma}^{n-1}, p_h^{n-1})$ is given, we look for $(\mathbf{u}_h^n, p_h^n) \in \mathbf{V}_h \times M_h$ and $(\mathbf{u}_{h,\gamma}^n, p_{h,\gamma}^n) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$ such that, for $i \in \{1, 2\}$,*

$$a_i(\mathbf{u}_h^n, \mathbf{v}) - b_i(\mathbf{v}, p_h^n) = -(p_{h,\gamma}^n, \mathbf{v} \cdot \mathbf{n}_i)_\gamma \qquad \forall \mathbf{v} \in \mathbf{V}_h. \tag{2.6a}$$

$$c_i(p_h^n - p_h^{n-1}, q) + \tau^n b_i(\mathbf{u}_h^n, q) = \tau^n (f^n, \mu)_{\Omega_i} \qquad \forall q \in M_h, \tag{2.6b}$$

$$(\xi(\mathbf{u}_{h,\gamma}^n), \mathbf{v}) + a_\gamma(\mathbf{u}_{h,\gamma}^n, \mathbf{v}) - b_\gamma(\mathbf{v}, p_{h,\gamma}^n) = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_{h,\gamma}, \tag{2.6c}$$

$$c_\gamma(p_{h,\gamma}^n - p_{h,\gamma}^{n-1}, \mu) + \tau^n b_\gamma(\mathbf{u}_{h,\gamma}^n, \mu) - \tau^n (\llbracket \mathbf{u}_h^n \cdot \mathbf{n} \rrbracket, \mu)_\gamma = \tau^n (f_\gamma^n, \mu)_\gamma \quad \forall \mu \in M_{h,\gamma}. \tag{2.6d}$$

2.2 Reduction into an interface problem

Following the algorithm in [1], we reduce the mixed-dimensional scheme in Definition 2.1 into a non-linear interface one posed on γ, which can be solved using an appropriate combination of a linearization method and an iterative Krylov solver. For $i \in \{1, 2\}$, we let

$$p_{h,i}^n = p_{h,i}^*(\lambda_{h,\gamma}^n) + \bar{p}_{h,i}^n \quad \text{and} \quad \mathbf{u}_{h,i}^n = \mathbf{u}_{h,i}^*(\lambda_{h,\gamma}^n) + \bar{\mathbf{u}}_{h,i}^n, \quad 1 \leq n \leq N, \tag{2.7}$$

where for $\lambda_{h,\gamma}^n \in M_{h,\gamma}$, $(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}^n), p_{h,i}^*(\lambda_{h,\gamma}^n)) \in \mathbf{V}_{h,i} \times M_{h,i}$ solves

$$a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}^n), \mathbf{v}) - b_i(\mathbf{v}, p_{h,i}^*(\lambda_{h,\gamma}^n)) = -(\lambda_{h,\gamma}^n, \mathbf{v} \cdot \mathbf{n}_i)_\gamma \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}, \tag{2.8a}$$

$$c_i(p_{h,i}^*(\lambda_{h,\gamma}^n), q) + \tau^n b_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}^n), q) = 0 \qquad \forall q \in M_{h,i}, \tag{2.8b}$$

and $(\bar{\mathbf{u}}_{h,i}^n, \bar{p}_{h,i}^n) \in \mathbf{V}_{h,i} \times M_{h,i}$ solves

$$a_i(\bar{\mathbf{u}}_i^n, \mathbf{v}) - b_i(\mathbf{v}, \bar{p}_{h,i}^n) = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_{h,i}, \tag{2.9a}$$

$$c_i(\bar{p}_{h,i}^n - p_{h,i}^{n-1}, q) + \tau^n b_i(\bar{\mathbf{u}}_{h,i}^n, q) = \tau^n (f^n, \mu)_{\Omega_i} \qquad \forall q \in M_{h,i}, \tag{2.9b}$$

$$(\bar{p}_{h,i}^0, \mu)_{\Omega_i} = (p_{h,i}^0, \mu)_{\Omega_i} \qquad \forall \mu \in M_{h,i}. \tag{2.9c}$$

Define the forms $s_{\gamma,i} : M_{h,\gamma} \times M_{h,\gamma} \to \mathbb{R}$, $i \in \{1, 2\}$, $s_\gamma : M_{h,\gamma} \times M_{h,\gamma} \to \mathbb{R}$, and $g_\gamma^n : M_{h,\gamma} \to \mathbb{R}$,

$$s_{\gamma,i}(\lambda_{h,\gamma}^n, \mu) := (\mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\lambda_{h,\gamma}^n), \mu)_\gamma := -(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}^n) \cdot \mathbf{n}_i, \mu)_\gamma, \tag{2.10a}$$

$$s_\gamma(\lambda_{h,\gamma}^n, \mu) := (\mathcal{S}_\gamma^{\mathrm{RtN}}(\lambda_{h,\gamma}^n), \mu)_\gamma := \sum_{i=1}^{2} s_{\gamma,i}(\lambda_{h,\gamma}^n, \mu), \tag{2.10b}$$

$$g_\gamma^n(\mu) := (g_\gamma^n, \mu)_\gamma := \sum_{i=1}^{2} (\bar{\mathbf{u}}_{h,i}^n \cdot \mathbf{n}_i, \mu)_\gamma, \tag{2.10c}$$

where $\mathcal{S}_{\gamma,i}^{\mathrm{RtN}} : M_{h,\gamma} \to M_{h,\gamma}$, $1 \leq i \leq 2$, and $\mathcal{S}_\gamma^{\mathrm{RtN}} := \sum_{i=1}^{2} \mathcal{S}_{\gamma,i}^{\mathrm{RtN}}$ are Robin-to-Neumann type operators. Obviously, the operator $\mathcal{S}_{\gamma,i}^{\mathrm{RtN}}$ is linear. It is easy to verify that the non-linear mixed-dimensional scheme (2.6) is equivalent to the non-linear interface scheme.

Definition 2.2 (The reduced scheme). *Given $n \geq 1$ and $\lambda_{h,\gamma}^{n-1}$, find $(\mathbf{u}_{h,\gamma}^n, \lambda_{h,\gamma}^n) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$ such that*

$$(\xi(\mathbf{u}_{h,\gamma}^n), \mathbf{v})_\gamma + a_\gamma(\mathbf{u}_{h,\gamma}^n, \mathbf{v}) - b_\gamma(\mathbf{v}, \lambda_{h,\gamma}^n) = 0 \qquad \forall \mathbf{v} \in \mathbf{V}_{h,\gamma}, \tag{2.11a}$$

$$c_\gamma(\lambda_{h,\gamma}^n - \lambda_{h,\gamma}^{n-1}, \mu) + \tau^n b_\gamma(\mathbf{u}_{h,\gamma}^n, \mu) + \tau^n s_\gamma(\lambda_{h,\gamma}^n, \mu) = \tau^n (f_\gamma^n + g_\gamma^n, \mu)_\gamma \quad \forall \mu \in M_{h,\gamma}. \tag{2.11b}$$

In the next section, we propose two iterative approaches based on the L-scheme to solve (2.11). The first approach entails an inner-outer procedure of the form *linearize → solve the DD → update*, so that the L-scheme is used for the outer loop and the GMRes or any Krylov solver for the inner loop. The second approach is a one-loop procedure in which the L-scheme acts iteratively and simultaneously on the linearization and DD.

3 Robust L-type Domain-Decomposition (LDD) schemes

For the presentation of the algorithms, we shall denote the time step simply by τ, keeping in mind it may depend on n.

3.1 A monolithic LDD scheme

The monolithic LDD scheme (MoLDD) used to solve the interface problem (2.11) reads:

Algorithm 3.1 (The MoLDD scheme).

1. *Give the initial data $(\lambda_{h,\gamma}^0, p_h^0) \in M_{h,\gamma} \times M_h$, the stabilization parameter $L_\gamma > 0$ and the tolerance $\epsilon > 0$.*
2. **Do**
   (a) *Increase $n := n + 1$.*
   (b) *Choose an initial approximation $\mathbf{u}_{h,\gamma}^{n,-1} \in \mathbf{V}_{h,\gamma}$ of $\mathbf{u}_{h,\gamma}^n$. Set $k := -1$.*
   (c) **Do**
      i. *Increase $k := k + 1$.*
      ii. *Compute $(\mathbf{u}_{h,\gamma}^{n,k}, \lambda_{h,\gamma}^{n,k}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$ such that, for all $(\mathbf{v}, \mu) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,*

$$(\xi(\mathbf{u}_{h,\gamma}^{n,k-1}) + L_\gamma(\mathbf{u}_{h,\gamma}^{n,k} - \mathbf{u}_{h,\gamma}^{n,k-1}), \mathbf{v})_\gamma + a_\gamma(\mathbf{u}_{h,\gamma}^{n,k}, \mathbf{v}) - b_\gamma(\mathbf{v}, \lambda_{h,\gamma}^{n,k}) = 0, \tag{3.1a}$$

$$c_\gamma(\lambda_{h,\gamma}^{n,k} - \lambda_{h,\gamma}^{n-1}, \mu) + \tau b_\gamma(\mathbf{u}_{h,\gamma}^{n,k}, \mu) + \tau s_\gamma(\lambda_{h,\gamma}^{n,k}, \mu) = \tau(f_\gamma^n + g_\gamma^n, \mu)_\gamma. \tag{3.1b}$$

   while $$\frac{\|(\mathbf{u}_{h,\gamma}^{n,k}, \lambda_{h,\gamma}^{n,k}) - (\mathbf{u}_{h,\gamma}^{n,k-1}, \lambda_{h,\gamma}^{n,k-1})\|_\gamma}{\|(\mathbf{u}_{h,\gamma}^{n,k-1}, \lambda_{h,\gamma}^{n,k-1})\|_\gamma} \geq \epsilon.$$

   (d) *Update the subdomain solutions via* (2.7).

   while $n \leq N$.

Remark 3.2 (Advantages of MoLDD-scheme). *The advantages of Algorithm 3.1 are multiple: (i) the algorithm is Jacobian-free and independent of the initialization, (ii) subdomain solves can be done in parallel, (iii) we can reuse of existing d- and $(d-1)$-dimensional codes for solving linear Darcy problem, and (iv) optimal convergence rate is obtained with a stabilization amount determined efficiently through L_γ.*

The MoLDD scheme involves the solution of a linear Darcy interface problem (3.1) at each iteration $k \geq 0$. To see that, we introduce the linear operators $\mathbf{A}_{L,\gamma} : \mathbf{V}_{h,\gamma} \to \mathbf{V}_{h,\gamma}$ and $\mathbf{B}_\gamma : \mathbf{V}_{h,\gamma} \to M_{h,\gamma}$, defined as $(\mathbf{A}_{L,\gamma}\mathbf{u}, \mathbf{v})_\gamma := a_\gamma(\mathbf{u}, \mathbf{v}) + L_\gamma(\mathbf{u}, \mathbf{v})_\gamma$, $\forall \mathbf{u}, \mathbf{v} \in \mathbf{V}_{h,\gamma}$, and $(\mathbf{B}_\gamma \mathbf{u}, q) := b_\gamma(\mathbf{u}, q)$, $\forall \mathbf{v} \in \mathbf{V}_{h,\gamma}$, $\forall q \in M_{h,\gamma}$. Now (3.1) becomes

$$\mathcal{A}_{\mathrm{DD}} \begin{bmatrix} \mathbf{u}_{h,\gamma}^{n,k} \\ \lambda_{h,\gamma}^{n,k} \end{bmatrix} := \begin{bmatrix} \mathbf{A}_{L,\gamma} & \mathbf{B}_\gamma^{\mathrm{T}} \\ \mathbf{B}_\gamma & \mathcal{S}_\gamma^{\mathrm{RtN}} + \mathbf{I}/\tau \end{bmatrix} \begin{bmatrix} \mathbf{u}_{h,\gamma}^{n,k} \\ \lambda_{h,\gamma}^{n,k} \end{bmatrix} = \begin{bmatrix} L_\gamma \mathbf{u}_{h,\gamma}^{n,k-1} - \xi(\mathbf{u}_{h,\gamma}^{n,k-1}) \\ g_\gamma^n + f_\gamma^n + \lambda_{h,\gamma}^{n-1}/\tau \end{bmatrix} := \mathcal{F}_\gamma, \tag{3.2}$$

which can solved using a Krylov type method, such as GMRes or MINRes. Given an initial guess $\mathbf{w}_{h,\gamma}^{(0)} = [\mathbf{u}_{h,\gamma}^{n,k,0}, \lambda_{h,\gamma}^{n,k,0}]^{\mathrm{T}}$, the GMRes algorithm generates a sequence of iterates $\{\mathbf{w}_{h,\gamma}^{(m)}\}_{m\geq 1}$, where $\mathbf{w}_{h,\gamma}^{(m)}$ is a solution of the finite-dimensional minimization problem

$$\mathbf{w}_{h,\gamma}^{(m)} := \underset{\Psi \in \mathbf{w}_{h,\gamma}^{(0)} + \mathcal{K}_m}{\arg\min} ||\mathcal{F}_\gamma - \mathcal{A}_{\mathrm{DD}}\Psi||_\gamma, \tag{3.3}$$

as an approximate solution to (3.2), where $\mathcal{K}_m$ is the m-th Krylov subspace generated by the initial residual $\mathbf{r}_\gamma^{(0)} := \mathcal{F}_\gamma - \mathcal{A}_{\mathrm{DD}}\mathbf{w}_{h,\gamma}^{(0)}$, i.e.,

$$\mathcal{K}_m := \mathcal{K}_m(\mathcal{A}_{\mathrm{DD}}, \mathbf{r}_\gamma^{(0)}) := \mathrm{span}(\mathbf{r}_\gamma^{(0)}, \mathcal{A}_{\mathrm{DD}}\mathbf{r}_\gamma^{(0)}, \cdots, \mathcal{A}_{\mathrm{DD}}^{m-1}\mathbf{r}_\gamma^{(0)}).$$

At each GMRes iteration $m \geq 1$, we need to evaluate the action of the Robin-to-Neumann type operator $\mathcal{S}_\gamma^{\mathrm{RtN}}$ via (2.10), representing physically the contributions on the flow from the subdomains by solving Robin subdomain problems (2.8). Therefore, the GMRes algorithm is implemented in the matrix-free context [1, 30, 32]. We summarize the evaluation of the interface operator by the following steps:

Algorithm 3.3 (Evaluating the action of $\mathcal{S}_\gamma^{\mathrm{RtN}}$).

1. *Enter interface data* $\lambda_{h,\gamma}$.
2. **For** $i = 1 : 2$
   (a) *Project mortar pressure onto subdomain boundary, i.e.,* $\varphi_{h,\gamma,i} = \mathcal{Q}_{h,i}(\lambda_{h,\gamma})$.
   (b) *Solve the subdomain problem* (2.8) *with Robin data* $\varphi_{h,\gamma,i}$.
   (c) *Project the resulting flux onto the space* $M_{h,\gamma}$, *i.e.,* $\mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\lambda_{h,\gamma}) = -\mathcal{Q}_{h,i}^{\mathrm{T}}\mathbf{u}_{h,i}^*(\varphi_{h,\gamma,i}) \cdot \mathbf{n}_i$.

   EndFor
3. *Compute the flow contribution from the subdomains to the fracture given by the flux jump across the fracture,*

$$\mathcal{S}_\gamma^{\mathrm{RtN}}(\lambda_{h,\gamma}) = \sum_{i\in\{1,2\}} \mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\lambda_{h,\gamma}).$$

Remark 3.4 (Computational cost). *The evaluation of* $\mathcal{S}_\gamma^{\mathrm{RtN}}$ *dominates the total computational costs in Algorithm 3.1 (step 2(b) of Algorithm 3.3). The number of subdomain solves required by this method at each time step* $n \geq 1$ *is approximately equal to* $\sum_{k=1}^{N_{\mathrm{Lin}}^n} N_{\mathrm{DD}}^k$, *where* N_{Lin} *is the number of iterations of the L-scheme, and* N_{DD}^k *denotes the number of inner DD iterations. To set-up the right-hand side term* f_γ^n, *we need to solve once in the subdomains at each time step* $n \geq 1$.

3.2 A robust iterative LDD-scheme

An alternative LDD-scheme to solve the interface problem (2.11) is to let the L-scheme act iteratively not only on the non-linearity as in Algorithm 3.1, but also on the fracture-matrix coupling. Additional stabilization term is then required for the inter-dimensional coupling. This iterative scheme reads:

Algorithm 3.5 (The ItLDD scheme).

1. *Give* $(\lambda_{h,\gamma}^0, p_h^0) \in M_{h,\gamma} \times M_h$, *the stabilization parameters* $(L_{\gamma,p}, L_{\gamma,u}) > 0$, *and the tolerance* $\epsilon > 0$.
2. **Do**
   (a) *Increase* $n := n + 1$.
   (b) *Choose an initial approximation* $(\mathbf{u}_{h,\gamma}^{n,-1}, \lambda_{h,\gamma}^{n,-1}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$ *of* $(\mathbf{u}_{h,\gamma}^n, \lambda_{h,\gamma}^n)$. *Set* $k := -1$.
   (c) **Do**
      i. *Increase* $k := k + 1$.
      ii. *Compute* $(\mathbf{u}_{h,\gamma}^{n,k}, \lambda_{h,\gamma}^{n,k}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$ *such that, for all* $(\mathbf{v}, \mu) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,

$$(\xi(\mathbf{u}_{h,\gamma}^{n,k-1}) + L_{\gamma,u}(\mathbf{u}_{h,\gamma}^{n,k} - \mathbf{u}_{h,\gamma}^{n,k-1}), \mathbf{v})_\gamma + a_\gamma(\mathbf{u}_{h,\gamma}^{n,k}, \mathbf{v}) - b_\gamma(\mathbf{v}, \lambda_{h,\gamma}^{n,k}) = 0. \tag{3.4a}$$

$$c_\gamma(\lambda_{h,\gamma}^{n,k} - \lambda_{h,\gamma}^{n-1}, \mu) + \tau L_{\gamma,p}(\lambda_{h,\gamma}^{n,k} - \lambda_{h,\gamma}^{n,k-1}, \mu)_\gamma + \tau s_\gamma(\lambda_{h,\gamma}^{n,k-1}, \mu) + \tau b_\gamma(\mathbf{u}_{h,\gamma}^{n,k}, \mu) = \tau(f_\gamma^n + g_\gamma^n, \mu)_\gamma, \tag{3.4b}$$

While $\dfrac{\|(\mathbf{u}_{h,\gamma}^{n,k},\lambda_{h,\gamma}^{n,k})-(\mathbf{u}_{h,\gamma}^{n,k-1},\lambda_{h,\gamma}^{n,k-1})\|_\gamma}{\|(\mathbf{u}_{h,\gamma}^{n,k-1},\lambda_{h,\gamma}^{n,k-1})\|_\gamma}\geq\epsilon.$

(d) *Update the subdomain solutions via* (2.7).

while $n\leq N$.

Remark 3.6 (Computational cost). *The linear problem* (3.4) *is solved with the GMRes iterations* (3.3)*. It requires at each iteration $k\geq 1$ only one solve per subdomain to evaluate the action of $\mathcal{S}_\gamma^{\mathrm{RtN}}$ via Algorithm 3.3 at the previous iteration, and this at each time step $n\geq 1$.*

Remark 3.7 (Advantages of ItLDD-scheme). *The advantages of the iterative approach described in Algorithm 3.5 are: (i) at each iteration $k\geq 1$, the systems in the fracture and the rock matrices cooperate sequentially in one loop and with negligible inter-processor communication, (ii) optimal convergence rate is obtained with precise stabilization parameters $(L_{\gamma,p},L_{\gamma,u})$, (iii) subdomain solves can be done in parallel, and (iv) existing codes for d- and $(d-1)$-dimensional Darcy problems can be cheaply combined and used for practical simulations.*

4 Analysis of MoLDD-scheme

The complete analysis of Algorithm 3.1 will be carried out in two steps: (i) we first study the stability of the iterate DD scheme (inner solver) and estimate the condition number, and (ii) we prove the convergence of the LDD scheme (outer solver), show the well-posedness of the discrete scheme, estimate the convergence rate and subsequently determine the optimal stabilization parameter. Throughout the paper, we will frequently use the standard identity

$$(a-b)\cdot a=\frac{1}{2}\left(a^2-b^2+(a-b)^2\right),\quad a,b\in\mathbb{R},\tag{4.1}$$

and inequality

$$|ab|\leq\frac{1}{2\delta}a^2+\frac{2}{\delta}b^2,\quad a,b,\delta\in\mathbb{R},\ \delta>0.\tag{4.2}$$

A key point in the analysis of the methods below are inverse inequalities.

Lemma 4.1 (Inverse inequalities). *There exist positive constants $C_{\mathrm{dTr}},C_{\mathrm{inv}}>0$ depending only on the shape regularity of the mesh such that*

$$||\mathbf{u}_h\cdot\mathbf{n}||_{\partial\Omega_i}\leq C_{\mathrm{dTr}}h^{-1/2}||\mathbf{u}_h||_{\Omega_i}\qquad\forall\mathbf{u}_h\in\mathbf{V}_{h,i},\tag{4.3}$$

$$||\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma\leq C_{\mathrm{inv}}h^{-1}||\mathbf{u}_{h,\gamma}||_\gamma\qquad\forall\mathbf{u}_{h,\gamma}\in\mathbf{V}_{h,\gamma}.\tag{4.4}$$

4.1 Analysis of the DD step

To simplify the analysis, we rewrite problem (3.1) as: find $(\mathbf{u}_{h,\gamma}^{n,k},\lambda_{h,\gamma}^{n,k})\in\mathbf{V}_{h,\gamma}\times M_{h,\gamma}$ such that,

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}^{n,k},\lambda_{h,\gamma}^{n,k}),(\mathbf{v},\mu))+s_\gamma(\lambda_{h,\gamma}^{n,k},\mu)=\mathcal{F}_\gamma^{n,k-1}(\mathbf{v},\mu)\qquad\forall(\mathbf{v},\mu)\in\mathbf{V}_{h,\gamma}\times M_{h,\gamma},\tag{4.5}$$

where

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma},\lambda_{h,\gamma}),(\mathbf{v},\mu)):=a_\gamma(\mathbf{u}_{h,\gamma},\mathbf{v})+L_\gamma(\mathbf{u}_{h,\gamma},\mathbf{v})_\gamma+\frac{1}{\tau}(\lambda_{h,\gamma},\mu)_\gamma+b_\gamma(\mathbf{u}_{h,\gamma},\mu)-b_\gamma(\mathbf{v},\lambda_{h,\gamma}),\tag{4.6a}$$

$$\mathcal{F}_\gamma^{n,k-1}(\mathbf{v},\mu):=(\xi(\mathbf{u}_{h,\gamma}^{n,k-1})+L_\gamma\mathbf{u}_{h,\gamma}^{n,k-1},\mathbf{v})_\gamma+(f_\gamma^n+g_\gamma^n,\mu)_\gamma.\tag{4.6b}$$

Therein, $\mathcal{A}_\gamma$ is the linearized flow system on the fracture and s_γ is the flow contribution from the rock matrices. The first result concerns the properties of the coupling term s_γ.

Lemma 4.2 (Properties of the DD operator). *The interface bilinear form s_γ satisfies:*

- *s_γ is symmetric positive and semi-definite on $L^2(\gamma)$.*

- *There exists a constant $C_1 > 0$ independent of h such that, for all $\lambda_{h,\gamma} \in M_{h,\gamma}$,*

$$\left(C_1 \frac{C_{\mathbf{K}}}{\sqrt{c_{\mathbf{K}}}} + \frac{1}{\sqrt{\alpha_\gamma}}\right)^{-2} ||\lambda_{h,\gamma}||_\gamma^2 \leq s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) \leq \alpha_\gamma ||\lambda_{h,\gamma}||_\gamma^2. \tag{4.7}$$

Proof. Recalling (2.10), we take $\mathbf{v} = \mathbf{u}_{h,i}^*(\mu)$ and $q = p_{h,i}^*(\mu)$ in (2.8) to see that the bilinear form s_γ can be expressed as

$$s_\gamma(\lambda_{h,\gamma}, \mu) = \sum_{i=1}^{2} \{a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}), \mathbf{u}_{h,i}^*(\mu)) + c_i(p_{h,i}^*(\lambda_{h,\gamma}), p_{h,i}^*(\mu))\}. \tag{4.8}$$

It is now easy to see that the bilinear form s_γ is symmetric and positive semi-definite on $L^2(\gamma)$. We now show that if $s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) = 0$, then $\lambda_{h,\gamma} = 0$ on $M_{h,\gamma}$. Note that $s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) = 0$ implies that $\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) = p_{h,i}^*(\lambda_{h,\gamma}) = 0$. Again, (2.8) implies $(\mathcal{Q}_{h,i}\lambda_{h,\gamma}, \mathbf{v} \cdot \mathbf{n}_i)_\gamma = (\lambda_{h,\gamma}, \mathbf{v} \cdot \mathbf{n}_i)_\gamma = 0$ for any $\mathbf{v} \in \mathbf{V}_{h,i}$. Thus, we can find some $\mathbf{v}$ so that $\mathbf{v} \cdot \mathbf{n}_i = \mathcal{Q}_{h,i}\lambda_{h,\gamma}$ and then $||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma = 0$. Finally, (2.5) shows that $\lambda_{h,\gamma} = 0$ on γ.
We now infer the upper bound on s_γ. The assumption (**A2**) directly implies

$$c_{\mathbf{K}} ||\mathbf{u}_{h,i}||_{\Omega_i}^2 + \frac{1}{\alpha_\gamma} ||\mathbf{u}_{h,i} \cdot \mathbf{n}_i||_\gamma^2 \leq a_i(\mathbf{u}_{h,i}, \mathbf{u}_{h,i}), \quad \forall \mathbf{u}_{h,i} \in \mathbf{V}_{h,i}. \tag{4.9}$$

The definition (2.10) of s_γ gives

$$\begin{aligned} s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) &:= -\sum_{i=1}^{2} (\lambda_{h,\gamma}, \mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i)_\gamma \leq \sum_{i=1}^{2} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i||_\gamma ||\lambda_{h,\gamma}||_\gamma \\ &\leq \sum_{i=1}^{2} \alpha_\gamma^{1/2} a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}), \mathbf{u}_{h,i}^*(\lambda_{h,\gamma}))^{1/2} ||\lambda_{h,\gamma}||_\gamma. \end{aligned} \tag{4.10}$$

This result together with (4.8) lead to the upper bound in (4.7). We prove the lower bound by induction. To this aim, we let $(\psi_i, \mathbf{r}_i)$, $i \in \{1, 2\}$, be the solution of the auxiliary subdomain problem

$$\mathbf{r}_i + \mathbf{K}_i \nabla \psi_i = \mathbf{0}, \quad \text{in } \Omega_i, \tag{4.11a}$$
$$\nabla \cdot \mathbf{r}_i = 0, \quad \text{in } \Omega_i, \tag{4.11b}$$
$$\psi_i = 0, \quad \text{on } \Gamma_i, \tag{4.11c}$$
$$\mathbf{r}_i \cdot \mathbf{n}_i = \mathcal{Q}_{h,i}\lambda_{h,\gamma}, \quad \text{on } \gamma. \tag{4.11d}$$

For fracture network with immersed fractures or for subdomains with $\Gamma_i = \emptyset$, $\lambda_{h,\gamma}$ approximates the pressure on γ, which is determined up to a constant. This constant is fixed by a zero mean value constraint for $M_{h,\gamma}$ [9,27]. Thus, the auxiliary problem is well-posed since $(\mathbf{r}_i \cdot \mathbf{n}_i, 1)_{\partial\Omega_i} = (\mathcal{Q}_{h,i}\lambda_{h,\gamma}, 1)_{\partial\Omega_i} = 0$. Now, we choose $\mathbf{v} = \tilde{\Pi}_i \mathbf{r}_i$ in (2.8), to obtain

$$\begin{aligned} ||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma^2 &= (\lambda_{h,\gamma}, \tilde{\Pi}_i \mathbf{r}_i \cdot \mathbf{n}_i)_\gamma = -a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}), \tilde{\Pi}_i \mathbf{r}_i) + b_i(\tilde{\Pi}_i \mathbf{r}_i, p_{h,i}^*(\lambda_{h,\gamma})) \\ &= -a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}), \tilde{\Pi}_i \mathbf{r}_i), \leq C C_{\mathbf{K}} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma})||_{\Omega_i} ||\mathbf{r}_i||_{1/2,\Omega_i} + \alpha_\gamma^{-1} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i||_\gamma ||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma \\ &\leq \left(C C_{\mathbf{K}} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma})||_{\Omega_i} + \alpha_\gamma^{-1} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i||_\gamma\right) ||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma \\ &\leq \left(C \frac{C_{\mathbf{K}}}{\sqrt{c_{\mathbf{K}}}} + \frac{1}{\sqrt{\alpha_\gamma}}\right) \sqrt{a_i(\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}), \mathbf{u}_{h,i}^*(\lambda_{h,\gamma}))} ||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma, \end{aligned} \tag{4.12}$$

where we used (4.9), assumption (**A2**) and the elliptic regularity (2.4) together with

$$||\mathbf{r}_i||_{1/2,\Omega_i} \lesssim ||\mathcal{Q}_{h,i}\lambda_{h,\gamma}||_\gamma. \tag{4.13}$$

The bound (4.12) in combination with (4.8)-(4.9) and (2.5) delivers the lower bound in (4.7). □

As announced in the introduction, it is interesting to study the robustness of Algorithm 3.1 and Algorithm 3.5 for the limiting case in which the coefficient $\alpha_\gamma \to \infty$ in the transmission conditions (1.3). This case is physically corresponding to a continuous pressure over the fracture interface.

Lemma 4.3 (Parameter-robustness ($\alpha_\gamma \to \infty$)). *In the case of continuous pressure across γ, there exists a constant $C_2 > 0$ such that, for all $\lambda_{h,\gamma} \in M_{h,\gamma}$,*

$$C_2 \frac{c_{\mathbf{K}}}{C_{\mathbf{K}}^2} ||\lambda_{h,\gamma}||_\gamma^2 \leq s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) \leq \frac{C_{\mathrm{dTr}}^2}{c_{\mathbf{K}}} h^{-1} ||\lambda_{h,\gamma}||_\gamma^2. \tag{4.14}$$

Proof. Recalling the definition (2.10) of s_γ, we have

$$0 \leq s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) = -\sum_{i=1}^{2} (\lambda_{h,\gamma}, \mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i)_\gamma \leq \sum_{i=1}^{2} ||\mathbf{u}_{h,i}^*(\lambda_{h,\gamma}) \cdot \mathbf{n}_i||_\gamma ||\lambda_{h,\gamma}||_\gamma, \tag{4.15}$$

$$\leq \sum_{i=1}^{2} C_{\mathrm{dTr}} h^{-1/2} ||\mathbf{u}_{h,i}(\lambda_{h,\gamma})||_{\Omega_i} ||\lambda_{h,\gamma}||_\gamma, \tag{4.16}$$

where in that case we used the discrete trace inequality (4.3). This result together with (4.8) and (4.9) leads to the upper bound in (4.14). By inspection of the proof of Lemma (4.2), starting as in (4.12) we promptly get the lower bound of (4.14). □

In the following, we denote by $||\cdot||_{s,\gamma}$ the induced semi-norm from s_γ on $L^2(\gamma)$,

$$||\mu||_{s,\gamma} := s_\gamma(\mu, \mu)^{1/2}, \quad \forall \mu \in L^2(\gamma). \tag{4.17}$$

We will also consider the following discrete norms:

$$||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{0,\tau,\star}^2 := ||\mathbf{K}_\gamma^{-\frac{1}{2}} \mathbf{v}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}} \mathbf{v}_{h,\gamma}||_\gamma^2 + ||\tau^{-\frac{1}{2}} \mu_{h,\gamma}||_\gamma^2, \tag{4.18a}$$

$$||\mathbf{v}_{h,\gamma}||_{\mathbf{V}_{h,\gamma}}^2 := ||\mathbf{K}_\gamma^{-\frac{1}{2}} \mathbf{v}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}} \mathbf{v}_{h,\gamma}||_\gamma^2 + ||\tau^{\frac{1}{2}} \nabla_\tau \cdot \mathbf{v}_{h,\gamma}||_\gamma^2, \tag{4.18b}$$

$$||\mu_{h,\gamma}||_{M_{h,\gamma}}^2 := ||\mu_{h,\gamma}||_{s,\gamma}^2 + ||\tau^{-\frac{1}{2}} \mu_{h,\gamma}||_\gamma^2, \tag{4.18c}$$

$$||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}^2 := ||\mathbf{v}_{h,\gamma}||_{\mathbf{V}_{h,\gamma}}^2 + ||\mu_{h,\gamma}||_{M_{h,\gamma}}^2. \tag{4.18d}$$

We start with the estimate below.

Lemma 4.4 (Inverse energy estimates). *There holds for all $(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,*

$$||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} \leq \sqrt{\max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + \alpha_\gamma \tau))} ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{0,\tau,\star}. \tag{4.19}$$

Furthermore, if $1/\alpha_\gamma \to 0$, there holds

$$||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} \leq \sqrt{\max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + C_{\mathrm{dTr}}^2 c_{\mathbf{K}}^{-1} \tau h^{-1}))} ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{0,\tau,\star}. \tag{4.20}$$

Proof. Owing to the inverse inequality (4.4), together with (4.7), we obtain (4.19), and if $\alpha_\gamma \to \infty$, we make use (4.14) to get (4.20). □

The following results are immediately verified.

Lemma 4.5 (Boundedness on A_γ). *There holds for all $(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})$, $(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,*

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) \leq ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} ||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}. \tag{4.21}$$

Lemma 4.6 (Positivity on A_γ). *There holds for all $(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,*

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})) = ||\mathbf{K}_\gamma^{-\frac{1}{2}} \mathbf{u}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}} \mathbf{u}_{h,\gamma}||_\gamma^2 + ||\tau^{-\frac{1}{2}} \lambda_{h,\gamma}||_\gamma^2. \tag{4.22}$$

The above results are then used to prove the following stability estimate for $A_\gamma + s_\gamma$.

Theorem 4.7 (Stability results). *Let* $(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$,

$$\frac{1}{6(1+\tau\alpha_\gamma)^2}||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} \leq \sup_{(\mathbf{v}_{h,\gamma},\mu_{h,\gamma})\in \mathbf{V}_{h,\gamma}\times M_{h,\gamma}} \frac{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma})}{||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}}. \tag{4.23}$$

If $\alpha_\gamma \to \infty$, *we have*

$$\frac{1}{6(1+C_{\mathrm{dTr}}^2 c_{\mathbf{K}}^{-1}\frac{\tau}{h})^2}||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} \leq \sup_{(\mathbf{v}_{h,\gamma},\mu_{h,\gamma})\in \mathbf{V}_{h,\gamma}\times M_{h,\gamma}} \frac{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma})}{||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}}. \tag{4.24}$$

Proof. Let us first recall this inf-sup condition; given $\lambda_{h,\gamma} \in M_{h,\gamma}$, we construct an element $\mathbf{r}_{h,\gamma} \in \mathbf{V}_{h,\gamma}$ such that

$$b_\gamma(\mathbf{r}_{h,\gamma}, \lambda_{h,\gamma}) = ||\lambda_{h,\gamma}||_\gamma^2, \text{ and } ||\lambda_{h,\gamma}||_\gamma \leq C(\gamma)||\mathbf{r}_{h,\gamma}||_\gamma. \tag{4.25}$$

Let $\Psi_\gamma \in H_0^2(\gamma)$ be the solution of $-\Delta_\tau\Psi_\gamma = \tau^{-1}\lambda_{h,\gamma}$. Pose $\mathbf{r}_\gamma = -\nabla_\tau\Psi_\gamma$ and let $\mathbf{r}_{h,\gamma} = \Pi_{h,\gamma}\mathbf{r}_\gamma$, where $\Pi_{h,\gamma}$ is the Raviart-Thomas projection onto $\mathbf{V}_{h,\gamma}$ [14,38]. Then, we have $\nabla_\tau\cdot\mathbf{r}_{h,\gamma} = \Pi_{h,\gamma}\nabla_\tau\cdot\mathbf{r}_\gamma = \tau^{-1}\lambda_{h,\gamma}$. Hence, $b_\gamma(\mathbf{r}_{h,\gamma}, \lambda_{h,\gamma}) = ||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2$. Furthermore, we have $||\mathbf{r}_{h,\gamma}||_\gamma^2 = ||\Pi_{h,\gamma}\mathbf{r}_\gamma||_\gamma^2 \leq C||\mathbf{r}_\gamma||_{1,\gamma}^2 = C||\nabla_\tau\Psi_\gamma||_{1,\gamma}^2 \leq C||\Psi_\gamma||_{2,\gamma}^2 \leq C(\gamma)||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2$.

Now, consider $\delta_1, \delta_2 > 0$, and let $\mathbf{v}_{h,\gamma} = \mathbf{u}_{h,\gamma} - \delta_2\mathbf{r}_{h,\gamma}$ and $\mu_{h,\gamma} = \lambda_{h,\gamma} + \delta_1\tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma}$, where $\mathbf{r}_{h,\gamma}$ is from (4.25). We have

$$\begin{aligned}\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma}) &= \{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma})\}\\ &+ \delta_1\{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), \tau(\mathbf{0}, \nabla_\tau\cdot\mathbf{u}_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})\}\\ &- \delta_2\{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{r}_{h,\gamma}, 0))\}.\end{aligned} \tag{4.26}$$

For the first term on the right-hand side of (4.26), we obtain using estimate (4.22) together with (4.17),

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \lambda_{h,\gamma}) = ||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2 + ||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2 + ||\lambda_{h,\gamma}||_{s,\gamma}^2.$$

For the second term, we get for all $\epsilon_1 > 0$,

$$\begin{aligned}&\{\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), \tau(\mathbf{0}, \nabla_\tau\cdot\mathbf{u}_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})\}\\ &\quad= ||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma^2 + (\lambda_{h,\gamma}, \nabla_\tau\cdot\mathbf{u}_{h,\gamma})_\gamma + s_\gamma(\lambda_{h,\gamma}, \tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})\\ &\quad\geq ||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma^2 - ||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma - \alpha_\gamma\tau||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma\\ &\quad\geq (1-\epsilon_1\frac{(1+\tau\alpha_\gamma)}{2})||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma^2 - \frac{(1+\tau\alpha_\gamma)}{2\epsilon_1}||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2.\end{aligned} \tag{4.27}$$

where we have used the continuity of s_γ, i.e.,

$$s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma}) \leq ||\lambda_{h,\gamma}||_{s,\gamma}||\mu_{h,\gamma}||_{s,\gamma} \leq \alpha_\gamma||\lambda_{h,\gamma}||_\gamma||\mu_{h,\gamma}||_\gamma. \tag{4.28}$$

For the last term, using (**A2**) together with (4.25) (first equation), we obtain for all $\epsilon_2 > 0$,

$$\begin{aligned}\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{r}_{h,\gamma}, 0)) &\leq \frac{1}{2\epsilon_2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2) + \frac{\epsilon_2}{2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{r}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{r}_{h,\gamma}||_\gamma^2) - b_\gamma(\mathbf{r}_{h,\gamma}, \lambda_{h,\gamma})\\ &= \frac{1}{2\epsilon_2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2) + \frac{\epsilon_2}{2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{r}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{r}_{h,\gamma}||_\gamma^2) - ||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2\\ &\leq \frac{1}{2\epsilon_2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2 + ||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2) + \frac{\epsilon_2}{2}(C_{\mathbf{K},\gamma} + L_\gamma)C(\gamma)||\mathbf{r}_{h,\gamma}||_\gamma^2 - ||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2.\end{aligned} \tag{4.29}$$

Thus, with (4.25) (second equation),

$$-\delta_2\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma},\lambda_{h,\gamma}),(\mathbf{r}_{h,\gamma},0)\geq\delta_2\left(1-\epsilon_2\frac{C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)}{2}\right)||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2-\frac{\delta_2}{2\epsilon_2}(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2+||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2). \tag{4.30}$$

Collecting the previous results we get

$$\begin{aligned}&\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma},\lambda_{h,\gamma}),(\mathbf{v}_{h,\gamma},\mu_{h,\gamma}))+s_\gamma(\lambda_{h,\gamma},\mu_{h,\gamma})\\&\quad\geq\left(1-\frac{\delta_2}{2\epsilon_2}\right)\left(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2+||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2\right)+\delta_1\left(1-\epsilon_1\frac{(1+\tau\alpha_\gamma)}{2}\right)||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma^2\\&\qquad+\left(1-\delta_1\frac{(1+\tau\alpha_\gamma)}{2\epsilon_1}\right)||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2+||\lambda_{h,\gamma}||_{s,\gamma}^2+\delta_2\left(1-\epsilon_2\frac{C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)}{2}\right)||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2.\end{aligned} \tag{4.31}$$

Now, let us choose the parameters ϵ_i and δ_i such that all the norms in (4.31) are multiplied by positive coefficients. We choose $\epsilon_1=1/(1+\tau\alpha_\gamma)$ and $\delta_1=1/(1+\tau\alpha_\gamma)^2$, and then $\epsilon_2=2/[C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)]$ and $\delta_2=2/[C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)+1]$, to get

$$\begin{aligned}&\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma},\lambda_{h,\gamma}),(\mathbf{v}_{h,\gamma},\mu_{h,\gamma}))+s_\gamma(\lambda_{h,\gamma},\mu_{h,\gamma})\\&\quad\geq\frac{C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)+2}{2(C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)+1)}\left(||\mathbf{K}_\gamma^{-\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2+||L_\gamma^{\frac{1}{2}}\mathbf{u}_{h,\gamma}||_\gamma^2\right)+\frac{1}{2(1+\tau\alpha_\gamma)^2}||\tau^{\frac{1}{2}}\nabla_\tau\cdot\mathbf{u}_{h,\gamma}||_\gamma^2\\&\qquad+\frac{1}{2}||\tau^{-\frac{1}{2}}\lambda_{h,\gamma}||_\gamma^2+||\lambda_{h,\gamma}||_{s,\gamma}^2.\end{aligned} \tag{4.32}$$

Thus,

$$\mathcal{A}_\gamma((\mathbf{u}_{h,\gamma},\lambda_{h,\gamma}),(\mathbf{v}_{h,\gamma},\mu_{h,\gamma}))+s_\gamma(\lambda_{h,\gamma},\mu_{h,\gamma})\geq\frac{1}{2(1+\tau\alpha_\gamma)^2}||(\mathbf{u}_{h,\gamma},\lambda_{h,\gamma})||_{1,\tau,\star}^2. \tag{4.33}$$

We also have

$$\begin{aligned}||(\mathbf{v}_{h,\gamma},\mu_{h,\gamma})||_{1,\tau,\star}&=||(\mathbf{u}_{h,\gamma}-\delta_2\mathbf{r}_{h,\gamma},\lambda_{h,\gamma}+\delta_1\tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})||_{1,\tau,\star}\\&\leq||(\mathbf{u}_{h,\gamma},\lambda_{h,\gamma})||_{1,\tau,\star}+\delta_1||(\mathbf{0},\tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})||_{1,\tau,\star}+\delta_2||(\mathbf{r}_{h,\gamma},0)||_{1,\tau,\star}.\end{aligned}$$

With simple calculations, it is inferred that

$$\delta_1||(\mathbf{0},\tau\nabla_\tau\cdot\mathbf{u}_{h,\gamma})||_{1,\tau,\star}\leq\frac{1}{(1+\tau\alpha_\gamma)^{\frac{3}{2}}}||(\mathbf{u}_{h,\gamma},\lambda_{h,\gamma})||_{1,\tau,\star}, \tag{4.34a}$$

$$\delta_2||(\mathbf{r}_{h,\gamma},0)||_{1,\tau,\star}\leq\frac{2\sqrt{C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)+1}}{C(\gamma)(C_{\mathbf{K},\gamma}+L_\gamma)+2}||(\mathbf{u}_{h,\gamma},\lambda_{h,\gamma})||_{1,\tau,\star}. \tag{4.34b}$$

This implies that we have

$$||(\mathbf{v}_{h,\gamma},\mu_{h,\gamma})||_{1,\tau,\star}\leq3||(\mathbf{u}_{h,\gamma},\lambda_{h,\gamma})||_{1,\tau,\star}. \tag{4.35}$$

This result together with (4.33) leads to (4.23). For the limit case when $\alpha_\gamma\to\infty$, we repeat the same lines as before while using (4.14) instead of (4.7), we promptly arrive to (4.24). □

Lemma 4.8 (Well-posedness of the DD scheme). *The domain decomposition scheme* (4.5) *is well-posed, and all eigenvalues of the induced system $\mathcal{A}_\gamma+s_\gamma$ are bounded away from zero.*

Proof. The matrix associated to $\mathcal{A}_\gamma+s_\gamma$ is non-singular, that is to say that the system (4.5) has a unique solution. Moreover, the stability estimate (4.23) (also (4.24)) guarantees that the lowest eigenvalue is bounded away from zero. □

Let us comment on the robustness of the stability estimate in Theorem 4.7. First, (4.23) states that, regardless of the choice of the space and time discretization, the stability constant with respect to the norm $||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star}$ is independent of the coefficients $\mathbf{K}$, $\mathbf{K}_\gamma$, and the stabilization parameter L_γ. One can also show that this estimate is asymptotically robust and bounded independently of $(\tau, \alpha_\gamma, h) \to 0$ and the stability constant tends to $1/6$. The only issue can happen having a large coefficient α_γ, but this case is resolved in (4.24). Therein, as the ratio $\tau/h \to 0$, the stability constant is approximately $1/6$.

Following the approach of Ern and Guermond [23], we now provide an estimate for the condition number of the stiffness matrix associated with the domain decomposition system $\mathcal{A}_\gamma + s_\gamma$. This condition number estimate is important in our analysis as any algorithm is stable if every step is well-conditioned. This will also encourage the development of the flux basis framework in Section 6. Let us first introduce some basic notation in order to provide the definition of the condition number. We recall the stiffness matrix $\mathcal{A}_{\mathrm{DD}}$ introduced in (3.2) associated with the domain decomposition scheme (4.5),

$$(\mathcal{A}_{\mathrm{DD}} V, W)_N := \mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma}), \tag{4.36}$$

for all $(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})$, $(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma}) \in \mathbf{V}_{h,\gamma} \times M_{h,\gamma}$, where $(V, W)_N := \sum_{i=1}^N V_i W_i$ denotes the inner product in $\mathbb{R}^N$ and $|V|_N^2 := (V, V)_N$ is the corresponding Euclidean norm. The condition number is defined by

$$\kappa(\mathcal{A}_{\mathrm{DD}}) := |\mathcal{A}_{\mathrm{DD}}|_N |\mathcal{A}_{\mathrm{DD}}|_N^{-1}, \tag{4.37}$$

where

$$|\mathcal{A}_{\mathrm{DD}}|_N := \sup_{V \in \mathbb{R}^N \setminus \mathbf{0}} \sup_{W \in \mathbb{R}^N \setminus \mathbf{0}} \frac{(\mathcal{A}_{\mathrm{DD}} V, W)_N}{|V|_N |W|_N}, \tag{4.38}$$

which is equivalent to

$$|\mathcal{A}_{\mathrm{DD}}|_N := \sup_{V \in \mathbb{R}^N \setminus \mathbf{0}} \frac{|\mathcal{A}_{\mathrm{DD}}|_N}{|V|_N}. \tag{4.39}$$

We recall the following estimate that holds true for a conforming, quasi-uniform mesh $\mathcal{T}_h$ [23]; there exists $c_\mu, C_\mu > 0$ such that the following equivalence holds

$$c_\mu h^{d/2} |V|_N \leq ||V||_{0,\tau,\star} \leq C_\mu h^{d/2} |V|_N. \tag{4.40}$$

Theorem 4.9 (Condition number estimate). *The condition number of the domain decomposition scheme* (4.5) *is bounded as*

$$\kappa(\mathcal{A}_{\mathrm{DD}}) \lesssim 6(1 + \tau\alpha_\gamma)^2 \max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + \alpha_\gamma \tau)). \tag{4.41}$$

Furthermore, if $\alpha_\gamma \to \infty$,

$$\kappa(\mathcal{A}_{\mathrm{DD}}) \lesssim 6(1 + C_{\mathrm{dTr}}^2 c_{\mathbf{K}}^{-1} \tau h^{-1})^2 \max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + C_{\mathrm{dTr}}^2 c_{\mathbf{K}}^{-1} \tau h^{-1})). \tag{4.42}$$

Proof. We need to bound $|\mathcal{A}_{\mathrm{DD}}|_N$ and $|\mathcal{A}_{\mathrm{DD}}|_N^{-1}$. By definition, for all $V, W \in \mathbb{R}^N$,

$$\begin{aligned}
(\mathcal{A}_{\mathrm{DD}} V, W)_N &= \mathcal{A}_\gamma((\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma}), (\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})) + s_\gamma(\lambda_{h,\gamma}, \mu_{h,\gamma}), \\
&\leq ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} ||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}, \\
&\leq \max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + \alpha_\gamma \tau)) ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{0,\tau,\star} ||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{0,\tau,\star}, \\
&\lesssim \max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + \alpha_\gamma \tau)) h^d |V|_N |W|_N,
\end{aligned} \tag{4.43}$$

where the estimate (4.19) and the equivalence (4.40) were successively used. Consequently,

$$|\mathcal{A}_{\mathrm{DD}}|_N \lesssim \max((1 + C_{\mathrm{inv}} c_{\mathbf{K},\gamma} \tau h^{-2}), (1 + \alpha_\gamma \tau)) h^d. \tag{4.44}$$

To estimate $|\mathcal{A}_{\mathrm{DD}}|_N^{-1}$, start from (4.23) and use (4.40) to get

$$\begin{aligned}
(\mathcal{A}_{\mathrm{DD}} V, W)_N &\geq \frac{1}{6(1 + \tau\alpha_\gamma)^2} ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{1,\tau,\star} ||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{1,\tau,\star}, \\
&\geq \frac{1}{6(1 + \tau\alpha_\gamma)^2} ||(\mathbf{u}_{h,\gamma}, \lambda_{h,\gamma})||_{0,\tau,\star} ||(\mathbf{v}_{h,\gamma}, \mu_{h,\gamma})||_{0,\tau,\star}, \\
&\gtrsim \frac{h^d}{6(1 + \tau\alpha_\gamma)^2} |V|_N |W|_N,
\end{aligned} \tag{4.45}$$

and hence $|V|_N \lesssim 6(1+\tau\alpha_\gamma)^2 h^{-d}|\mathcal{A}_{\rm DD}V|_N$. Now setting $V = \mathcal{A}_{\rm DD}^{-1}W$, we easily conclude that $|\mathcal{A}_{\rm DD}^{-1}|_N \lesssim 6(1+\tau\alpha_\gamma)^2 h^{-d}$. Combining estimates for $|\mathcal{A}_{\rm DD}^{-1}|_N$ and $|\mathcal{A}_{\rm DD}|_N$ we get (4.41). The estimate (4.42) of the limiting case $\alpha_\gamma \to \infty$ is obtained similarly by using (4.20) and (4.24) in the proof. □

4.2 Convergence of MoLDD-scheme

The second step of our analysis is to prove the convergence of Algorithm 3.1. The idea is to prove that this algorithm is a contraction and then apply the Banach fixed-point theorem [46]. To this purpose, we let $\delta^k_{\mathbf{u},h} = \mathbf{u}^{n,k}_{h,\gamma} - \mathbf{u}^{n,k-1}_{h,\gamma}$ and $\delta^k_{\lambda,h} = \lambda^{n,k}_{h,\gamma} - \lambda^{n,k-1}_{h,\gamma}$ be the differences between the solutions at iteration k and $k-1$ of the problem (3.1), respectively.

Theorem 4.10 (Convergence of MoLDD-scheme). *Assuming that Assumptions* (**A1**)–(**A5**) *hold true and that* $L_\gamma(\zeta) = L_\xi/2(1-\zeta)$*, with a parameter* $\zeta \in [0,1)$*, Algorithm 3.1 defines a contraction given by*

$$||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \left(\frac{L_\gamma}{2} + c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma \leq \left(\frac{L_\gamma}{2} - \zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma, \tag{4.46}$$

where ζ *is chosen to improve the convergence rate of the scheme. Furthermore, the limit is the unique solution of* (2.11)*.*

Proof. By subtracting (3.1) at k from the ones at $k-1$, we obtain

$$(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \mathbf{v})_\gamma + L_\gamma(\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}, \mathbf{v})_\gamma + a_\gamma(\delta^k_{\mathbf{u},h}, \mathbf{v}) - b_\gamma(\mathbf{v}, \delta^k_{\lambda,h}) = 0 \quad \forall \mathbf{v} \in \mathbf{V}_{h,\gamma}, \tag{4.47a}$$

$$c_\gamma(\delta^k_{\lambda,h}, \mu) + \tau b_\gamma(\delta^k_{\mathbf{u},h}, \mu) + \tau s_\gamma(\delta^k_{\lambda,h}, \mu) = 0 \quad \forall \mu \in M_{h,\gamma}. \tag{4.47b}$$

Taking $\mu = \delta^k_{\lambda,h}$ in (4.47b) and $\mathbf{v} = \tau\delta^k_{\mathbf{u},h}$ in (4.47a) and summing the equations gives

$$||\delta^k_{\lambda,h}||^2 + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \tau a_\gamma(\delta^k_{\mathbf{u},h}, \mathbf{v}) + \tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \delta^k_{\mathbf{u},h})_\gamma + L_\gamma\tau(\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}, \delta^k_{\mathbf{u},h})_\gamma = 0.$$

Following [50], we let $\zeta \in [0,1)$ and split the third term while applying the lower bound of $\mathbf{K}_\gamma^{-1}$,

$$\begin{aligned} ||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + c_{\mathbf{K},\gamma}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \zeta\tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \delta^{k-1}_{\mathbf{u},h})_\gamma \\ + (1-\zeta)\tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \delta^{k-1}_{\mathbf{u},h})_\gamma + \tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h})_\gamma \\ + L_\gamma\tau(\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}, \delta^k_{\mathbf{u},h})_\gamma \leq 0. \end{aligned} \tag{4.48a}$$

We use the monotonicity and Lipschitz continuity of ξ given by (**A1**) together with the identity (4.1), to get

$$\begin{aligned} ||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + c_{\mathbf{K},\gamma}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \zeta\xi_m\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{(1-\zeta)}{L_\xi}\tau||\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma})||^2_\gamma \\ + \frac{L_\gamma}{2}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \frac{L_\gamma}{2}\tau||\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}||^2_\gamma \leq \frac{L_\gamma}{2}\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma - \tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma}), \delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h})_\gamma. \end{aligned} \tag{4.48b}$$

We apply Young's inequality (4.2) for the last term in the right-hand side to obtain

$$\begin{aligned} ||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + c_{\mathbf{K},\gamma}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \zeta\xi_m\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma \\ + \frac{(1-\zeta)}{L_\xi}\tau||\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma})||^2_\gamma + \frac{L_\gamma}{2}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \frac{L_\gamma}{2}\tau||\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}||^2_\gamma \\ \leq \frac{L_\gamma}{2}\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{L_\gamma}{2}\tau||\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{1}{2L_\gamma}\tau||\xi(\mathbf{u}^{n,k-1}_{h,\gamma}) - \xi(\mathbf{u}^{n,k-2}_{h,\gamma})||^2_\gamma. \end{aligned} \tag{4.48c}$$

We choose $L_\gamma = L_\xi/2(1-\zeta)$, we immediately obtain (4.46). The inequality (4.46) imply that the sequence $\delta^{n,k}_{\lambda,h}$ tends to 0 in $L^2(\gamma)$ and $\delta^k_{\mathbf{u},h}$ tends to 0 in $\mathbf{L}^2(\gamma)$. Now we choose $\mu = \nabla_\tau \cdot \delta^k_{\mathbf{u},h}$ in (4.47b) to obtain

$$\begin{aligned} \tau||\nabla_\tau \cdot \delta^k_{\mathbf{u},h}||^2_\gamma &= -c_\gamma(\delta^k_{\lambda,h}, \nabla_\tau \cdot \delta^k_{\mathbf{u},h}) - \tau s_\gamma(\lambda^{n,k}_{h,\gamma}, \nabla_\tau \cdot \delta^k_{\mathbf{u},h}), \\ &\leq ||\delta^k_{\lambda,h}||_\gamma||\nabla_\tau \cdot \delta^k_{\mathbf{u},h}||_\gamma + \tau\alpha_\gamma||\delta^k_{\lambda,h}||_\gamma||\nabla_\tau \cdot \delta^k_{\mathbf{u},h}||_\gamma. \end{aligned}$$

Thus,

$$\tau||\nabla_\tau \cdot \delta^k_{\mathbf{u},h}||_\gamma \leq (\alpha_\gamma \tau + 1)||\delta^k_{\lambda,h}||_\gamma. \tag{4.49}$$

Hence, by (4.46), we have $||\nabla_\tau \cdot \delta^k_{\mathbf{u},h}||_\gamma$ tends to 0 in $L^2(\gamma)$. This shows that $\delta^k_{\mathbf{u},h}$ tends to 0 in $\mathbf{H}(\mathrm{div}_\tau, \gamma)$. □

Corollary 4.11 (Optimal MoLDD-convergence rate). *If $\xi_m > 0$, the minimum of the convergence rate of Algorithm 3.1 is reached for the optimal parameter*

$$\zeta^* = \underset{0<\zeta<1}{\arg\min} \rho(\zeta) = 1 + \frac{L_\xi \xi_m - \sqrt{(L_\xi \xi_m)^2 + 4L_\xi \xi_m^2 c_{\mathbf{K},\gamma} + 4L_\xi \xi_m c^2_{\mathbf{K},\gamma}}}{4\xi_m c_{\mathbf{K},\gamma}}, \tag{4.50a}$$

where $\rho(\zeta)$ is the convergence rate from (4.46)*,*

$$\rho(\zeta) = \frac{L_\gamma - 2\xi_m \zeta}{L_\gamma + 2c_{\mathbf{K},\gamma}} < 1. \tag{4.50b}$$

Therefore, the optimal stabilization parameter is given by

$$L_{\gamma,\mathrm{opt}} = \frac{L_\xi}{2(1-\zeta^*)}. \tag{4.50c}$$

Proof. Plugging $L_\gamma = L_\xi/2(1-\zeta)$ in the contraction estimate (4.46) leads to $||\delta^k_{\mathbf{u},h}||^2_\gamma \leq \rho(\zeta)||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma$, where

$$\rho(\zeta) = \frac{L_\xi - 4(1-\zeta)\xi_m \zeta}{L_\xi + 4(1-\zeta)c_{\mathbf{K},\gamma}} < 1, \tag{4.51}$$

which clearly can be minimal when choosing the optimal value of ζ. To calculate ζ^*, we differentiate (4.51) with respect to ζ and infer the resulting roots and we find that the minimum of (4.51) is obtained for the optimal choice given by (4.50a). Replacing back the resulting value into $L_\gamma(\zeta)$ delivers the optimal stabilization parameter (4.50c). □

Lemma 4.12 (Well-posedness of the mixed-dimensional problem). *There exists a unique solution to the mixed-dimensional problem* (2.6)*.*

Proof. Problem (2.11) is equivalent to (2.6). Since we know from Theorem (4.10) that (2.11) has a unique solution, this equivalence implies that (2.6) is uniquely solvable. □

We continue with some important remarks concerning the results above and the implications to the convergence rate of MoLDD-scheme.

Remark 4.13 (Dependence of the convergence rate). *Obviously, the rate of convergence* (4.50b) *depends only on the strength of the non-linearity (not on the domain decomposition as an inner solver) by means of the Lipschitz constant L_ξ, the lower bound ξ_m and the fracture permeability $\mathbf{K}_\gamma$. Particularly, the rate is **independent** of the fracture-matrix coupling parameter α_γ, the mesh size h and the time step τ.*

Remark 4.14 (Global convergence). *The convergence of MoLDD-scheme is global, i.e. independent of the initialization and particularly of the used inner DD solver (like GMRes). Nevertheless, it is obviously beneficial if one starts MoLDD-scheme iterations with the solution of the last time step.*

5 Analysis of ItLDD-scheme

We turn now to the analysis of the iterative LDD-scheme (Algorithm 3.5). In contrast to MoLDD-scheme, in which two levels of calculations (Linearization+DD) are necessary to achieve the required solution, the iterative LDD-scheme treats simultaneously the non-linearity and DD. We introduce $\delta^k_{\mathbf{u},h} := \mathbf{u}^{n,k}_{h,\gamma} - \mathbf{u}^n_{h,\gamma}$ and $\delta^k_{\lambda,h} := \lambda^{n,k}_{h,\gamma} - \lambda^n_{h,\gamma}$, stating the differences between the solution of the problem (3.1) at iteration k and the solution of the problem (2.11). Thus, the next result is to be understood as the convergence for the combined Linearization-DD processes.

Theorem 5.1 (Convergence of ItLDD-scheme). *Assuming that Assumptions* (**A1**)–(**A5**) *hold true and that* $L_{\gamma,u}(\zeta) = L_\xi/2(1-\zeta)$*, where* ζ *is a parameter to be optimized in* $[0,1)$*, and* $L_{\gamma,p} \geq \alpha_\gamma$*, the ItLDD-scheme given by Algorithm 3.5 is linearly convergent. There holds*

$$\left(1+\tau\frac{L_{\gamma,p}}{2}\right)||\delta^k_{\lambda,h}||^2_\gamma + \frac{\tau}{2}||\delta^k_{\lambda,h}||^2_{s,\gamma} + \left(\frac{L_{\gamma,u}}{2}+c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma \leq \left(\frac{L_{\gamma,u}}{2}-\zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \tau\frac{L_{\gamma,p}}{2}||\delta^{k-1}_{\lambda,h}||^2_\gamma. \tag{5.1}$$

Proof. By subtracting (3.1) at the iteration k from (2.11), we obtain

$$(\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma}),\mathbf{v})_\gamma + L_{\gamma,u}(\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h},\mathbf{v})_\gamma + a_\gamma(\delta^k_{\mathbf{u},h},\mathbf{v}) - b_\gamma(\mathbf{v},\delta^k_{\lambda,h}) = 0 \quad \forall \mathbf{v}\in\mathbf{V}_{h,\gamma}, \tag{5.2a}$$

$$c_\gamma(\delta^k_{\lambda,h},\mu) + \tau L_{\gamma,p}(\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h},\mu)_\gamma + \tau s_\gamma(\delta^{k-1}_{\lambda,h},\mu) + \tau b_\gamma(\delta^k_{\mathbf{u},h},\mu) = 0 \quad \forall \mu\in M_{h,\gamma}. \tag{5.2b}$$

Taking $\mathbf{v}=\tau\delta^k_{\mathbf{u},h}$ in (5.2a) and $\mu=\delta^k_{\lambda,h}$ in (5.2b), and summing up the equations gives

$$\begin{aligned}&||\delta^k_{\lambda,h}||^2_\gamma + \tau L_{\gamma,p}(\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h},\delta^k_{\lambda,h})_\gamma + \tau s_\gamma(\delta^{k-1}_{\lambda,h},\delta^k_{\lambda,h})\\ &\quad+\tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma}),\delta^k_{\mathbf{u},h})_\gamma + L_{\gamma,u}\tau(\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h},\delta^k_{\mathbf{u},h})_\gamma + \tau a_\gamma(\delta^k_{\mathbf{u},h},\delta^k_{\mathbf{u},h}) = 0.\end{aligned} \tag{5.3}$$

For any $\zeta\in[0,1)$, this is equivalent to,

$$\begin{aligned}&||\delta^k_{\lambda,h}||^2_\gamma + \tau L_{\gamma,p}(\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h},\delta^k_{\lambda,h})_\gamma + \tau s_\gamma(\delta^k_{\lambda,h},\delta^k_{\lambda,h}) + \tau\zeta(\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma}),\delta^{k-1}_{\mathbf{u},h})_\gamma\\ &\quad+\tau(1-\zeta)(\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma}),\delta^{k-1}_{\mathbf{u},h})_\gamma + L_{\gamma,u}\tau(\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h},\delta^k_{\mathbf{u},h})_\gamma + \tau a_\gamma(\delta^k_{\mathbf{u},h},\delta^k_{\mathbf{u},h})\\ &\quad=-\tau s_\gamma(\delta^{k-1}_{\lambda,h}-\delta^k_{\lambda,h},\delta^k_{\lambda,h}) - \tau(\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma}),\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h})_\gamma.\end{aligned} \tag{5.4}$$

We apply the lower bound in the last term of the left-hand side and then use the monotonicity and Lipschitz continuity of the operator ξ, followed by Cauchy-Schwarz and Young's inequalities in the second term of the right-hand side, to get

$$\begin{aligned}&\left(1+\tau\frac{L_{\gamma,p}}{2}\right)||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \tau\frac{L_{\gamma,p}}{2}||\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h}||^2_\gamma + c_{\mathbf{K},\gamma}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \zeta\xi_m||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma\\ &\quad+\frac{(1-\zeta)}{L_\xi}\tau||\xi(\mathbf{u}^{n,k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma})||^2_\gamma + \frac{L_{\gamma,u}}{2}\tau||\delta^k_{\mathbf{u},h}||^2_\gamma + \frac{L_{\gamma,u}}{2}\tau||\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h}||^2_\gamma\\ &\quad\leq\frac{L_{\gamma,u}}{2}\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{L_{\gamma,p}}{2}\tau||\delta^{k-1}_{\lambda,h}||^2_\gamma + \frac{L_{\gamma,u}}{2}\tau||\delta^k_{\mathbf{u},h}-\delta^{k-1}_{\mathbf{u},h}||^2_\gamma - \tau s_\gamma(\delta^{k-1}_{\lambda,h}-\delta^k_{\lambda,h},\delta^k_{\lambda,h})\\ &\quad\quad+\frac{1}{2L_{\gamma,u}}\tau||\xi(\mathbf{u}^{k-1}_{h,\gamma})-\xi(\mathbf{u}^n_{h,\gamma})||^2_\gamma.\end{aligned} \tag{5.5}$$

The continuity of s_γ gives

$$|s_\gamma(\delta^{k-1}_{\lambda,h}-\delta^k_{\lambda,h},\delta^k_{\lambda,h})| \leq ||\delta^k_{\lambda,h}||_{s,\gamma}||\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h}||_{s,\gamma} \leq \alpha_\gamma^{1/2}||\delta^k_{\lambda,h}||_{s,\gamma}||\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h}||_\gamma, \tag{5.6}$$

where we have used (4.7). Applying Young's inequality to (5.6) and plugging the result in (5.5), then choose $L_{\gamma,u}=L_\xi/2(1-\zeta)$, it is inferred,

$$\begin{aligned}&\left(1+\tau\frac{L_{\gamma,p}}{2}\right)||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \tau\frac{L_{\gamma,p}}{2}||\delta^k_{\lambda,h}-\delta^{k-1}_{\lambda,h}||^2_\gamma + \left(\frac{L_{\gamma,u}}{2}+c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma\\ &\quad\leq\left(\frac{L_{\gamma,u}}{2}-\zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \tau\frac{L_{\gamma,p}}{2}||\delta^{k-1}_{\lambda,h}||^2_\gamma + \tau\frac{\alpha_\gamma}{2}||\delta^{k-1}_{\lambda,h}-\delta^k_{\lambda,h}||^2_\gamma + \frac{\tau}{2}||\delta^k_{\lambda,h}||^2_{s,\gamma}.\end{aligned} \tag{5.7}$$

We let $L_{\gamma,p}\geq\alpha_\gamma$, to obtain the estimate (5.1) which is clearly a contraction. We finally repeat the same techniques as in (4.49), to get that $||\nabla_\tau\cdot\delta^k_{\mathbf{u},h}||_\gamma$ tends to 0 in $L^2(\gamma)$. This altogether shows that $\delta^{n,k}_{\lambda,h}$ tends to 0 in $L^2(\gamma)$ and $\delta^k_{\mathbf{u},h}$ tends to 0 in $\mathbf{H}(\mathrm{div}_\tau,\gamma)$. □

Remark 5.2 (Contraction factor). *Our contraction estimate shows that the strength of the non-linearity and the matrix fracture (DD) coupling controls the convergence rate. In practice, the contraction factor is better if we take into account the energy norm $\tau||\delta^k_{\lambda,h}||^2_{s,\gamma}/2$ using the bound (4.7). As the stabilization term is such that $L_{\gamma,p} \geq \alpha_\gamma$, thus, we have to study the robustness of the algorithm when $\alpha_\gamma \to \infty$, corresponding physically to the case of continuous pressure across the fracture.*

Lemma 5.3 (Contraction-robustness). *Assuming continuous pressure across γ ($\alpha_\gamma \to \infty$), then let $L_{\gamma,u}(\zeta) = L_\xi/2(1-\zeta)$ with ζ to be chosen in $[0,1)$, and $L_{\gamma,p} \geq C^2_{\mathrm{dTr}}/(c_{\mathbf{K}}h)$, the contraction (5.1) holds true for the ItLDD-scheme in Algorithm 3.5.*

Proof. Recall the estimate (5.5) which holds true in that case. We then estimate the coupling term $|s_\gamma(\delta^k_{\lambda,h} - \delta^{k-1}_{\lambda,h}, \delta^k_{\lambda,h})|$ with the help of (4.14),

$$|s_\gamma(\delta^k_{\lambda,h} - \delta^{k-1}_{\lambda,h}, \delta^k_{\lambda,h})| \leq ||\delta^k_{\lambda,h}||_{s,\gamma}||\delta^k_{\lambda,h} - \delta^{k-1}_{\lambda,h}||_{s,\gamma}, \leq C_{\mathrm{dTr}}c_{\mathbf{K}}^{-1/2}h^{-1/2}||\delta^k_{\lambda,h}||_{s,\gamma}||\delta^k_{\lambda,h} - \delta^{k-1}_{\lambda,h}||_\gamma. \tag{5.8}$$

We apply Young's inequality to (5.6) and replace the result in (5.5), while choosing $L_\gamma = L_\xi/2(1-\zeta)$,

$$\begin{aligned}\left(1 + \tau\frac{L_{\gamma,p}}{2}\right)||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \tau\frac{L_{\gamma,p}}{2}||\delta^k_{\lambda,h} - \delta^{k-1}_{\lambda,h}||^2_\gamma + \left(\frac{L_{\gamma,u}}{2} + c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma \\ \leq \left(\frac{L_{\gamma,u}}{2} - \zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{L_{\gamma,p}}{2}\tau||\delta^{k-1}_{\lambda,h}||^2_\gamma + \frac{C^2_{\mathrm{dTr}}}{c_{\mathbf{K}}}h^{-1}\frac{\tau}{2}||\delta^{k-1}_{\lambda,h} - \delta^k_{\lambda,h}||^2_\gamma + \frac{\tau}{2}||\delta^k_{\lambda,h}||^2_{s,\gamma}.\end{aligned}$$

We choose $L_{\gamma,p} \geq C^2_{\mathrm{dTr}}/(c_{\mathbf{K}}h)$, we end up with the contraction (5.1). The rest of the proof is as in Theorem 5.1. □

We complete our analysis of Algorithm 3.5 by giving alternative convergence results when $(h, 1/\alpha_\gamma) \to 0$, leading to extremely large stabilization parameter $L_{\gamma,p}$, which deteriorates the convergence rate of ItLDD scheme. These results are then important to show the robustness of the iterative LDD-scheme for extreme physical and/or discretization situations.

Proposition 5.4 (Alternative convergence results). *If $L_{\gamma,p} = 0$, and $L_{\gamma,u} = L_\xi/2(1-\zeta)$ with $\zeta \in [0,1)$, Algorithm 3.5 is convergent under the constraint on the time step $\tau \leq 1/\alpha_\gamma$. The following estimate for Algorithm 3.5 holds true and defines a contraction*

$$\left(1 - \frac{\alpha_\gamma}{2}\tau\right)||\delta^k_{\lambda,h}||^2_\gamma + \left(\frac{L_{\gamma,u}}{2} + c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma \leq \left(\frac{L_{\gamma,u}}{2} - \zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma + \frac{\alpha_\gamma}{2}\tau||\delta^{k-1}_{\lambda,h}||^2_\gamma. \tag{5.9}$$

Moreover, if $\alpha_\gamma \to \infty$, Algorithm 3.5 is convergent as the ratio $\tau/h \leq c_{\mathbf{K}}/C^2_{\mathrm{dTr}}(=: C^{-1}_{\gamma,s})$ holds true, and the resulting estimate is a contraction given by

$$\left(1 - \frac{C_{\gamma,s}}{2}\frac{\tau}{h}\right)||\delta^k_{\lambda,h}||^2 + \left(\frac{L_{\gamma,u}}{2} + c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2 \leq \left(\frac{L_{\gamma,u}}{2} - \zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2 + \frac{C_{\gamma,s}}{2}\frac{\tau}{h}||\delta^{k-1}_{\lambda,h}||^2. \tag{5.10}$$

Proof. We let $L_{\gamma,p} = 0$ in the estimate (5.3) to get

$$||\delta^k_{\lambda,h}||^2_\gamma + \tau(b_\gamma(\mathbf{u}^{n,k-1}_{h,\gamma}) - b_\gamma(\mathbf{u}^n_{h,\gamma}), \delta^k_{\mathbf{u},h})_\gamma + L_{\gamma,u}\tau(\delta^k_{\mathbf{u},h} - \delta^{k-1}_{\mathbf{u},h}, \delta^k_{\mathbf{u},h})_\gamma + \tau a_\gamma(\delta^k_{\mathbf{u},h}, \delta^k_{\mathbf{u},h}) = -\tau s_\gamma(\delta^{k-1}_{\lambda,h}, \delta^k_{\lambda,h}).$$

With the same techniques used to get (5.5), we get for $L_{\gamma,u} = L_\xi/2(1-\zeta)$ with $\zeta \in [0,1)$,

$$\begin{aligned}\left(1 + \tau\frac{L_{\gamma,p}}{2}\right)||\delta^k_{\lambda,h}||^2_\gamma + \tau||\delta^k_{\lambda,h}||^2_{s,\gamma} + \left(\frac{L_{\gamma,u}}{2} + c_{\mathbf{K},\gamma}\right)\tau||\delta^k_{\mathbf{u},h}||^2_\gamma \\ \leq \left(\frac{L_{\gamma,u}}{2} - \zeta\xi_m\right)\tau||\delta^{k-1}_{\mathbf{u},h}||^2_\gamma - \tau s_\gamma(\delta^{k-1}_{\lambda,h}, \delta^k_{\lambda,h}).\end{aligned} \tag{5.11}$$

The coupling term in the right-hand side is now estimated as follows

$$|s_\gamma(\delta^{k-1}_{\lambda,h}, \delta^k_{\lambda,h})| \leq \alpha_\gamma||\delta^{k-1}_{\lambda,h}||_\gamma||\delta^k_{\lambda,h}||_\gamma,$$

where we used (4.7). Applying Young inequality and inserting the result in (5.11), we infer (5.9). That of the second estimate (5.10), when $\alpha_\gamma \to \infty$, is obtained similarly to (5.9), but with using (4.14) to bound the coupling term. □

Remark 5.5 (Time step vs stabilization). *The constraint on the ratio τ/h is less restrictive than the constrain on the stabilization parameter $L_{\gamma,p}$ in Lemma 5.3. We also note that the constraint on the time step $\tau \leq 1/\alpha_\gamma$ may have the same implication on the convergence rate as taking $L_{\gamma,p} \geq \alpha_\gamma$ in Theorem 5.1. In practice, the choice between the two constraints may depend on the physical situation. All the results show a strong correlation between the Robin parameter α_γ, and the time step τ (or τ/h) or the stabilization parameter $L_{\gamma,p}$.*

6 The LDD Iterations with Multiscale Flux Basis Implementation

In this section, we propose an alternative to the matrix-free method (see Subsection 3.1) by forming the inter-dimensional map $\mathcal{S}_\gamma^{\mathrm{RtN}}$ based on the construction of a multiscale mortar flux basis (MFB) from [1,31]. We recall that the goal of this paper is to solve the reduced scheme of Definition 2.2 for: 1) different physical parameters and various realizations of the L-scheme parameters, 2) various PDEs by changing the non-linearity ξ, and 3) when computing and comparing the two LDD solvers (MoLDD *vs* ItLDD). We also recall that the dominant computational cost in the LDD algorithms comes from the subdomain solves to evaluate the action of $\mathcal{S}_\gamma^{\mathrm{RtN}}$ using Algorithm 3.3 (step 2(b)). These solves are required at each inner and outer iterations of Algorithm 3.1 and each iteration of Algorithm 3.5 (see Remark 3.4 and 3.6 for the overall cost). Therefore, the computation cost of the algorithms may become large since, first, the LDD solver may require a large number of iterations for complex problems, and second, we have seen that the condition number (4.41)-(4.42) of the linearized interface problem grows with refining the grids or increasing α_γ (normal permeability) and permeability contrast.

The construction of the inter-dimensional mapping is achieved by pre-computing and storing the flux subdomain responses, called *multiscale flux basis*, associated with each fracture pressure degree of freedom on each subdomain. We define $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$ to be the set of basis functions on the interface pressure space $M_{h,\gamma}$, where $\mathcal{N}_{h,\gamma}$ is the number of pressure degrees of freedom on γ [32]. As a result, on the fracture interface, we let $\mu_{h,\gamma} := \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \mu_{h,\gamma}^\ell \Phi_{h,\gamma}^\ell$, and compute the MFB functions corresponding to $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$ using the following algorithm:

Algorithm 6.1 (Assembly of the multiscale flux basis).

1. *Enter the basis* $(\Phi_{h,\gamma}^\ell)_{\ell=1}^{\mathcal{N}_{h,\gamma}}$. *Set* $\ell := 0$.
2. **Do**
   (a) *Increase* $\ell := \ell + 1$.
   (b) *Project* $\Phi_{h,\gamma}^\ell$ *on the subdomain boundary,* $\lambda_{h,i}^\ell = \mathcal{Q}_{h,i}(\Phi_{h,\gamma}^\ell)$.
   (c) *Solve problem* (2.8) *in each subdomain* Ω_i.
   (d) *Project the resulting flux onto the pressure space on the fracture,* $\Psi_{h,\gamma,i}^\ell := -\mathcal{Q}_{h,i}^{\mathrm{T}} \mathbf{u}_{h,i}^*(\lambda_{h,i}^\ell) \cdot \mathbf{n}_i$.
   While $\ell \leq \mathcal{N}_{h,\gamma}$.
3. *Form the multiscale flux basis for subdomain* Ω_i, *i.e.,* $\left\{\Psi_{h,\gamma,i}^1, \Psi_{h,\gamma,i}^2, \cdots, \Psi_{h,\gamma,i}^{\mathcal{N}_{h,\gamma}}\right\} \subset M_{h,\gamma}$.

Once the multiscale flux basis functions are constructed for each subdomain, the action of $\mathcal{S}_\gamma^{\mathrm{RtN}}$ is replaced by a linear combination of the multiscale flux basis functions $\Psi_{h,\gamma,i}^\ell$. Specifically, *at any time step* $n \geq 1$, *and at any iteration* $m \geq 1$ *of any of the algorithms,* for an interface datum $\lambda_{h,\gamma}^{n,m} \in M_{h,\gamma}$, we have $\lambda_{h,\gamma}^{n,m} = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \lambda_{h,\gamma}^{n,m,\ell} \Phi_{h,\gamma}^\ell$, and for $i \in \{1,2\}$,

$$\mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\lambda_{h,\gamma}^{n,m}) = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \lambda_{h,\gamma}^{n,m,\ell} \mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\Phi_{h,\gamma}^\ell) = \sum_{\ell=1}^{\mathcal{N}_{h,\gamma}} \lambda_{h,\gamma}^{n,m,\ell} \Psi_{h,\gamma,i}^\ell. \tag{6.1a}$$

We then compute the jump across the fracture

$$\mathcal{S}_\gamma^{\mathrm{RtN}}(\lambda_{h,\gamma}^{n,m}) = \sum_{i\in\{1,2\}} \mathcal{S}_{\gamma,i}^{\mathrm{RtN}}(\lambda_{h,\gamma}^{n,m}). \tag{6.1b}$$

We continue with some important remarks on the applicability of the MFB.

Remark 6.2 (Fracture network). *We observe that each fracture pressure basis function $\Phi^{\ell}_{h,\gamma}$ on the fracture interface corresponds to exactly two different multiscale flux basis functions, one for Ω_1 and one for Ω_2. For the case of a fracture network, say $\gamma = \cup_{i\neq j}\gamma_{ij}$, where γ_{ij} is the fracture between the subdomain Ω_i and Ω_j, the previous basis reconstruction is then applied independently on each fracture.*

Remark 6.3 (On the MFB - gain). *Note that* (6.1) *permits now retrieving the action of $\mathcal{S}^{\mathrm{RtN}}_{\gamma}$ on $M_{h,\gamma}$, for* ***any outer or inner iteration of the LDD solvers****, and* ***for all time steps*** *$n \in \{1, 2, \cdots, N\}$. Thus, the use of MFB eliminates the dependence between the total number of subdomain solves and the number of iterations of each LDD solver.*

Remark 6.4 (On the MFB - cost). *For large scale simulations, the multiscale (Robin-to-Neumann) functions are stored on the subdomain level then on different processors, so that the inter-dimensional mapping $\mathcal{S}^{\mathrm{RtN}}_{\gamma}$ need not be assembled. As detailed in Subsection 3.1, a Krylov method is used to solve for the GMRes update in* (3.3) *(for* (3.4) *for the ItLDD solver) which requires only the action of the Robin-to-Neumann operator on each Krylov vector. We have explained through the paper that, with constructing the operator $\mathcal{S}^{\mathrm{RtN}}_{\gamma}$, the cost in solving the problem is much smaller than the cost of solving subdomain problems [1], where the only issue being the storage capacity. Precisely, the cost of the MFB is associated with constructing the $\mathcal{S}^{\mathrm{RtN}}_{\gamma}$ directly depends on the number of degrees of freedom on the fractures. Thus, the MFB framework is favourable for 1) highly heterogeneous parts of the media where subdomain solves are affected by heterogeneities, 2) those fractures affected by strong non-linearities, and 3) lower permeable or blocking fractures where a coarse mortar space can be used without sacrificing accuracy. Otherwise, a robust preconditioner [7, 18] can be used in the Krylov method, as well as a coarse mortar space that is compensated by taking higher order mortars [9, 55].*

7 Numerical examples

In this section, we present several test cases to show how the schemes behave (1) for different values for numerical and physical parameters (2) with coarsening/refining mortar grids (3) on extensions to other governing equations. We subsequently study the value of $L_{\gamma,opt}$ in the MoLDD scheme and the relationship between $L_{\gamma,u}$ and $L_{\gamma,p}$ in the ItLDD scheme. The performance of schemes is measured in the overall number of iterations needed for each scheme to reach the stopping criteria. In the implementation of both schemes, we consider that the solution has converged if the relative error of the fracture solution is less than 10^{-5}, if the value at the previous iteration step is not zero. Otherwise we use the absolute error.

To keep the presentation simple, we consider domain and several parameters in common in all the examples in relation to the first test case in [43]. The domain $\Omega := (0, 2) \times (0, 1)$ is intersected with a fracture defined as $\gamma := \{x = 1\}$. On the boundaries of the rock matrix $\{x = 0\}$ and $\{x = 2\}$ we impose pressure boundary condition with values 0 and 1, respectively. We set zero flux boundary condition on the rest of $\partial\Omega$. The boundary of the fracture at the tips $\{y = 1\} \cap \partial\gamma$ and $\{y = 0\} \cap \partial\gamma$ inherits the pressure boundary conditions from the rock matrix. The examples are set on the time interval $I = (0, 1)$ with homogeneous pressure initial condition. As for the physical parameters, we take the permeability matrix for the bulk $\mathbf{K}_i = \mathbf{I}$, while the source terms f_i and f_γ are equal to zero.

Remark 7.1 (On the inner DD solver). *Previously we have mentioned the computational cost of both methods in the context of an iterative Krylov solver. However, the following examples are reduced to a one-dimensional fracture problem with negligible number of degrees of freedom (DOF) for such a solver to perform efficiently, Therefore, we only use the direct methods to solve the interface problem and rather demonstrate the robustness of our methods with regards to discretization, temporal, physical and L-scheme parameters. We still emphasize the need for a Krylov solver, such as GMRes, for large-scale problems.*

First, we consider the Forchheimer flow model where the non linear term is $\xi(\mathbf{u}_\gamma) = \beta|\mathbf{u}_\gamma|\mathbf{u}_\gamma$. The parameter β is a fluid dependent non-negative scalar known as the Forchheimer coefficient, and $|\cdot|$ denotes the Euclidean vector norm $|\mathbf{u}_\gamma|^2 = \mathbf{u}_\gamma \cdot \mathbf{u}_\gamma$. It is straightforward to see that ξ is a simply increasing function and satisfies condition (**A1**). For more details see [33, 38] and references therein.

$h\backslash n$	1	2	3	4	5
2^{-1}	17	12	11	10	9
2^{-3}	17	11	10	9	8
2^{-5}	17	11	10	9	8

τ																
2^{-2}	17				11				10				9			
2^{-3}	17		10		9		9		8		8		7		6	
2^{-4}	16	10	9	9	9	8	8	8	7	7	7	6	6	6	5	5

$h\backslash n$	1	2	3	4	5
2^{-1}	17	8	7	7	6
2^{-3}	17	9	8	7	7
2^{-5}	17	9	8	7	7

τ																
2^{-2}	17				9				8				7			
2^{-3}	17		10		9		8		8		7		6		6	
2^{-4}	16	10	9	9	9	8	8	8	7	7	7	6	6	6	5	5

Table 1: Results for the example of Subsection 7.1. Top two tables correspond to solving with the MoLDD scheme, while bottom two correspond to solving with the ItLDD scheme. On the left we report the number of iterations by varying the mesh size h for a fixed time step $\tau = 2^{-4}$, while on the right depending on the time step size τ for a fixed mesh size $h = 2^{-5}$.

7.1 Stability with respect to the user-given parameters

We first study the performances of MoLDD and ItLDD solvers by varying the time step τ, the mesh size h, and the L-scheme parameters $(L_{\gamma,u}, L_{\gamma,p})$. We let $\mathbf{K}_\gamma = 1$, $\alpha_\gamma = 10^4$ and $\beta = 1$ and according to the theoretical results, the L-scheme parameters are given by $L_{\gamma,u} \approx 1$ and $L_{\gamma,p} = 10^3$. Results in Table 1 report the number of iterations required by the two LDD solvers while varying the mesh size h and time step size τ. Each column of the tables represent results for a time step n.

Regardless of the choice of scheme, we can observe that the number of iterations is independent from the mesh size and slightly dependent on the time step size. The reason for the latter might be related to the fact that we consider the solution at previous iteration as the initial guess for the next iteration. Thus, by decreasing the time step size, the variation of the solution between steps varies less and so the number of iterations. Overall, the sequential ItLDD and the monolithic MoLDD solvers behave similarly; one can also see a slightly better results for the iterative solver in Table 1 (left). Note that any comparison of the two solvers does not make sense for the simple reason that the amounts of stabilization fixed by $L_{\gamma,p}$ and $L_{\gamma,u}$ are not yet optimal. Another explanation, may also be, the amount of stabilization in the monolithic solver MoLDD is set solely by $L_{\gamma,u}$, in contrast to the iterative solver ItLDD where two stabilization parameters $L_{\gamma,p}$ and $L_{\gamma,u}$ are used.

Finally, we recall that with the use of the multiscale flux basis, the computational costs of the two solvers is practically the same. In other words, any computational overhead of any of the solvers is free from any additional costs. The main cost is done offline using the multiscale flux basis which is mostly related to the number of mortar degrees on the fracture. As an example, the computational cost needed to draw the results in the last line (for $h = 2^{-5}$) of the two right tables (in Table 1) is approximately equal to 96 subdomain solves (Num. of DOF $*$ Num. of subdo. $+ 2 * N$), where two solves per time step are required to form the right-hand side in (3.1) (for MoLDD) and (3.4) (for ItLDD). Without MBF, the cost should be $\sum_{n=1}^{N} \sum_{k=1}^{N_{\text{Lin}}^n} * N_{\text{dd}}^k + 2 * N$, where N_{Lin}^n is the number of iterations of the L-scheme, and N_{dd}^k denotes the number of DD iterations (GMRes or any Krylov solver). Thus, if we assume a fixed N_{dd}^k along all the simulation, say $N_{\text{dd}}^k = 2$, this number will be at least 1012 subdomain solves, so that with MFB, we make a save of approximately 91% of the total subdomain solves.

In Figure 1, we plot the number of iterations with various realizations of the user-given $L_{\gamma,u}$ in MoLDD solver. We consider 100 values of $L_{\gamma,u}$, from 0 to 2.5 with uniform step 0.025. The other parameters are fixed as follows, $\beta = 1$, $h = 0.125$ and $\tau = 0.2$. The graph in this figure behaves very similarly to what is usually observed for the L-type schemes (a typical V-shape graph), highlighting a numerically optimal value $L_{\gamma,opt}$ between 0.5 and 1. By increasing $L_{\gamma,u}$, the number of iterations slowly increases, while they increase more drastically for small value of $L_{\gamma,u}$. This behavior is common for all time steps. We expect such a behavior when choosing $L_{\gamma,u}$ close to zero because it directly influences the contraction factor in (4.46). As a side result, we can see that the identified parameter $L_{\gamma,opt} \approx 1$ is close to the optimal one. On the other hand, we can observe the performance of the ItLDD solver with regards to changing parameters $L_{\gamma,u}$ and $L_{\gamma,p}$. We consider $L_{\gamma,u}$ taking 50 values uniformly distributed on the interval $(0, 2.5)$, while $L_{\gamma,p} = 10^x$, where x are 21 equidistant values on the interval $(2.2, 4.2)$ with step 0.1. As in the previous figure, we can observe

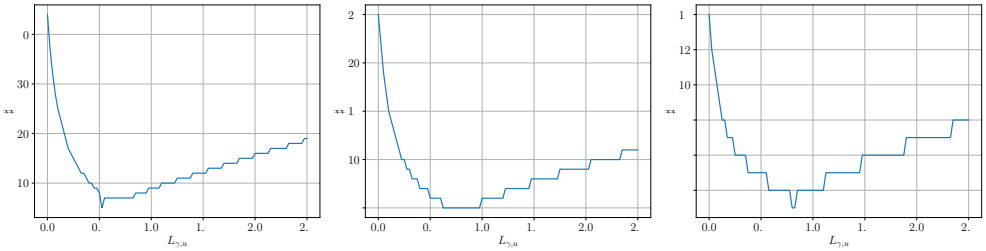


Figure 1: Results for the example of Subsection 7.1 using MoLDD scheme. We report the number of iterations ♯ for different values of $L_{\gamma,u}$. On the left for the first time step, in the centre for the third, and on the right for the last time step.

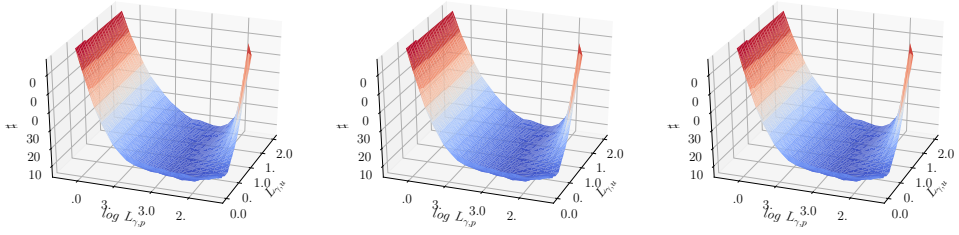


Figure 2: Results for the example of Subsection 7.1 using ItLDD scheme. We report the number of iterations ♯ for different values of $L_{\gamma,u}$ and $L_{\gamma,p}$. On the left for the first time step, in the centre for the third, and on the right for the last time step.

$\beta\backslash n$	1	2	3	4	5
0.1	17	9	8	7	7
1	17	9	8	7	7
100	9	8	7	6	5

$\beta\backslash n$	1	2	3	4	5
0.1	17	11	10	9	8
1	17	11	10	9	8
100	14	10	9	9	8

$\alpha_\gamma\backslash n$	1	2	3	4	5
10^2	17	9	8	7	7
10^4	17	9	8	7	7
10^6	17	9	8	7	7
10^8	17	9	8	7	7

$\alpha_\gamma\backslash n$	1	2	3	4	5
10^2	17	11	10	9	9
10^4	17	11	10	9	8
10^6	17	11	10	9	8
10^8	17	11	10	9	8

Table 2: Results for the example of Subsection 7.2 reporting the number of iterations by varying the parameter β (top) and by varying α_γ (bottom). Left tables correspond to solving with the MoLDD solver, while the right ones correspond to solving with the ItLDD solver.

on the surface plots that there is a global minimum that determines the optimal choice for $L_{\gamma,u}$ and $L_{\gamma,p}$. For example, the minimum number of iterations for this flow model is 5 for $L_{\gamma,u}$ between 0.59 and 1.1 and $\log(L_{\gamma,p})$ between 2.8 and 3, in all time steps. Similar to the monolithic approach, the number of iterations required by the ItLDD solver increases when the L-scheme parameters assume low values. Particularly, the scheme diverges when $L_{\gamma,p}$ is less or equal to 10^2. In the analysis of the scheme we require that $L_{\gamma,p} \geq \alpha_\gamma$, but the lower values also allow a good convergence behaviour concluding that the theoretical lower bound is possibly too strict, but it certainly exists. Therefore, in practice, we can slightly relax the bounds on the L-scheme parameters to still obtain good performance of the solver. It is also relevant to mention that the normal permeability constant $\alpha_\gamma = 10^4$ is sufficiently large to apply the limit case results in Lemma 5.3.

Crucially, we want to mention that the computational cost of the realizations needed to draw Figure 1 and 2, is exactly equal to only one realization with fixed $(L_{\gamma,u}, L_{\gamma,p})$, permitting easier calculation of these parameters, and confirming the utility of the MFB on fixing the total cost and avoiding any computational overhead if these parameters are not optimal.

7.2 Robustness with respect to the physical parameters

In this set of test examples, we want to show the robustness of the algorithms with respect to α_γ and β. Note that α_γ controls the strength of the fracture-matrix coupling, while β controls the strength of the non-linearity. We fix the mesh size $h = 0.125$ and the time step $\tau = 2^{-3}$.

In Table 2 (top), we study the dependency of the number of iterations on the Forchheimer coefficient β. The LDD solvers show a weak dependency of the number of iterations on the values of β, giving slightly better results for larger values. Overall, the monolithic solver MoLDD performs slightly better then the iterative one ItLDD. Bear in mind that changing β, directly influences $L_{\gamma,u}$. This shows that this parameter should be optimized in accordance to the given value of β. Again, we suggest that the decrease in number of iterations over time steps may be due to using the previous iteration solution as the initial guess in the subsequent iteration. Clearly, we can conclude that the two solvers remain robust when strengthening the non-linearity effects. Moreover, all the simulations in Table 2 (top) are run with a fixed computational cost. The number of subdomain solves needed to carry out the simulations in Table 2 (top) is equal to 32 subdomain solves (Num. of DOF $*$ Num. of subdo. $+ 2 * N$). Thus, strengthening or changing the non-linearity effects, for which maybe the number of iterations increases if the amount of stabilization via $L_{\gamma,u}$ and/or $L_{\gamma,p}$ is not carefully set, has no practical effects on the total computational costs. This gain in the computational resources which is in *conformity* with the spirit of *reduced basis* confirms that the MFB is an essential tool in the implementation of our LDD solvers.

Turning now to the effect of the fracture-matrix coupling on the two LDD solvers, we plot in Table 2 (bottom) the dependency of the number of iterations on the Robin parameter α_γ. Clearly, the number of iterations remains stable when strengthening or weakening matrix-fracture coupling, confirming and concluding the robustness of both schemes with respect to α_γ. Example of solution is reported in Figure 3. For the computational cost of the results in Table 2 (bottom), any change of α_γ requires re-computing the

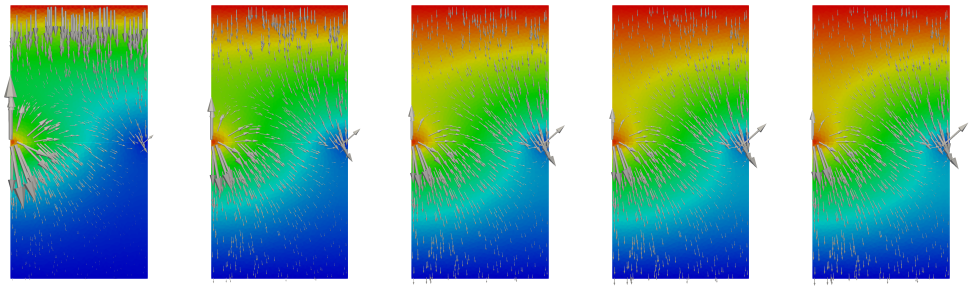


Figure 3: $p \in [0,1]$ and $\mathbf{u}$ for the example in Subsection 7.2 with $\beta = 10^2$ and $\alpha_\gamma = 10^4$.

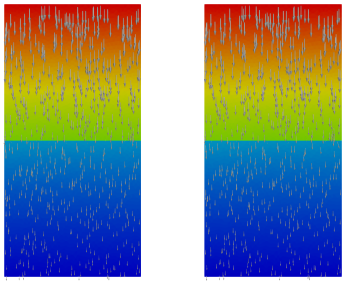


Figure 4: $p \in [0,1]$ and $\mathbf{u}$ for the example in Subsection 7.3 with $\mathbf{K}_\gamma = 10^{-4}\mathbf{I}$, $\beta = 1$ and $\alpha_\gamma = 1$. With fine (left) and coarse (right) mortar grids.

multiscale flux basis. However, this cost remains fixed when running and comparing the two LDD solvers for a fixed α_γ.

7.3 Flexibility of coarsening/refining the mortar grids

In this set of simulations, we consider the case of weak inter-dimensional coupling by fixing $\alpha_\gamma = 1$, with a low permeable fracture with $\mathbf{K}_\gamma = 10^{-4}\mathbf{I}$. We fix the following parameters: $h = 2^{-5}$ (on the matrix), $L_{\gamma,u} = 1$, $L_{\gamma,p} = 2 \cdot 10^2$ and $\beta = 1$. We allow for a coarse scale of the mortar grids on the fracture; $h_\gamma = 2^{-3}$, $h_\gamma = 2^{-4}$, $h_\gamma = 2^{-5}$, where the last choice corresponds to matching grids on the fracture. In Table 3, we plot the resulting number of iterations required by each LDD solver. Particularly, we can see that the sequential ItLDD solver in the matching grids has more difficulty to converge, so the effectiveness of the MFB is more pronounced. The monolithic solver MoLDD seems to be more robust with refining the mortar grids. Here, the computational cost of the construction of the inter-dimensional operator benefits from fewer mortar degrees on the fracture. Example of a solution is depicted in Figure 4, where we can see that conforming and non-conforming gridddig (with $h_\gamma = 2^{-3}$) on the fracture give indistinguishable results.

7.4 Extension to other flow models: the Cross model

The aim of this test case is to show that our LDD solvers can be applied to more general flow models. On the fracture, we assume the Cross flow model to relate p_γ and $\mathbf{u}_\gamma$. We have the non-linear term given by

$$\xi(\mathbf{u}_\gamma) = \frac{(\omega_0 - \omega_\infty)\mathbf{u}_\gamma}{1 + K_\gamma |\mathbf{u}_\gamma|^{2-r}}.$$

♯cells\n	1	2	3	4	5
8	3	3	3	3	2
16	3	3	3	3	2
64	3	3	3	3	2

♯cells\n	1	2	3	4	5
8	11	10	10	9	9
16	11	10	10	9	9
64	18	15	15	15	14

Table 3: Results for the example of Subsection 7.3 reporting the number of iterations for conforming and non-conforming (coarse scale) grids on the fracture. Left table corresponds to solving with the MoLDD scheme, while the right one corresponds to solving with the ItLDD scheme.

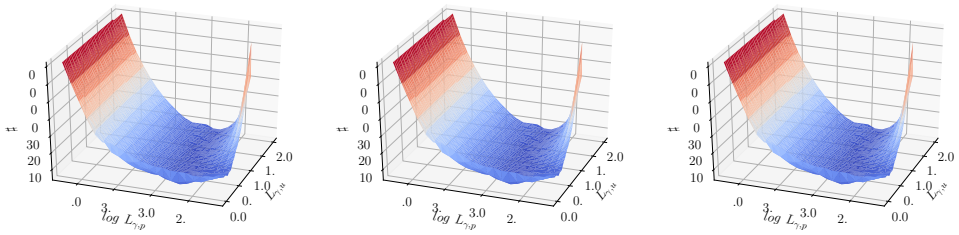


Figure 5: Results for the example of Subsection 7.4 using ItLDD scheme. We report the number of iterations ♯ for different values of $L_{\gamma,u}$ and $L_{\gamma,p}$. On the left for the first time step, in the centre for the third, and on the right for the last time step.

The parameters $0 \leq \omega_\infty < \omega_0$, ζ and r are positive scalars related to the rheology of the considered liquid. In (1.2a), $\mathbf{K}_\gamma$ is now replaced by ω_∞. We let $\omega := \omega_\infty - \omega_0$ and set $\omega_0 = 2$, $\omega_\infty = 1$, $\zeta = 1$, and $r = 1.5$. It is easy to verify that ξ satisfies the assumption (**A1**) . For more details see [24, 25] and the references therein.

We choose the iterative solver ItLDD and re-compute the simulations of Subsection 7.1 and 7.2 for the Cross flow model. We set then $L_{\gamma,u} = L_\xi/2 = 0.5$ and $L_{\gamma,p} = \alpha_\gamma = 10^3$ as derived from the theory. The results (not shown) demonstrate first the stability of the ItLDD solver with respect to the parameters h and τ. Crucially, all the simulations in this example do not require additional computational cost (except fracture solves), as we use the same MFB inherited from the Forchheimer model. We set $h = 2^{-5}$ with slightly coarse grids on the fracture $h_\gamma = 2^{-4}$ and $\tau = 2^{-4}$.

In Figure 5, we show the results for the ItLDD solver on a set of realizations of $(L_{\gamma,u}, L_{\gamma,p})$. The results do not differ greatly comparing to the case of Forchheimer's flow model. The convexity of the surface plots in all time steps is clear giving away an optimal combination of values for $L_{\gamma,u}$ and $L_{\gamma,p}$. For example, we can find minimum of 5 iterations for $L_{\gamma,u}$ between approximately 0.73 and 1.57, and $L_{\gamma,p}$ between $10^{2.8}$ and 10^3. Note that the parameters prescribed by the theoretical results, $L_{\gamma,u} = 0.5$ and $L_{\gamma,p} = 10^3$, form a good candidate in this simulation. Finally, we mention that we can use an optimization process, as detailed in [50], in order to get the optimal values. Precisely, the fact that the choice of the stabilization parameters is independent of the mesh size, one can then run the LDD solver on a coarse spatial mesh and one time step, and study the stabilization parameters in specific intervals centred around the theoretical values. The parameters that give the lowest number of iterations are then used for the real computations. This "brute-force" optimization is simple to do in practice when using the MFB.

In Table 4, we consider to test the dependency of the number of iterations on the rheology parameters of

$\omega\backslash n$	1	2	3	4	5
0.1	15	11	10	9	8
1	10	9	8	8	8
2.5	30	19	16	14	12

$\zeta\backslash n$	1	2	3	4	5
1	10	9	8	8	7
10	11	9	9	8	7
100	16	11	10	9	8

$r\backslash n$	1	2	3	4	5
1	10	9	9	8	7
1.5	10	9	8	8	7
4.5	17	11	10	9	8

Table 4: Results for the example of Subsection 7.4. On the left the number of iterations by varying the values of ω. In the center when ζ changes, while on the right for different values of r.

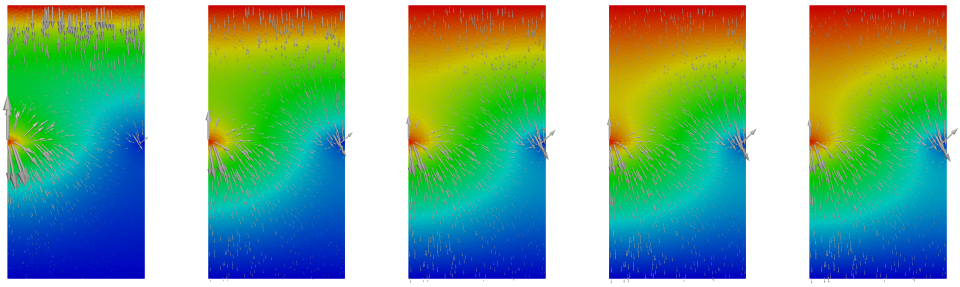


Figure 6: $p \in [0, 1]$ and $\mathbf{u}$ for the example in Subsection 7.4 with $\omega = 1$, $\zeta = 1$, and $r = 1.5$.

the flow model. We provide results of several tests on ω, ζ, and r. While testing for one of the parameters, the other two are fixed to either $\omega = 1$, $\zeta = 1$ or $r = 1.5$. We can observe that ω strongly influences the performance of both methods making it difficult to converge when ω gets larger, that is, when the non-linearity is stronger. For larger values of ω the number of iterations increases drastically, suggesting the necessity to adjust the L-scheme parameters as well as to use the MFB. The number of iterations was less dependent of the parameter ζ. This parameter itself contributes less to the strength of the non-linearity in comparison to ω, and, thus, influencing less the performance of the solver. Finally, we can again notice a moderate dependency of number of iterations on parameter r. This is especially shown when $r > 2$ and the exponent on the vector norm of $\mathbf{u}_\gamma$ becomes negative. Thus, the non-linear flow function ξ is exponential in the values of $\mathbf{u}_\gamma$ and accounts for the very fast flow in the fractures. We finally recall that the robustness study drawn in Table 4 has the cost of one realization with fixed-parameters, confirming the role of the MFB in our solvers. For the robustness of LDD solvers with respect to the matrix-fracture coupling effects induced by the parameter α_γ, we have seen that both solvers are robust when strengthening or weakening the coupling effects (results not shown). Example of solution is reported in Figure 6.

8 Conclusions

In this study, we have presented two new strategies to solve a compressible single-phase flow problem in a porous medium with a fracture. In the porous medium, we have considered the classical Darcy relation between the velocity and the pressure while, in the fracture, a general non-linear law. We employ the L-scheme to handle the non-linearity term, but also to treat the inter-dimensional coupling in the second proposed algorithm. To further achieve computational speed-up, the linear Robin-to-Neumann co-dimensional map is constructed in an offline phase resulting in a problem reduced only to the fracture interface. This approach allows to change the fracture parameters, or the fracture flow model in general, without the need to recompute the problem associated with the rock matrix. We have shown the existence of optimal values for the L-scheme parameters, which are validated through several numerical tests. Future developments can be explored towards domain decomposition in time, where fast and slow fractures are solved asynchronously.

Acknowledgements

We acknowledge the financial support from the Research Council of Norway for the TheMSES project (project no. 250223) and the ANIGMA project (project no. 244129/E20) through the ENERGIX program.

References

[1] E. Ahmed, A. Fumagalli, and A. Budiša, *A multiscale flux basis for mortar mixed discretizations of reduced darcy-forchheimer fracture models*, Computer Methods in Applied Mechanics and Engineering,

(2019), https://doi.org/https://doi.org/10.1016/j.cma.2019.05.034.

[2] E. Ahmed, S. A. Hassan, C. Japhet, M. Kern, and M. Vohralík, *A posteriori error estimates and stopping criteria for space-time domain decomposition for two-phase flow between different rock types*. working paper or preprint, June 2017, https://hal.inria.fr/hal-01540956.

[3] E. Ahmed, J. Jaffré, and J. E. Roberts, *A reduced fracture model for two-phase flow with different rock types*, Math. Comput. Simulation, 137 (2017), pp. 49–70, https://doi.org/10.1016/j.matcom.2016.10.005.

[4] 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*, arXiv:1901.01206 [math.NA], (2019), https://arxiv.org/abs/1901.01206.

[5] 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), vol. 295 of Contemp. Math., Amer. Math. Soc., Providence, RI, 2002, pp. 13–24, https://doi.org/10.1090/conm/295/04999.

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

[7] P. F. Antonietti, J. De Ponti, L. Formaggia, and A. Scotti, *Preconditioning techniques for the numerical solution of flow in fractured porous media*, MOX Report 17, Politecnico di Milano, 2019.

[8] P. F. Antonietti, L. Formaggia, A. Scotti, M. Verani, and N. Verzott, *Mimetic finite difference approximation of flows in fractured porous media*, ESAIM Math. Model. Numer. Anal., 50 (2016), pp. 809–832, https://doi.org/10.1051/m2an/2015087.

[9] T. Arbogast, G. Pencheva, M. F. Wheeler, and I. Yotov, *A multiscale mortar mixed finite element method*, Multiscale Model. Simul., 6 (2007), pp. 319–346, https://doi.org/10.1137/060662587.

[10] M. Arshad, E.-J. Park, and D.-w. Shin, *Analysis of multiscale mortar mixed approximation of nonlinear elliptic equations*, Comput. Math. Appl., 75 (2018), pp. 401–418, https://doi.org/10.1016/j.camwa.2017.09.031.

[11] A. Benaceur, V. Ehrlacher, A. Ern, and S. Meunier, *A progressive reduced basis/empirical interpolation method for nonlinear parabolic problems*, SIAM J. Sci. Comput., 40 (2018), pp. A2930–A2955, https://doi.org/10.1137/17M1149638.

[12] H. Berninger, S. Loisel, and O. Sander, *The 2-Lagrange multiplier method applied to nonlinear transmission problems for the Richards equation in heterogeneous soil with cross points*, SIAM J. Sci. Comput., 36 (2014), pp. A2166–A2198, https://doi.org/10.1137/120901064, https://doi.org/10.1137/120901064.

[13] I. Berre, F. Doster, and E. Keilegavlen, *Flow in fractured porous media: A review of conceptual models and discretization approaches*, Transport in Porous Media, (2018), https://doi.org/10.1007/s11242-018-1171-6.

[14] W. M. Boon, J. M. Nordbotten, and I. Yotov, *Robust discretization of flow in fractured porous media*, SIAM J. Numer. Anal., 56 (2018), pp. 2203–2233, https://doi.org/10.1137/17M1139102.

[15] M. Borregales, F. A. Radu, K. Kumar, and J. M. Nordbotten, *Robust iterative schemes for non-linear poromechanics*, Comput. Geosci., 22 (2018), pp. 1021–1038, https://doi.org/10.1007/s10596-018-9736-6.

[16] K. Brenner, J. Hennicker, R. Masson, and P. Samier, *Hybrid-dimensional modelling of two-phase flow through fractured porous media with enhanced matrix fracture transmission conditions*, J. Comput. Phys., 357 (2018), pp. 100–124, https://doi.org/10.1016/j.jcp.2017.12.003.

[17] M. K. Brun, E. Ahmed, I. Berre, J. M. Nordbotten, and F. A. Radu, *Monolithic and splitting based solution schemes for fully coupled quasi-static thermo-poroelasticity with nonlinear convective transport*, arXiv:1902.05783 [math.NA], (2019), https://arxiv.org/abs/1902.05783.

[18] A. Budiša and X. Hu, *Block preconditioners for mixed-dimensional discretization of flow in fractured porous media*, arXiv:1905.13513 [math.NA], (2019), https://arxiv.org/abs/1905.13513.

[19] S. Chen and H. Rui, *A two-grid decoupled algorithm for fracture models*, Comput. Math. Appl., 76 (2018), pp. 1161–1173, https://doi.org/10.1016/j.camwa.2018.06.005.

[20] C. D'Angelo and A. Scotti, *A mixed finite element method for Darcy flow in fractured porous media with non-matching grids*, Mathematical Modelling and Numerical Analysis, 46 (2012), pp. 465–489, https://doi.org/10.1051/m2an/2011148.

[21] M. Del Pra, A. Fumagalli, and A. Scotti, *Well posedness of fully coupled fracture/bulk Darcy flow with XFEM*, SIAM J. Numer. Anal., 55 (2017), pp. 785–811, https://doi.org/10.1137/15M1022574.

[22] V. Dolean, P. Jolivet, and F. Nataf, *An introduction to domain decomposition methods*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015, pp. x+238, https://doi.org/10.1137/1.9781611974065.ch1. Algorithms, theory, and parallel implementation.

[23] A. Ern and J.-L. Guermond, *Evaluation of the condition number in linear systems arising in finite element approximations*, M2AN Math. Model. Numer. Anal., 40 (2006), pp. 29–48, https://doi.org/10.1051/m2an:2006006.

[24] V. J. Ervin, E. W. Jenkins, and S. Sun, *Coupling nonlinear Stokes and Darcy flow using mortar finite elements*, Appl. Numer. Math., 61 (2011), pp. 1198–1222, https://doi.org/10.1016/j.apnum.2011.08.002.

[25] V. J. Ervin, H. Lee, and A. J. Salgado, *Generalized Newtonian fluid flow through a porous medium*, J. Math. Anal. Appl., 433 (2016), pp. 603–621, https://doi.org/10.1016/j.jmaa.2015.07.054.

[26] 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 (2014), pp. 1089–1116, https://doi.org/10.1051/m2an/2013132.

[27] N. Frih, V. Martin, J. E. Roberts, and A. Saâda, *Modeling fractures as interfaces with nonmatching grids*, Computational Geosciences, 16 (2012), pp. 1043–1060, https://doi.org/10.1007/s10596-012-9302-6.

[28] N. Frih, J. E. Roberts, and A. Saada, *Modeling fractures as interfaces: a model for Forchheimer fractures*, Comput. Geosci., 12 (2008), pp. 91–104, https://doi.org/10.1007/s10596-007-9062-x.

[29] A. Fumagalli and A. Scotti, *A reduced model for flow and transport in fractured porous media with non-matching grids*, in Numerical Mathematics and Advanced Applications 2011, A. Cangiani, R. L. Davidchack, E. Georgoulis, A. N. Gorban, J. Levesley, and M. V. Tretyakov, eds., Berlin, Heidelberg, 2013, Springer Berlin Heidelberg, pp. 499–507, https://doi.org/10.1007/978-3-642-33134-3_53.

[30] B. Ganis, G. Pencheva, M. F. Wheeler, T. Wildey, and I. Yotov, *A frozen Jacobian multiscale mortar preconditioner for nonlinear interface operators*, Multiscale Model. Simul., 10 (2012), pp. 853–873, https://doi.org/10.1137/110826643.

[31] B. Ganis, D. Vassilev, C. Wang, and I. Yotov, *A multiscale flux basis for mortar mixed discretizations of Stokes-Darcy flows*, Comput. Methods Appl. Mech. Engrg., 313 (2017), pp. 259–278, https://doi.org/10.1016/j.cma.2016.09.037.

[32] B. Ganis and I. Yotov, *Implementation of a mortar mixed finite element method using a multiscale flux basis*, Comput. Methods Appl. Mech. Engrg., 198 (2009), pp. 3989–3998, https://doi.org/10.1016/j.cma.2009.09.009.

[33] V. Girault and M. F. Wheeler, *Numerical discretization of a Darcy-Forchheimer model*, Numer. Math., 110 (2008), pp. 161–198, https://doi.org/10.1007/s00211-008-0157-7.

[34] H. Hægland, A. Assteerawatt, H. Dahle, G. Eigestad, and R. Helmig, *Comparison of cell- and vertex-centered discretization methods for flow in a two-dimensional discrete-fracture–matrix system*, Advances in Water Resources, 32 (2009), pp. 1740–1755, https://doi.org/10.1016/j.advwatres.2009.09.006.

[35] T.-T.-P. Hoang, C. Japhet, M. Kern, and J. E. Roberts, *Space-time domain decomposition for reduced fracture models in mixed formulation*, SIAM J. Numer. Anal., 54 (2016), pp. 288–316, https://doi.org/10.1137/15M1009651.

[36] E. Keilegavlen, A. Fumagalli, R. Berge, I. Stefansson, and I. Berre, *Porepy: An open source simulation tool for flow and transport in deformable fractured rocks*, tech. report, arXiv:1712.00460 [cs.CE], 2017, https://arxiv.org/abs/1712.00460.

[37] M.-Y. Kim and E.-J. Park, *Fully discrete mixed finite element approximations for non-Darcy flows in porous media*, Comput. Math. Appl., 38 (1999), pp. 113–129, https://doi.org/10.1016/S0898-1221(99)00291-6.

[38] P. Knabner and J. E. Roberts, *Mathematical analysis of a discrete fracture model coupling Darcy flow in the matrix with Darcy-Forchheimer flow in the fracture*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 1451–1472, https://doi.org/10.1051/m2an/2014003.

[39] M. Lesinigo, C. D'Angelo, and A. Quarteroni, *A multiscale Darcy-Brinkman model for fluid flow in fractured porous media*, Numer. Math., 117 (2011), pp. 717–752, https://doi.org/10.1007/s00211-010-0343-2.

[40] F. List, K. Kumar, I. S. Pop, and F. A. Radu, *Upscaling of unsaturated flow in fractured porous media*, arXiv preprint arXiv:1807.05993, (2018).

[41] F. List and F. A. Radu, *A study on iterative methods for solving Richards' equation*, Comput. Geosci., 20 (2016), pp. 341–353, https://doi.org/10.1007/s10596-016-9566-3.

[42] W. Liu and Z. Sun, *A block-centered finite difference method for reduced fracture model in Karst aquifer system*, Comput. Math. Appl., 74 (2017), pp. 1455–1470, https://doi.org/10.1016/j.camwa.2017.06.028.

[43] V. Martin, J. Jaffré, and J. E. Roberts, *Modeling Fractures and Barriers as Interfaces for Flow in Porous Media*, SIAM J. Sci. Comput., 26 (2005), pp. 1667–1691, https://doi.org/10.1137/S1064827503429363.

[44] K. Mitra and I. Pop, *A modified l-scheme to solve nonlinear diffusion problems*, Computers & Mathematics with Applications, 77 (2019), pp. 1722 – 1738, https://doi.org/https://doi.org/10.1016/j.camwa.2018.09.042. 7th International Conference on Advanced Computational Methods in Engineering (ACOMEN 2017).

[45] J. M. Nordbotten and P. E. Bjø rstad, *On the relationship between the multiscale finite-volume method and domain decomposition preconditioners*, Comput. Geosci., 12 (2008), pp. 367–376, https://doi.org/10.1007/s10596-007-9066-6, https://doi.org/10.1007/s10596-007-9066-6.

[46] I. S. Pop, F. Radu, and P. Knabner, *Mixed finite elements for the Richards' equation: linearization procedure*, J. Comput. Appl. Math., 168 (2004), pp. 365–373, https://doi.org/10.1016/j.cam.2003.04.008.

[47] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Numerical Mathematics and Scientific Computation, The Clarendon Press, Oxford University Press, New York, 1999. Oxford Science Publications.

[48] N. Schwenck, B. Flemisch, R. Helmig, and B. I. Wohlmuth, *Dimensionally reduced flow models in fractured porous media: crossings and boundaries*, Comput. Geosci., 19 (2015), pp. 1219–1230, https://doi.org/10.1007/s10596-015-9536-1.

[49] D. Seus, K. Mitra, I. S. Pop, F. A. Radu, and C. Rohde, *A linear domain decomposition method for partially saturated flow in porous media*, Comput. Methods Appl. Mech. Engrg., 333 (2018), pp. 331–355, https://doi.org/10.1016/j.cma.2018.01.029.

[50] E. Storvik, J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu, *On the optimization of the fixed-stress splitting for biot's equations*, International Journal for Numerical Methods in Engineering, (2019), https://doi.org/10.1002/nme.6130.

[51] S. G. Thomas and M. F. Wheeler, *Enhanced velocity mixed finite element methods for modeling coupled flow and transport on non-matching multiblock grids*, Comput. Geosci., 15 (2011), pp. 605–625, https://doi.org/10.1007/s10596-011-9227-5.

[52] K. Urban and B. Wieland, *Affine decompositions of parametric stochastic processes for application within reduced basis methods*, IFAC Proceedings Volumes, 45 (2012), pp. 716–721, https://doi.org/10.3182/20120215-3-AT-3016.00127. 7th Vienna International Conference on Mathematical Modelling.

[53] A. Veneziani, *Certified reduced basis methods for parametrized partial differential equations*, SIAM Rev., 59 (2017), pp. 219–221.

[54] F. Xing, R. Masson, and S. Lopez, *Parallel vertex approximate gradient discretization of hybrid dimensional Darcy flow and transport in discrete fracture networks*, Comput. Geosci., 21 (2017), pp. 595–617, https://doi.org/10.1007/s10596-016-9606-z.

[55] Y. Yang, E. T. Chung, and S. Fu, *An enriched multiscale mortar space for high contrast flow problems*, Commun. Comput. Phys., 23 (2018), pp. 476–499, https://doi.org/10.4208/cicp.OA-2016-0147.

Graphic design: Communication Division, UiB / Print: Skipnes Kommunikasjon AS

uib.no

ISBN: 9788230869925 (print)
9788230869000 (PDF)