THESIS FOR THE DEGREE OF LICENTIATE OF PHILOSOPHY

Numerical approximation of mixed dimensional partial differential equations

Malin Mosquera

CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
Gothenburg, Sweden 2023

Numerical approximation of mixed dimensional partial differential equations
Malin Mosquera
Gothenburg 2023

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Sweden
Telephone +46 (0)31 772 1000

Typeset with LaTeX
Printed by Stema Specialtryck AB, Borås, Sweden, 2023

Numerical approximation of mixed dimensional partial differential equations

Malin Mosquera

Division of Applied Mathematics and Mathematical Statistics
Department of Mathematical Sciences
Chalmers University of Technology and University of Gothenburg

Abstract

In this thesis, we explore numerical approximation of elliptic partial differential equations posed on domains with a high number of interfaces running through. The finite element method is a well-studied numerical method to solve partial differential equations, but requires alterations to handle interfaces. This can result in either unfitted or fitted methods. In this thesis, our focus lies on fitted methods.

From finite element methods, one obtains large linear systems that need to be solved, either directly or via an iterative method. We discuss an iterative method, which converges faster when using a preconditioner on the linear system. The preconditioner that we utilise is based on domain decomposition.

In Paper I, we consider this kind of partial differential equation posed on a domain with interfaces, and show existence and uniqueness of a solution. We state and prove a regularity result in two dimensions. Further, we propose a fitted finite element approximation and derive error estimates to show convergence. We also present a preconditioner based on domain decomposition that we use together with an iterative method, and analyse the convergence. Finally, we perform numerical experiments that confirm the theoretical findings.

Keywords: Finite element method, mixed dimensional partial differential equation, a priori error analysis, subspace decomposition, preconditioner.

List of appended papers

This thesis is based on the work represented by the following paper:

I. Hellman, F., Målqvist, A., **Mosquera, M.** (2023). Well-Posedness and Finite Element Approximation of Mixed Dimensional Partial Differential Equations. Submitted, doi: 10.48550/arXiv.2212.14387.

Author contributions

I. Worked on the idea behind the research in collaboration with the co-authors. Derived and proved theoretical results, as well as drafted and edited the paper in collaboration with the co-authors. Implemented and performed the numerical examples.

Acknowledgements

First and foremost, I would like to thank my supervisor Axel Målqvist for his engaged guidance and continuous help throughout the work of this thesis. To have frequent meetings this whole time to discuss the research has been a major benefit and is something I truly value. I would also like to thank my co-author Fredrik Hellman, whose early contributions to the article helped very much to get me started with the research.

Further, I wish to express my gratitude towards all my current and former fellow PhD students at the department, with whom I always have interesting conversations and discussions. You all contribute to creating a friendly working environment.

Finally, I would like to thank my husband, who is always gladly taking responsibility for our family when I need to focus on work. Your love and support are invaluable to me.

Contents

1 Introduction

Partial differential equations (PDEs) constitute an important field in applied mathematics, with many problem formulations arising from physics and engineering. A few examples of areas that give rise to PDEs are aerodynamics, fluid dynamics and thermodynamics, although this list can be considerably extended.

In this thesis, our interest lies in domains with interfaces running through. The bulk and the interface areas are governed by different PDEs, connected via coupling conditions. This setting is a so-called mixed-dimensional model, where the governing equations are defined in different dimensions. An application of this can for instance be permeable rocks with cracks, which in reality are 3-dimensional structures, but may be modelled as 2-dimensional interfaces instead.

PDEs are in general very hard, often impossible, to solve analytically. This raises a need for numerical methods that are able to solve them approximately. Many such methods have been developed over the years, for instance the widely used finite difference methods and finite volume methods. The most prominent method though, and the one that will be subject to our interest in this thesis, is the finite element method (FEM). This method transforms infinite-dimensional PDE problems to finite-dimensional ones, resulting in linear systems that need to be solved.

In its basic form, FEM is defined on simple domains. In order to tackle domains with interfaces and other structures, several variations of the finite element method have been developed. In this thesis, we will focus on the so-called fitted finite element method, which means that the interfaces are taken into consideration when creating the discretisation of the domain.

To this thesis, one paper is appended, where we present a PDE problem on a domain with a large number of interfaces, a fitted finite element method to

discretise the problem, and an iterative solver to solve the linear system that arises from the fitted FEM. We show

- existence and uniqueness of a solution to the presented problem, along with regularity results in two dimensions,
- an a priori error estimate of the finite element solutions, and
- rapid convergence of the presented iterative solver.

The outline of the thesis is as follows.

In **Chapter 2**, we give some background information that is needed before one delves into the areas of the actual research presented in this thesis. This includes an introduction into elliptic partial differential equations, along with derivation of the weak formulation and some theoretical results. Further, we present the finite element method and an estimate of the error that comes with using it. Lastly, we present an iterative method that enables us to solve large linear systems.

In **Chapter 3**, we introduce the setting with interfaces that is our actual research interest. We first look at how the problem can be formulated, derive the weak formulation and present some theoretical results. We continue by presenting an altered version of FEM, the fitted finite element method, that can deal with our new setting. Lastly, we present an elevated version of the iterative method along with a preconditioner of the linear system that suits our presented problem.

In **Chapter 4**, we summarise the appended paper and describe the planned future work.

2 Background

In this chapter, we go through some mathematical background that is needed in order to understand the article and the ongoing work. In Section 2.1, we introduce an elliptic PDE problem together with some analysis of it. In order to solve partial differential equations, one generally needs a numerical method that provides an approximate solution. The finite element method is such a method and constitutes the foundation of both the article, ongoing work and other concepts that need to be explained before the article. Thus, it is natural to introduce it in Section 2.2.

With the help of FEM, one can discretise an infinite-dimensional problem by dividing it into smaller parts, finite elements, and thus simplifying the problem into being finite-dimensional. There are many books on the subject, see for example introductory books such as (Asadzadeh, 2020; Larsson and Thomée, 2009), or somewhat more advanced books like (Brenner and Scott, 2002). In this chapter, we describe the general idea behind the method, focusing on elliptic problems with homogeneous Dirichlet boundary conditions in order to demonstrate the method in a rather easy way. We will also go through some fundamental concepts. Some analysis of the method is done, in order to see that it produces good results. We focus our attention on error analysis, examining how close the obtained approximation is to the true solution of the problem.

The finite element method generates a, usually large, linear system that needs to be solved in order to obtain the solution. A linear system can either be solved via a direct method, e.g. Gaussian elimination, that provides the exact result up to machine precision, or an iterative method that provides an arbitrarily good approximation. With the large linear systems that we handle in this thesis, an iterative method is the viable option, and we thus present such a method, the *conjugate gradient (CG) method*, in the end of the chapter. The conjugate gradient method is also a well-studied method in applied mathematics, first proposed in (Hestenes and Stiefel, 1952). There are several books that present the CG method, see for example (Greenbaum, 1997; Demmel, 1997).

2.1 An elliptic model problem

In this section, we introduce an elliptic PDE problem with homogeneous Dirichlet boundary conditions. We present the strong form of the problem, derive the weak form and comment on the existence and uniqueness of a solution. Lastly, we examine the regularity of the solution.

2.1.1 Problem formulation

We consider an open and connected domain Ω in $\mathbb{R}^d$, where $d = 1$, 2 or 3, with Lipschitz boundary, on which we have the following elliptic partial differential equation equipped with homogeneous Dirichlet boundary conditions:

$$-\nabla \cdot (A_0 \nabla u) = f, \qquad \text{in } \Omega, \tag{2.1a}$$

$$u = 0, \qquad \text{on } \Omega, \tag{2.1b}$$

where the diffusion coefficient A_0 and the source term f are both functions defined on Ω. Henceforth, we will assume that the boundary $\partial\Omega$ is polytopal, since it is then easier to define the finite element method.

2.1.2 Variational formulation, existence and uniqueness

In order to express equation 2.1 weakly, we need to have a clear concept of certain norms and vector spaces. We will heavily use the Hilbert space $H_0^1(\Omega)$ equipped with the H^1-norm, defined by $\|\cdot\|^2_{H^1(\Omega)} = \|\cdot\|^2_{L^2(\Omega)} + \|\nabla\cdot\|^2_{L^2(\Omega)}$. The $L^2(\omega)$ scalar product will frequently be denoted $(\cdot,\cdot)_\omega$, and in the case when $\omega = \Omega$, we simply write $(\cdot,\cdot)$.

In the following, we will assume that V is a Hilbert space with norm $\|\cdot\|_V$ and scalar product $(\cdot,\cdot)_V$, typically $V = H_0^1(\Omega)$ equipped with the standard H^1-norm.

In equation (2.1), we are looking for a solution $u \in V$. The vector space V is then called the *trial space* of the problem.

In order to obtain the *variational formulation*, or *weak formulation*, of a PDE problem, one multiplies the differential equation by a *test function* from a *test space* V and integrates. The test space can differ from the trial space, but often they are the same, which they will be in this thesis. Here, in order to obtain

the weak form corresponding to (2.1), we multiply (2.1a) by a test function $v \in H_0^1(\Omega)$, and integrate to obtain

$$-\int_\Omega \nabla \cdot (A_0 \nabla u)\, v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}x.$$

Using Green's formula, we move a weak derivative to the test function:

$$\int_\Omega A_0 \nabla u \cdot \nabla v \,\mathrm{d}x = \int_\Omega f v \,\mathrm{d}x.$$

Now the variational formulation reads: find $u \in H_0^1(\Omega)$ such that

$$a(u, v) = L(v), \qquad \text{for all } v \in H_0^1(\Omega), \tag{2.2}$$

where $a(u, v) = (A_0 \nabla u, \nabla v)$ is a bilinear form and $L(v) = (f, v)$ is a linear functional.

Noteworthy is that the Lax-Milgram theorem states that if

- a is *continuous/bounded*, i.e. there is a positive constant $\overline{\alpha}$ such that $\big|a(u, v)\big| \leq \overline{\alpha} \|u\|_V \|v\|_V$ for all $u, v \in V$,
- a is *coercive*, i.e. there is a constant $\underline{\alpha} > 0$ such that $a(v, v) \geq \underline{\alpha} \|v\|_V^2$ for all $v \in V$,
- L is *continuous*, i.e. there is a positive constant C such that $\big|L(v)\big| \leq C \|v\|_V$ for all $v \in V$,

then there is a unique solution $u \in V$ such that (2.2) holds. For a proof of this, see for example (Larsson and Thomée, 2009; Asadzadeh, 2020).

If $A_0(x) \in [\underline{\alpha}, \overline{\alpha}]$ for all $x \in \Omega$, then the bilinear form a clearly satisfies continuity, and when also applying the Poincaré inequality,

$$\|v\|_{L^2(\Omega)} \leq C \|\nabla v\|_{L^2(\Omega)} \qquad \text{for all } H_0^1(\Omega),$$

one additionally obtains coercivity. If $f \in L^2(\Omega)$, then the linear functional L is bounded. In fact, it actually suffices that $f \in H^{-1}(\Omega)$, with $\|f\|_{H^{-1}(\Omega)} = \sup_{v \in H_0^1} \big|f(v)\big| / \|v\|_{H^1(\Omega)}$, see (Asadzadeh, 2020), but for simplicity one often assumes that $f \in L^2(\Omega)$. Henceforth, we will assume that these requirements are fulfilled and that A_0 is smooth. Thus, Lax-Milgram gives us existence and uniqueness of a solution.

Remark 1. *If one wants to examine an elliptic PDE problem with other types of boundary conditions, this may lead to other test and trial spaces, and different variational formulations. For example, with Robin conditions, equation* (2.1b) *would instead read* $A_0 \frac{\partial u}{\partial \mathbf{n}} + h\,(u - g) = 0$*, and one would use the space* $H^1(\Omega)$ *as test and trial space instead of* $H_0^1(\Omega)$*. In this case, the bilinear form would be defined by* $a(u, v) = (A_0 \nabla u, \nabla v)_\Omega + (hu, v)_{\partial\Omega}$ *and the linear functional would be defined by* $L(v) = (f, v)_\Omega + (hg, v)_{\partial\Omega}$*.*

2.1.3 Regularity

The Lax-Milgram theorem gives us that there exists a solution $u \in H_0^1(\Omega)$. One question that arises is whether or not the solution has higher regularity than this. That is, whether $u \in H^k(\Omega)$ for $k > 1$. For this analysis, we further assume that the domain Ω is convex.

Looking again at equation (2.1a), one can rewrite it into

$$\Delta u = \frac{-f - \nabla A_0 \cdot \nabla u}{A_0},$$

since A_0 is assumed to be strictly positive. Since $f \in L^2(\Omega)$, A_0 is assumed to be smooth and $\nabla u \in L^2(\Omega)$ by Lax-Milgram, we conclude that $\Delta u \in L^2(\Omega)$. For convex polygonal domains Ω, it holds that $|u|_{H^2(\Omega)} \leq \|\Delta u\|_{L^2(\Omega)}$. Thus, we conclude that $u \in H^2(\Omega)$ and u is a strong solution to equation (2.1) as well.

2.2 The finite element method

The finite element method is a method to transform the potentially very hard problem (2.1) to a solvable finite-dimensional linear system, typically solved using a computer. The overall process of the finite element method, after one has obtained the weak form (2.2), can be described in the following steps:

1. Discretise the equation (2.2) by restricting the trial and test spaces to the finite-dimensional space $V_h \subset V$. The problem is then expressed in the *finite element formulation*.
2. Choose a basis for the finite-dimensional trial and test spaces in order to formulate the problem as a linear system.
3. Solve the linear system.

In this section, we first present the finite element formulation, introduce interpolation and then derive an a priori error estimate to make sure that the FEM converges properly.

2.2.1 Finite element formulation

To obtain the finite element formulation, we first introduce some *triangulation* $\mathcal{T}_h = \{K\}$ of our domain Ω, i.e. a subdivision of Ω into closed simplices that fulfil that no vertex lies in the interior of an edge of another simplex, see Figure 2.1. Here, $h = \max h_K$, where h_K is the diameter of simplex K. Letting h vary and become smaller, we have a *family of triangulations*, $\{\mathcal{T}_h\}_h$. When refining the mesh, one wants to somehow preserve the shape of the simplices. The family of triangulations should be *non-degenerate*, meaning that there is a positive constant C such that each simplex K contains a ball of radius $\rho_K \geq Ch_K$. Essentially, this requirement prevents the simplices from becoming arbitrarily thin.

Figure 2.1: Examples of a valid (left) and invalid (right) triangulation.

We restrict the infinite-dimensional space V to a finite-dimensional space $V_h \subset V$ based on $\mathcal{T}_h$. Typically, V_h can be e.g. the set of all piecewise linear functions on $\mathcal{T}_h$. The finite element formulation of our problem thus reads: find $u_h \in V_h$ such that

$$a(u_h, v_h) = L(v_h), \qquad \text{for all } v_h \in V_h. \tag{2.3}$$

Note that since $V_h \subset V$, it also holds that $a(u, v_h) = L(v_h)$ for all $v_h \in V_h$. Subtracting (2.3) from this, we obtain the so-called *Galerkin orthogonality*: $a(u - u_h, v_h) = 0$ for all $v_h \in V_h$. This means that the error $u - u_h$ between the actual solution and the approximation is orthogonal to the space V_h with respect to the scalar product $a(\cdot, \cdot)$. It also implies that the finite element (FE) approximation is the best approximation with respect to the so-called *energy norm*, which is defined by $\|\cdot\|_a^2 = a(\cdot, \cdot)$.

To concretise the FE formulation further, we choose a basis $\{\phi_i\}_{i \in I}$ for V_h, where I is the set of interior nodes in the triangulation, and let u_h and v_h be

linear combinations of the basis functions. This results in the system

$$\sum_{j\in I} \xi_j \int_\Omega A_0 \nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}x = \int_\Omega f\phi_i \,\mathrm{d}x, \qquad \text{for all } i \in I,$$

where ξ_j are the degrees of freedom. Defining a *stiffness matrix* $A = (a_{ij})$ with entries $a_{ij} = \int_\Omega A_0 \nabla\phi_j \cdot \nabla\phi_i \,\mathrm{d}x$, a vector $\xi = (\xi_j)$ with obvious entries, and a *load vector* $b = (b_i)$ with entries $b_i = \int_\Omega f\phi_i \,\mathrm{d}x$, this can be written as a matrix equation. The goal is then to find a vector ξ such that

$$A\xi = b.$$

Thus, problem (2.1) is now approximated by a finite-dimensional linear system, which can be solved with the help of a computer.

One feature that affects how fast the resulting linear system can be solved is the condition number $\kappa(A)$ of the matrix A. In section 2.3, we will see more precisely how it affects the conjugate gradient method, but first, let us see which order of magnitude one can expect for the condition number to have.

We assume a *quasi-uniform* family of triangulations, which is defined by the existence of a $\rho > 0$ such that the largest ball inside any simplex has a diameter of at least $\rho h \operatorname{diam}\Omega$. In essence, this means that the simplices in one triangulation should be around the same size. Then, for $d \geq 2$, it can be shown that for some constant C,

$$C^{-1}h^2 v^T v \leq v^T A v \leq C v^T v,$$

meaning that

$$\begin{aligned} &\lambda_{\min}(A) \geq \frac{1}{C}h^2, \\ &\lambda_{\max}(A) \leq C \end{aligned} \qquad \Rightarrow \qquad \kappa(A) = \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)} \leq \frac{C^2}{h^2}. \tag{2.4}$$

For more details on this, we refer to (Brenner and Scott, 2002). In other words, the condition number of A depends on the mesh size of the triangulation and gets worse with a finer mesh.

2.2.2 Interpolation

In order to move forward with error estimates, one needs to discuss the concept of interpolation. There are many different interpolants to choose from, and they rely on the discretisation of the problem. The classical choice is the nodal

interpolant $\mathcal{I}_h : C(\overline{\Omega}) \to V_h$. It is defined by

$$\mathcal{I}_h v(x_i) = v(x_i),$$

where x_i are the nodal points of the triangulation. The interpolation operator thus maps a function to the space V_h, but requires a continuous function to begin with. If $v \in H^2(\Omega)$, it is continuous by Sobolev's inequality, but the interpolant is not well-defined for arbitrary $v \in H^1(\Omega)$. There are other interpolants that produce similar results, that one can use instead. One example is the Scott-Zhang interpolant, which will be introduced in chapter 3.

Assuming that $v \in H^2(\Omega)$ and moving forward with the nodal interpolant, it holds that

$$\begin{aligned} \|\mathcal{I}_h v - v\|_{L^2(\Omega)} &\leq Ch^2 \|v\|_{H^2(\Omega)}, \\ \left\|\nabla\left(\mathcal{I}_h v - v\right)\right\|_{L^2(\Omega)} &\leq Ch \|v\|_{H^2(\Omega)} \end{aligned} \tag{2.5}$$

for all $v \in H^2(\Omega)$, where h is the maximum diameter of the triangulation mesh. The constant C is independent of h since the family of triangulations is non-degenerate. We refer to (Brenner and Scott, 2002; Asadzadeh, 2020; Larsson and Thomée, 2009) for further details.

2.2.3 A priori error estimate

One natural question that arises from approximating the solution u to (2.2) with the solution u_h to (2.3), is how close this approximation is to the actual one, or in other words, how big the error $u - u_h$ is. This can be measured in different ways. Firstly, one can be interested in this error in terms of either the (often unknown) solution u or the numerically gained solution u_h. An estimate in terms of the former is called an *a priori* error estimate, whereas an estimate based on the latter is called an *a posteriori* error estimate. Here, we derive an a priori error estimate as this gives us convergence of the method, as opposed to a posteriori error estimates.

Secondly, one needs to decide in which norm the error estimates should be computed. Two natural choices of the norm is either the energy norm, as introduced above, or the vector space norm, which in our example is the H^1-norm. The energy norm and the vector space norm are equivalent, see (Larsson and Thomée, 2009). Here, we use the vector space norm $\|\cdot\|_{H^1(\Omega)}$.

In order to obtain an a priori error estimate, one gets from coercivity of a,

Galerkin orthogonality and boundedness of a, that

$$\underline{\alpha}\|u-u_h\|^2_{H^1(\Omega)} \leq a(u-u_h,u-u_h) \leq a(u-u_h,u-v) \leq \\ \leq \overline{\alpha}\|u-u_h\|_{H^1(\Omega)}\|u-v\|_{H^1(\Omega)},$$

for any $v \in V_h$, leading to

$$\|u-u_h\|_{H^1(\Omega)} \leq \frac{\overline{\alpha}}{\underline{\alpha}}\|u-v\|_{H^1(\Omega)}. \tag{2.6}$$

Equation (2.6) is usually referred to as *Céa's lemma*.

Using an interpolant and a corresponding interpolation estimate, such as (2.5), this can be simplified to

$$\|u-u_h\|_{H^1(\Omega)} \leq Ch\|u\|_{H^2(\Omega)}.$$

We conclude that the FE approximation u_h converges towards the exact solution u as $h \to 0$.

2.3 Iterative methods

Given a linear system

$$Ax = b, \tag{2.7}$$

where A is an $n \times n$ symmetric and positive-definite matrix, one can either solve it by using a direct method or an iterative one. When the linear system is large, which PDE problems often are, one needs to use an iterative algorithm, since direct methods require considerably more memory. The algorithm used in the appended article is based on the conjugate gradient method. We proceed by first explaining the idea behind it, then presenting the algorithm, and lastly analysing the convergence of the method.

2.3.1 The idea behind the conjugate gradient method

Two vectors u and v are said to be *A-orthogonal* or *conjugate* with respect to the matrix A if $u^TAv = 0$. This defines a scalar product since A is symmetric and positive-definite: $u^TAv = (u,Av) = (Au,v) = (u,v)_A$.

If $P = \{p_1,...,p_n\}$ are mutually conjugate vectors, P constitutes a basis for $\mathbb{R}^n$.

That means that any vector, and specifically the optimal solution x_* to our linear system (2.7), can be written as a sum of the vectors p_i: $x_* = \sum_i \alpha_i p_i$. Hence, $Ax_* = \sum_i \alpha_i A p_i$. Multiplying (2.7) by one of the conjugate vectors p_k, we obtain

$$p_k^T b = p_k^T A x_* = \sum_i \alpha_i p_k^T A p_i = \sum_i \alpha_i (p_k, p_i)_A = \alpha_k (p_k, p_k)_A,$$

because of conjugacy between the vectors p_i. This gives us $\alpha_k = \frac{(p_k, b)}{(p_k, p_k)_A}$.

In other words, the exact solution x_* can be found by first computing n conjugate vectors with respect to A, and then for each of them, computing α_k as above.

Finding n conjugate vectors can be a very time- and memory-consuming task, though, if the system (2.7) is large. In those cases, it might be better to turn this into an iterative algorithm by finding one or more conjugate vectors and terminating when the approximation of the solution is good enough.

2.3.2 The algorithm

In an iterative method, one needs an initial value x_0 that improves in each iteration. We can assume that $x_0 = 0$. Just as in the gradient descent method, in the first iteration, we choose to move in the direction of $p_0 = b - Ax_0$. In the following iterations, though, the direction p_k should be conjugate to the previous directions p_i, $i < k$. All directions p_k, $k > 0$, will be conjugate to the gradient $Ax - b$ of the minimising function $\frac{1}{2}x^T Ax - x^T b$, which is where the name of the method derives from.

The approximation x_k of the solution of equation (2.7) will be updated in each iteration as per

$$x_{k+1} = x_k + \alpha_k p_k, \tag{2.8}$$

for some step length α_k. The residual is defined as $r_k = b - Ax_k$. Using (2.8), it is easy to verify that r_{k+1} will be given by

$$r_{k+1} = r_k - \alpha_k A p_k, \tag{2.9}$$

which is computationally cheaper to compute in each iteration since the product Ap_k will need to be computed either way, and will thus be used from now on.

In addition to the directions p_k being mutually conjugate, we also require that the residuals r_k in each iteration are orthogonal to each other, since that leads

to several desirable properties. To construct the step directions p_k, we use this fact, and build the directions in a Gram-Schmidt-like manner, using the current residual and subtracting components of the previous directions p_i in order to preserve conjugacy between directions:

$$p_k = r_k - \sum_{i=0}^{k-1} \beta_{ki} p_i. \tag{2.10}$$

We will soon determine the constants β_{ki} in equation (2.10), but first, we will focus our attention on the step lengths α_k.

Since $p_0 = r_0$, equation (2.10) means that the direction p_k will be a linear combination of the residuals r_i, $i = 0, ..., k$. Because of our condition that the residuals are mutually orthogonal, that also means that the next residual r_{k+1} is orthogonal to the direction p_k. This makes sense since that means that we do not have to move in any of the old directions again. In other words, the step length in every iteration will be optimal.

Orthogonality between r_{k+1} and p_k, i.e. $(r_{k+1}, p_k) = 0$, together with equation (2.9) leads to $(r_k, p_k) - \alpha_k (p_k, p_k)_A = 0$. We thus obtain $\alpha_k = \frac{(r_k, p_k)}{(p_k, p_k)_A}$. Now, taking the inner product between equation (2.10) and r_k, we notice that

$$(p_k, r_k) = (r_k, r_k) - \sum_{i=0}^{k-1} \beta_{ki} (p_i, r_k) = (r_k, r_k),$$

which means that we can formulate α_k as

$$\alpha_k = \frac{(r_k, r_k)}{(p_k, p_k)_A}. \tag{2.11}$$

Now returning to the constants β_{ki}, we take the inner product between (2.10) and Ap_ℓ:

$$\begin{aligned} (p_k, Ap_\ell) &= (r_k, Ap_\ell) - \sum_{i=0}^{k-1} \beta_{ki} (p_i, Ap_\ell) \\ 0 &= (r_k, p_\ell)_A - \beta_{k\ell} (p_\ell, p_\ell)_A, \end{aligned}$$

leading to

$$\beta_{k\ell} = \frac{(r_k, p_\ell)_A}{(p_\ell, p_\ell)_A}, \qquad \ell < k.$$

In order to simplify the nominator, we take the inner product between r_ℓ and

equation (2.9):

$$\begin{aligned}(r_\ell, r_{k+1}) &= (r_\ell, r_k) - \alpha_k (r_\ell, Ap_k) \\ \alpha_k (r_\ell, Ap_k) &= (r_\ell, r_k) - (r_\ell, r_{k+1}) \qquad \{r_i \perp r_j, \text{ if } i \neq j\} \\ (r_\ell, p_k)_A &= \begin{cases} \frac{1}{\alpha_\ell}(r_\ell, r_\ell), & \ell = k, \\ \frac{1}{\alpha_{\ell-1}}(r_\ell, r_\ell), & \ell = k+1, \\ 0, & \text{else.} \end{cases}\end{aligned}$$

Thus, most of the β_{kl}-terms disappear:

$$\beta_{k\ell} = \begin{cases} -\frac{1}{\alpha_{k-1}} \frac{(r_k, r_k)}{(p_{k-1}, p_{k-1})_A}, & k = \ell + 1, \\ 0, & \text{else.} \end{cases}$$

Using this together with equation (2.11), we rename $\beta_{k\ell}$, change signs and simplify:

$$\beta_k = \frac{(r_{k+1}, r_{k+1})}{(r_k, r_k)}. \tag{2.12}$$

Thus, equation (2.10) can be updated to

$$p_{k+1} = r_{k+1} + \beta_k p_k. \tag{2.13}$$

We are now ready to present the conjugate gradient method as a whole in Algorithm 1.

Algorithm 1 The conjugate gradient method

x_0 gets some initial value
$r_0 = p_0 = b - Ax_0$
$k = 0$
while r_k small enough **do**
  $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = r_k - \alpha_k A p_k$
  $\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
  $p_{k+1} = r_{k+1} + \beta_k p_k$
  $k = k + 1$
end while
return x_k

2.3.3 Convergence

One of the implications of the conjugate gradient method is that both $\{r_i\}_{i=0}^k$ and $\{p_i\}_{i=0}^k$ span the same subspace. It is easy to see that this subspace is a so-called *Krylov subspace* - a subspace that is generated by repeatedly applying one matrix to one vector. In this case, the subspace is $\text{span}\{r_0, Ar_0, A^2r_0, ...\}$. Since the subspace is Krylov, one can express the error in iteration k as

$$x_* - x_k = P_k(A)\left(x_* - x_0\right),$$

where P_k is a polynomial of order $\leq k$ such that $P_k(0) = 1$. For more details on this, we refer to (Demmel, 1997).

Letting v_i denote the orthonormal eigenvectors to the matrix A, and λ_i the corresponding eigenvalues, we have that $P_k(A)v_i = P_k(\lambda_i)v_i$. With the initial error written using v_i as a basis, i.e. $x_* - x_0 = \sum_{i=1}^n \xi_i v_i$, in iteration k it holds that:

$$x_* - x_k = \sum_{i=1}^n \xi_i P_k(\lambda_i) v_i,$$

$$A(x_* - x_k) = \sum_{i=1}^n \xi_i P_k(\lambda_i)\lambda_i v_i,$$

which gives us

$$\|x_* - x_k\|_A^2 = \sum_{i=1}^n \xi_i^2 P_k(\lambda_i)^2 \lambda_i.$$

The conjugate gradient method finds x_k (and hence $P_k(A)$) so that this expression is minimised, meaning that we obtain

$$\begin{aligned}\|x_* - x_k\|_A^2 \leq \min_{\substack{P_k \\ P(0)=1}} \max_{\lambda\in\sigma(A)} P_k(\lambda)^2 \sum_{i=1}^n \xi_i^2\lambda_i = \\ = \min_{\substack{P_k \\ P(0)=1}} \max_{\lambda\in\sigma(A)} P_k(\lambda)^2 \|x_* - x_0\|_A^2,\end{aligned}$$

where $\sigma(A)$ denotes the spectrum of A. By rewriting the polynomial P_k as a scaled and shifted Chebyshev polynomial, one can show that the following holds:

$$\|x_k - x_*\|_A \leq 2\left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1}\right)^k \|x_0 - x_*\|_A, \tag{2.14}$$

where $\kappa(A) = \lambda_{\max}(A)/\lambda_{\min}(A)$ is the condition number of A. For more details on this, we again refer to (Demmel, 1997).

Our conclusion is thus that the convergence of the CG method depends on the condition number of the matrix A. A higher condition number leads to slower convergence. When the matrix A is ill-conditioned, which is the case for PDE problems in general, it is thus a good idea to find a suitable preconditioner in order to achieve faster convergence.

3 Mixed-dimensional PDEs

Oftentimes, in real world applications, thin cracks, fractures or reinforcements occur in domains where PDEs are applicable. This raises a need to expand the model problem in Section 2.2 to one that can handle such cases. Examples of applications with these kinds of structures can be flow in porous media, such as permeable rocks with cracks, cell membranes, see (Schwartz et al., 2005), or blood vessels in tissue, see (Fritz et al., 2022). It is often advantageous to model such structures as lower dimensional interfaces instead of narrow 3-dimensional cracks or similar. This results in mixed-dimensional models with bulk areas and interfaces that need to be coupled.

There are different ways to solve such mixed-dimensional PDEs. The main method in this thesis is the fitted finite element method, which is also seen in e.g. (Arrarás et al., 2019; Boon et al., 2018; Jaffre et al., 2006). There are also a few other techniques, for example trace based methods, see e.g. (Burman et al., 2019, 2015; Olshanskii et al., 2009).

In this chapter, we will first go through the problem formulation of mixed-dimensional PDEs, as formulated in the article, followed by a description of the fitted finite element method. Thereafter, we will use a domain decomposition approach to precondition the arising linear system, before solving it with the CG method.

3.1 A mixed-dimensional model problem

In this section, we start by formulating a mixed-dimensional PDE problem. First, we go through the geometry needed in order to properly formulate the problem, and then we present a strong formulation, which primarily serves as intuitional understanding of where the weak formulation comes from, since it is not certain that there is enough smoothness in the problem for a strong

solution to be well-defined. The weak formulation is then presented, along with some results on existence, uniqueness and regularity of a solution.

3.1.1 Problem formulation

We consider an open, connected domain $\Omega \in \mathbb{R}^d$, where $d = 2$ or 3. We let Ω be convex and partitioned into subdomains Ω^c of codimensionalities $c = 0, 1, ..., d$ such that

$$\Omega = \bigcup_c \Omega^c.$$

The codimensionality indicates the number of dimensions lower than d. For example, in $\mathbb{R}^2$, Ω^0 may consist of planes, Ω^1 of lines, and Ω^2 of dots. Each subdomain is similarly partitioned into subdomain segments $\Omega^c_{\ell_c}$:

$$\Omega^c = \bigcup_{\ell_c} \Omega^c_{\ell_c}.$$

We assume that each subdomain segment is open and dense. For details on the topologies, see the appended paper. We further assume that either $\Omega^{c+1}_{\ell_{c+1}} \subseteq \partial\Omega^c_{\ell_c}$ or $\Omega^{c+1}_{\ell_{c+1}} \cap \overline{\Omega^c_{\ell_c}} = \emptyset$, so the lower dimensional subdomain segments are either fully part of the boundary to a higher dimensional subdomain segment, or not at all.

Essentially, these requirements mean that interface segments separate bulk segments, and points (in a 2-dimensional setting) separate interface segments. See Figure 3.1 for an example domain. For simplicity, we also assume that all subdomain segments are polytopal with Lipschitz boundaries, ruling out slits that go into subdomain segments. Now, we can introduce the adjacency relation

$$(\ell_c, \ell_{c+1}) \in E_c \qquad \text{if } \Omega^{c+1}_{\ell_{c+1}} \subseteq \partial\Omega^c_{\ell_c}, \tag{3.1}$$

which is needed when expressing the PDE. Equation 3.1 says that if for example an interface, of codimensionality 1, is part of the boundary of a bulk segment, which has codimensionality 0, then the corresponding pair of indices belongs to the set E_0. The set E_0 will thus contain information about which interfaces coincide with which bulk boundaries and E_1 will correspondingly contain information about which segments of codimension 2 coincide with which interface boundaries.

We will restrict our focus to the subdomain codimensionalities $0, 1$ and 2. To

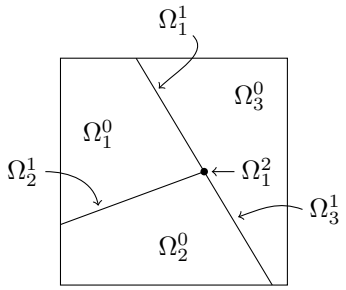

Figure 3.1: An example of a domain Ω with interfaces.

simplify notation, we therefore let

$$\Omega^0 = \bigcup_{i \in I} \Omega_i^0, \qquad \Omega^1 = \bigcup_{j \in J} \Omega_j^1, \qquad \Omega^2 = \bigcup_{k \in K} \Omega_k^2.$$

Moving on to the strong formulation of a PDE problem on such a domain, we consider an equation of the form (2.1a) on the bulk and interface areas, couple them by Robin conditions, and apply homogeneous Dirichlet conditions on the outer boundary $\partial\Omega$. We then seek a solution u, split into the parts $u_i^0 : \overline{\Omega_i^0} \to \mathbb{R}$ on the bulk segments and $u_j^1 : \overline{\Omega_j^1} \to \mathbb{R}$ on the interface segments. We formulate the differential equation as

$$\begin{aligned} -\nabla \cdot \left(A_i \nabla u_i^0 \right) &= f_i \quad \text{in } \Omega_i^0, \\ u_i^0 &= 0 \quad \text{on } \overline{\Omega_i^0} \cap \partial\Omega, \\ -\nabla_\tau \cdot \left(A_j \nabla_\tau u_j^1 \right) - B_j \left(u_i^0 - u_j^1 \right) &= f_j \quad \text{in } \partial\Omega_i^0 \cap \Omega_j^1, \\ u_j^1 &= 0 \quad \text{on } \overline{\Omega_j^1} \cap \partial\Omega, \end{aligned} \tag{3.2}$$

where ∇_τ denotes the gradient in the tangential direction. The diffusion coefficients $A_i : \overline{\Omega_i^0} \to \mathbb{R}$ and $A_j : \overline{\Omega_j^1} \to \mathbb{R}$ fulfil $0 < \underline{\alpha} \leq A_i, A_j \leq \overline{\alpha}$ for some constants $\underline{\alpha}$, $\overline{\alpha}$, the Robin coupling coefficient $B_j : \Omega_j^1 \to \mathbb{R}$ similarly fulfils $0 < \underline{\beta} \leq B_j \leq \overline{\beta}$ for constants $\underline{\beta}$, $\overline{\beta}$, and the source terms $f_i : \Omega_i^0 \to \mathbb{R}$ and $f_j : \overline{\Omega_j^1} \to \mathbb{R}$ are assumed to be L^2.

Looking at equation 3.2, it consists of differential equations on the bulk and interface areas along with boundary conditions at the outer boundary $\partial\Omega$. We need additional coupling conditions where different subdomains meet. We insert a Robin type coupling condition between the bulk and interface segments,

require continuity between interface segments and express a Kirchoff's law equation between interface segments:

$$\mathbf{n}_{\Omega_i^0} \cdot A_i \nabla u_i^0 + B_j u_i^0 = B_j u_j^1 \qquad \text{on } \partial\Omega_i^0 \cap \Omega_j^1, \tag{3.3a}$$

$$u_j^1 = u_{j'}^1 \qquad \text{on } \partial\Omega_j^1 \cap \partial\Omega_{j'}^1, \tag{3.3b}$$

$$\sum_{j:(j,k)\in E_1} \mathbf{n}_{\Omega_j^1} \cdot A_j \nabla_\tau u_j^1 = 0 \qquad \text{on } \partial\Omega_j^1 \cap \Omega_k^2. \tag{3.3c}$$

3.1.2 Variational formulation, existence and uniqueness

In order to express equations (3.2) - (3.3) weakly, we need to properly define the vector spaces and norms that we will use. For bulk functions v^0, we define the space

$$V^0 = H^1(\Omega^0) = \prod_{i\in I} H^1(\Omega_i^0),$$

with $V_0^0 = \{v \in V^0 : v|_{\partial\Omega} = 0\}$. Similarly, for interface functions v^1, we first define

$$V_b^1 = H^1(\Omega^1) = \prod_{j\in J} H^1(\Omega_j^1).$$

However, these functions need to be continuous over codimension 2 as per equation 3.3b, which they are not in V_b^1. Hence, we further define

$$V^1 = \{v^1 \in V_b^1 : v_j^1|_{\Omega_k^2} = v_{j'}^1|_{\Omega_k^2} \text{ for all pairs } (j,k),(j',k) \in E_1\},$$

where the continuity at intersections is enforced. We also let $V_0^1 = \{v \in V^1 : v\,|_{\partial\Omega} = 0\}$ be the corresponding space with homogeneous boundary conditions.

We now combine these spaces into:

$$V = V^0 \times V^1 \qquad \text{and} \qquad V_0 = V_0^0 \times V_0^1,$$

on which we also define the norm

$$\|v\|_V^2 = \|v^0\|_{H^1(\Omega^0)}^2 + \|v^1\|_{H^1(\Omega^1)}^2.$$

These spaces are Hilbert spaces, allowing us to later apply the Lax-Milgram theorem.

Through a similar process as in Section 2.2, we multiply the differential equations in 3.2 by $v^0 \in V_0^0$ and $v^1 \in V_0^1$ respectively, integrate, apply Green's formula and adhere to the boundary conditions (3.3). Consequently, we obtain the variational formulation: find $u \in V_0$ so that

$$a(u, v) = F(v) \text{ for all } v \in V_0, \tag{3.4}$$

where

$$\begin{aligned} a(v, w) = &\sum_{i \in I} (A_i \nabla v_i^0, \nabla w_i^0)_{\Omega_i^0} + \sum_{j \in J} (A_j \nabla_\tau v_j^1, \nabla_\tau w_j^1)_{\Omega_j^1} \\ &+ \sum_{(i,j) \in E_0} (B_j (v_i^0 - v_j^1), w_i^0 - w_j^1)_{\Omega_j^1}, \qquad \text{and} \\ F(w) = &\sum_{i \in I} (f_i, w_i^0)_{\Omega_i^0} + \sum_{j \in J} (f_j, w_j^1)_{\Omega_j^1}. \end{aligned}$$

Here, $w_{\ell_c}^c$ denotes the restriction of w^c to the subdomain segment $\Omega_{\ell_c}^c$.

In order to deduce existence and uniqueness of a solution, one has to show that $F(v)$ is bounded, which is true since we assume that $f_i \in L^2(\Omega_i^0)$ and $f_j \in L^2(\Omega_j^1)$, and that the bilinear form a is coercive and bounded.

It is not straightforward to show that a is coercive, since the domain geometry allows for more complicated partitions, where subdomain segments are detached from the boundary $\partial\Omega$. To show coercivity, one has to go through an iterative process, starting at some subdomain segment at the boundary and reaching all the other subdomain segments along the way. For a proof of coercivity, we refer to the paper appended to the thesis. One obtains the bound

$$a(v, v) \geq C \left(\underline{\alpha}^{-1} + \underline{\beta}^{-1} \right)^{-1} \|v\|_V^2 , \tag{3.5}$$

where C depends on the geometry of the problem.

The proof to show boundedness of a is more comprehensible, but again, we refer to the article to see it in detail. The bound one obtains reads

$$a(v, w) \leq C \left(\overline{\alpha} + \overline{\beta} \right) \|v\|_V \|w\|_V , \tag{3.6}$$

where again C depends on the geometry of the problem.

Using the Lax-Milgram theorem, we conclude that there exists a unique solution $u \in V_0$ that solves equation 3.4.

3.1.3 Regularity

For $\Omega \subset \mathbb{R}^2$, one can obtain regularity estimates through a similar process as the one presented in Section 2.1. The Robin coupling conditions complicate the proof slightly though, along with the fact that subdomain segments are allowed to be non-convex. For this setting, one obtains that $u_i^0 \in H^{3/2}(\Omega_i^0)$ and $u_j^1 \in H^2(\Omega_j^1)$. If the subdomains are convex, one additionally gets that $u_i^0 \in H^2(\Omega_i^0)$. For a proof of this, we refer to Appendix A in the appended paper.

In the case of H^2-regularity, the strong formulation (3.2) - (3.3) is well-defined and corresponds to the weak formulation (3.4).

3.2 Fitted finite element method

In this section, we go through numerical methods to solve mixed-dimensional systems, with heavy emphasis on the fitted finite element method. We further go through interpolation in the mixed-dimensional setting, a priori error estimates and, lastly, an iterative method to solve the resulting linear system.

3.2.1 Fitted finite element formulation

The main question when formulating a finite element method for a mixed-dimensional problem is how to define the triangulation on the more complex geometry that these kinds of problems have. There are several methods to manage geometries like these. In (Dziuk and Elliott, 2013), several methods to solve PDEs on this kind of geometries are reviewed. One plausible method is to use *CutFEM*, where one utilises an unfitted, structured mesh that is not adapted to the underlying geometry, where only the mesh cells that are "cut" by interfaces will need special treatment. See e.g. (Hansbo and Hansbo, 2002).

We will however turn our attention towards the *fitted* finite element method, which in contrast to CutFEM uses a fitted mesh that is adapted to the underlying geometry. Fitted FEM is also used in a lot of research, see for example (Arrarás et al., 2019; Boon et al., 2018; Jaffre et al., 2006).

To define a triangulation on a mixed-dimensional geometry, we first define a triangulation on the subdomains of codimension 1. Let $\mathcal{T}_{h,j}^1$ be triangulations of $\overline{\Omega_j^1}$, and $\mathcal{T}_h^1$ be the union of all of them. We define a family of triangulations

$\{\mathcal{T}^1_{h,j}\}_h$ that should be non-degenerate.

Correspondingly, we define $\mathcal{T}^0_{h,i}$ to be triangulations of $\overline{\Omega^0_i}$, with $\mathcal{T}^0_h = \bigcup_{i\in I} \mathcal{T}^0_{h,i}$ being the union of them. Again, we let $\{\mathcal{T}^0_h\}_h$ be a non-degenerate family of triangulations. The parameter h is the maximum diameter of the simplices.

The simplices $\{K^0\}$ of $\mathcal{T}^0_h$ and the simplices $\{K^1\}$ of $\mathcal{T}^1_h$ are connected in the following way: either K^1 constitutes an edge of a simplex K^0, or their intersection is at most part of the boundary of K^1. An example of such a triangulation in 2 dimensions is shown in figure 3.2.

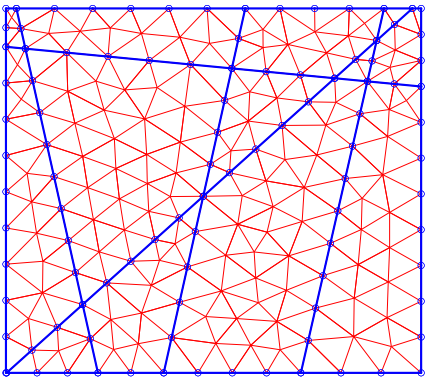

Figure 3.2: Example of a 2-dimensional domain with interfaces represented as blue thick lines. The mesh is marked in red in the 2-dimensional subdomains, and mesh points are marked with blue rings on the 1-dimensional interfaces.

When we have a discretisation of a domain, we need to define the discretised test and trial spaces. We let

$$V^0_h = \prod_{i\in I} \{v^0 \in \mathbb{C}(\overline{\Omega^0_i}) : v^0 \text{ piecewise linear on } \mathcal{T}^0_i, v^0|_{\partial\Omega} = 0\},$$

and

$$V^1_{b,h} = \prod_{j\in J} \{v^1 \in \mathbb{C}(\overline{\Omega^1_j}) : v^1 \text{ piecewise linear on } \mathcal{T}^1_j, v^1|_{\partial\Omega} = 0\},$$

$$V^1_h = \prod_{j\in J} \{v^1 \in V^1_{b,h} : v^1_j|_{\Omega^2,k} - v^1_{j'}|_{\Omega^2,k} = 0 \text{ for all pairs } (j,k),(j',k) \in E_1\}.$$

From these spaces, we also define the composite space $V_h = V^0_h \times V^1_h$.

Now, we can formulate the finite element problem: find $u_h \in V_h$ such that

$$a(u_h, v_h) = F(v_h) \text{ for all } v_h \in V_h. \tag{3.7}$$

Since V_h is also a Hilbert space, and the bounds (3.5) - (3.6) hold for equation (3.7) as well, Lax-Milgram gives us existence and uniqueness of a solution.

3.2.2 Interpolation

In our setup, the nodal interpolant would not be well-defined since there is no requirement that the functions are continuous on the different subdomain segments. We will instead make use of the Scott-Zhang interpolant and define an interpolant $\mathcal{I}_h : V_0 \to V_h$ by its components $\mathcal{I}_h(v^0, v^1) = (\mathcal{I}_h^0 v^0, \mathcal{I}_h^1 v^1)$, where $\mathcal{I}_h^0 : V_0^0 \to V_h^0$ and $\mathcal{I}_h^1 : V_0^1 \to V_h^1$ are Scott-Zhang interpolants.

To introduce the concept of the Scott-Zhang interpolant, we first need to make a few other definitions. All the nodes z in our triangulation lie on the boundary of simplices K. For each node z, we choose a simplex K_z such that $z \in K_z$. We then let $\tilde{K}_z$ be an edge (or equivalent in other dimensions) of K_z. If $z \in (\partial\Omega \cup \Omega^1)$, the simplex and edge are chosen such that $\tilde{K}_z \subset K_z \cap (\partial\Omega \cup \Omega^1)$. See an example of these notations in Figure 3.3.

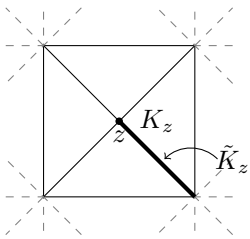

Figure 3.3: An example of a node z with a corresponding simplex K_z and $\tilde{K}_z$.

Further, we let $\tilde{\mathcal{P}}_z$ be the finite element space V_h restricted to $\tilde{K}_z$, and $\tilde{\mathcal{N}}_z$ be the set of nodes that correspond to $\tilde{\mathcal{P}}_z$. We let $\{\phi_N^{\tilde{K}_z} : N \in \tilde{\mathcal{N}}_z\}$ be the nodal basis for $\tilde{\mathcal{P}}_z$ and $\{\psi_N^{\tilde{K}_z} : N \in \tilde{\mathcal{N}}_z\}$ be the corresponding $L^2(\tilde{K}_z)$-basis. Then we have

$$\psi_M^{\tilde{K}_z}(\phi_N^{\tilde{K}_z}) = \int_{\tilde{K}_z} \psi_M^{\tilde{K}_z} \phi_N^{\tilde{K}_z} \, \mathrm{d}x = \delta_{MN} \qquad \text{for all } M, N \in \tilde{\mathcal{N}}_z,$$

where δ is the Kronecker delta.

The local average $\tilde{N}_z(v)$ of a function v around a node z can now be defined as

$$\tilde{N}_z(v) = \psi_{N_z}^{\tilde{K}_z}(v) = \int_{\tilde{K}_z} \psi_{N_z}^{\tilde{K}_z} v(x)\,\mathrm{d}x,$$

which leads us to defining the interpolants by

$$\mathcal{I}_h^0 v_i^0 = \sum_z \tilde{N}_z(v_i^0)\phi_z^0$$

for $v^0 \in V^0$ and

$$\mathcal{I}_h^1 v_j^1 = \sum_z \tilde{N}_z(v_j^1)\phi_z^1$$

for $v^1 \in V^1$. An important property of the Scott-Zhang interpolant is that $\mathcal{I}_h v = v$ for all $v \in V_h$.

With these definitions, we obtain the interpolation error estimate

$$\|v - \mathcal{I}_h v\|_V \leq Ch \left(\left\| D^2 v^0 \right\|_{L^2(\Omega^0)} + \left\| D^2 v^1 \right\|_{L^2(\Omega^1)} \right), \tag{3.8}$$

for all $v \in V$. For a proof of this, we refer to (Brenner and Scott, 2002; Scott and Zhang, 1990).

3.2.3 A priori error estimate

Now that we have an interpolation error estimate, we are ready to derive an a priori error estimate for this mixed-dimensional setting. From coercivity of a (equation (3.5)), Galerkin orthogonality and boundedness of a (equation (3.6)), we derive

$$\begin{aligned}\frac{1}{C(\underline{\alpha}^{-1} + \underline{\beta}^{-1})} \|u - u_h\|_V^2 &\leq a(u - u_h, u - u_h) \leq a(u - u_h, u - v) \leq \\ &\leq C \left(\overline{\alpha} + \overline{\beta} \right) \|u - u_h\|_V \|u - v\|_V ,\end{aligned}$$

for all $v \in V_h$, and thus we obtain Céa's lemma for this mixed-dimensional setting:

$$\|u - u_h\|_V \leq C \left(\underline{\alpha}^{-1} + \underline{\beta}^{-1} \right) \left(\overline{\alpha} + \overline{\beta} \right) \|u - v\|_V .$$

Letting $v = \mathcal{I}_h u$ and using equation (3.8), we formulate the final a priori error

estimate:

$$\|u - u_h\|_V \leq Ch \left(\underline{\alpha}^{-1} + \underline{\beta}^{-1}\right) \left(\overline{\alpha} + \overline{\beta}\right) \left(\left\|D^2 u^0\right\|_{L^2(\Omega^0)} + \left\|D^2 u^1\right\|_{L^2(\Omega^1)}\right). \tag{3.9}$$

We can thus conclude that the fitted finite element approximation u_h does converge towards u when $h \to 0$.

3.3 Iterative methods

In order to solve the linear system that arises from problem (3.7), we want to make use of the conjugate gradient method that was presented in section 2.3. The condition number is in general very large, though, so we we need to multiply it by a preconditioner, and hence use the slightly modified *preconditioned conjugate gradient (PCG) method* instead.

In this section, we will first go through how one can rewrite the linear system stemming from equation (3.7) into two separate equations corresponding to the bulk and interfaces, then we briefly introduce the PCG method, and lastly, we focus on constructing a preconditioner that improves the convergence of the method.

3.3.1 Schur complement

Equation (3.7) can be written in matrix form as $AU = b$. The solution u_h of equation (3.7) consists of two parts, u_h^0 and u_h^1. With $\{\varphi_k^0\}$ and $\{\varphi_\ell^1\}$ being the standard Lagrange basis functions of V_h^0 and V_h^1 respectively, u_h^0 and u_h^1 can be written as sums $u_h^0 = \sum_k U_k^0 \varphi_k^0$ and $u_h^1 = \sum_\ell U_\ell^1 \varphi_\ell^1$. Hence, the matrix form of the equation can instead be split up (possibly after reordering) into parts that correspond to the bulk and interface areas:

$$\left[\begin{array}{c|c} A_{00} & A_{01} \\ \hline A_{10} & A_{11} \end{array}\right] \begin{bmatrix} U_0 \\ U_1 \end{bmatrix} = \begin{bmatrix} b_0 \\ b_1 \end{bmatrix}. \tag{3.10}$$

Here, A_{00} describes the degrees of freedom in the bulk, and A_{11} describes the degrees of freedom on the interfaces. The submatrices A_{01} and A_{10}, which are each other's transposes, describe the connections between the bulk and

interfaces. From equation (3.10), we get two separate equations,

$$A_{00}U_0 + A_{01}U_1 = b_0, \qquad \text{and} \tag{3.11a}$$
$$A_{10}U_0 + A_{11}U_1 = b_1. \tag{3.11b}$$

Equation (3.11a) can be rewritten into

$$A_{00}U_0 = b_0 - A_{01}U_1. \tag{3.12}$$

This equation is solvable with reasonable time and memory consumption because of the block diagonal stucture of A_{00}, which comes from the fact that the subdomains $\{\Omega_i^0\}_i$ are disconnected from one another. It does require that we know U_1, though. In order to obtain U_1, we form what is known as the *Schur complement*, using equations (3.11b) and the resulting expression for U_0 from equation (3.12):

$$\left(A_{11} - A_{10}A_{00}^{-1}A_{01}\right)U_1 = b_1 - A_{10}A_{00}^{-1}b_0.$$

Letting $\tilde{A}_{11} = A_{11} - A_{10}A_{00}^{-1}A_{01}$ and $\tilde{b}_1 = b_1 - A_{10}A_{00}^{-1}b_0$, this can be rewritten as

$$\tilde{A}_{11}U_1 = \tilde{b}_1. \tag{3.13}$$

This linear equation does not have the nice properties that (3.12) has, making it more time and memory consuming to solve. Note that $\tilde{A}_{11}$ is still symmetric and positive definite.

In order to solve equation (3.13) more efficiently, we want to make use of the preconditioned conjugate gradient method using a suitable preconditioner.

3.3.2 Preconditioned conjugate gradient method

As noted before, the general linear system (2.7) often comes badly conditioned, as in the case of equation (3.13). By finding a symmetric, positive definite matrix T^{-1} whose inverse T multiplied by the linear system leads to a better conditioned system, one can from the CG method derive the preconditioned conjugate gradient (PCG) method. In other words, the PCG method approximately solves the preconditioned linear system

$$TAx = Tb.$$

The reason why this requires a modification of the CG method is that the product TA in general is *not* symmetric and positive definite. It is required to

be symmetric and positive definite with respect to the scalar product induced by A, though. We will not go through the PCG method here in detail because of its similarity to the CG method, but instead we refer to e.g. (Demmel, 1997) for interested readers. It is important to note, though, that the inequality (2.14) still holds for the preconditioned system, with the only difference being that κ instead is the condition number of the product TA:

$$\|x_k - x_*\|_A \leq 2 \left(\frac{\sqrt{\kappa(TA)} - 1}{\sqrt{\kappa(TA)} + 1} \right)^k \|x_0 - x_*\|_A , \tag{3.14}$$

Some examples of simple preconditioners one may use are the Jacobi preconditioner, where one keeps the diagonal values of A in order to construct T^{-1}, and let all non-diagonal entries be 0, and incomplete Cholesky factorisation. These preconditioners are not enough for the problems in the appended article, though, which is why we present another preconditioner, based on domain decomposition.

3.3.3 A preconditioner based on domain decomposition

The preconditioner that is implemented in the article is proposed in (Kornhuber and Yserentant, 2016; Görtz et al., 2022). The first step is to introduce a quasi-uniform coarse mesh $\mathcal{T}_H$ of the domain, that does not take the interfaces or the meshes $\mathcal{T}_h^0$ and $\mathcal{T}_h^1$ into consideration. See Figure 3.4. On this mesh, we define a corresponding finite element space W_H with standard Lagrange basis $\{\phi_j\}_{j=1}^N$.

The idea behind the preconditioner is then to compute the inverse of the restriction of $\tilde{A}_{11}$ to the overlapping subdomains (subdomains with regards to the mesh $\mathcal{T}_H$), defined by the support of the Lagrange basis functions. That means that the preconditioner will be a sum of a lot of matrices with non-zero elements only where the corresponding Lagrange basis functions are supported. To construct these parts of the preconditioner, we define $W_j = \{v \in V_h^1 : \text{supp}(v) \subset \text{supp}(\phi_j)\}$, which are subspaces of V_h^1, for $j = 1, ..., N$. Together, these spaces constitute V_h^1, making it possible to write any $v \in V_h^1$ as a sum of functions in these spaces; $v = \sum_{j=1}^N v_j$, where $v_j \in W_j$.

We denote the size of the subspace W_j by m_j and let $Q_j \in \mathbb{R}^{m \times m_j}$ be prolongation matrices associated with subspace W_j. Prolongation matrices are matrices that take a subdomain and prolong it to the full domain typically by using zeros, in this case thus mapping W_j to V_h^1. The transpose Q_j^T will on the other hand re-

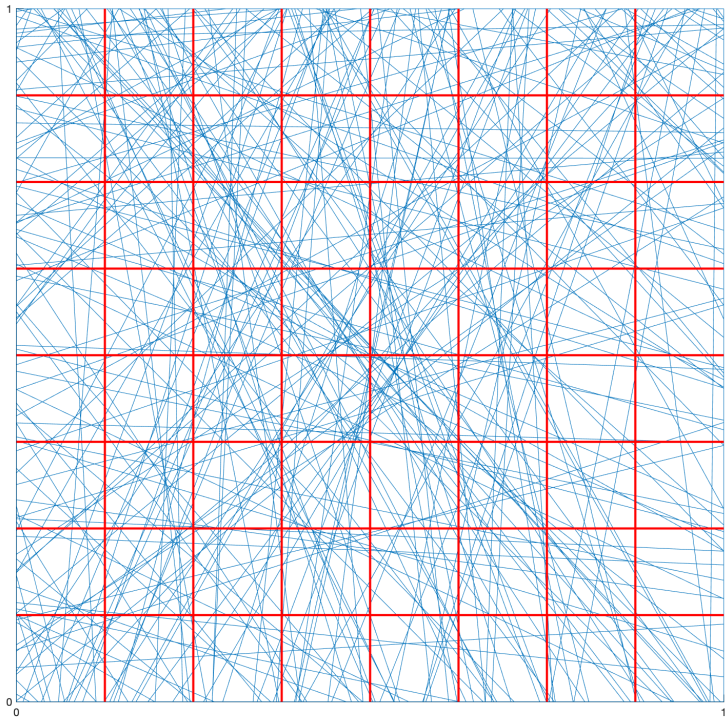

Figure 3.4: An example of a coarse mesh $\mathcal{T}_H$ (marked with red) of a domain Ω, which does not take into account where interfaces (marked with blue) lie.

strict a function v to patch j. The expression $x_j = (Q_j^T A Q_j)^{-1} Q_j^T v$ solves a linear system on the same patch, corresponding to the equation $a(\hat{x}_j, w) = (v, w)$ for all $w \in W_j$, where $\hat{x}_j$ is a function on W_j whose nodal values correspond to x_j. The product $Q_j x_j$ then prolongs the solution to the full space again.

With this logic, the contributing term to the preconditioner of each subdomain will be $T_j = Q_j (Q_j^T \tilde{A}_{11} Q_j)^{-1} Q_j^T$, $j = 1, ..., N$.

One could settle for these parts alone, but by also adding another matrix corresponding to the coarse mesh itself, but not the interfaces inside, one drastically improves the convergence of the PCG method, since the coarse scale effects are then captured directly. To construct this extra part, we let

$\mathcal{I}_h^{\text{nodal}} : W_H \to V_h^1$ be the nodal interpolant from W_H to V_h^1. Then we can define the space $W_0 = \mathcal{I}_h^{\text{nodal}} W_H$, which is a subset of V_h^1. We define Q_0 the same way as above, and similarly let $T_0 = Q_0(Q_0^T \tilde{A}_{11} Q_0)^{-1} Q_0^T$.

We are then ready to form the full preconditioner as

$$T = \sum_{j=0}^{N} T_j = \sum_{j=0}^{N} Q_j (Q_j^T \tilde{A}_{11} Q_j)^{-1} Q_j^T. \tag{3.15}$$

The final matrix T is of size $m \times m$.

By the construction of T_j (see above), $T_j \tilde{A}_{11}$ are orthogonal projections onto the space W_j with respect to the scalar product induced by $\tilde{A}_{11}$. That is, it holds that $(T_j \tilde{A}_{11} v, w)_{\tilde{A}_{11}} = (v, w)_{\tilde{A}_{11}}$ for all $w \in W_j$. Because of this, we have that

$$\begin{aligned}(u, T\tilde{A}_{11} w)_{\tilde{A}_{11}} &= \sum_{j=0}^{N} (u, T_j \tilde{A}_{11} w)_{\tilde{A}_{11}} = \sum_{j=0}^{N} (T_j \tilde{A}_{11} u, T_j \tilde{A}_{11} w)_{\tilde{A}_{11}} = \\ &= \sum_{j=0}^{N} (T_j \tilde{A}_{11} u, w)_{\tilde{A}_{11}} = (T\tilde{A}_{11} u, w)_{\tilde{A}_{11}},\end{aligned}$$

which shows that $T\tilde{A}_{11}$ is $\tilde{A}_{11}$-symmetric. It is also positive definite with respect to $\tilde{A}_{11}$ since $(T\tilde{A}_{11} v, T\tilde{A}_{11} v)_{\tilde{A}_{11}} = \|T\tilde{A}_{11} v\|^2_{\tilde{A}_{11}} \geq 0$ with equality if and only if $v = 0$. Thus, this preconditioner fulfils the requirement of being symmetric and positive definite with respect to $\tilde{A}_{11}$, which was required for the PCG method.

With a few assumptions on the underlying network of interfaces, PCG with preconditioner T does indeed converge. One can show that the eigenvalues λ of $T\tilde{A}_{11}$ fulfil

$$C_1(d, \overline{\alpha}, \underline{\alpha}, \overline{\beta}, \underline{\beta}) \leq \lambda(T\tilde{A}_{11}) \leq C_2(d, \overline{\alpha}, \underline{\alpha}, \overline{\beta}, \underline{\beta}),$$

where the constants C_1 and C_2 are independent of h and H, but C_1 additionally depends on the connectivity and density of the network of interfaces. This means that the condition number $\kappa(T\tilde{A}_{11})$ is independent of h and H and we thus gain a much better convergence as per equation (3.14) in comparison to using the CG-method without a preconditioner, where the condition number depends on h as described in section 2.2. For the full assumptions and proofs of convergence and eigenvalues, we refer to the appended article and to (Görtz et al., 2022).

When applying this preconditioner, one has to solve one coarse scale linear problem and n independent local problems. All of these problems can be solved using a direct solver since they are not very large, and thus applying this preconditioner for the PCG method results in a semi-iterative method.

4 Summary and future work

In this chapter, we provide a summary of Paper I, which is appended to this thesis. We describe the overall aims, findings and conclusions in order to facilitate the understanding of its contribution to the research field. Following this, we also discuss future work in this area.

4.1 Paper I: Well-Posedness and Finite Element Approximation of Mixed Dimensional Partial Differential Equations

We present a mixed-dimensional elliptic partial differential equation, which is given on a domain with a large number of interfaces. The bulk subdomains and the interfaces are both equipped with differential equations that are connected to each other via Robin boundary conditions, whilst the boundary of the full domain is equipped with homogeneous Dirichlet conditions. The PDE is given by equation (3.4). We show that the bilinear form is coercive and bounded, equations (3.5) – (3.6), and that the Lax-Milgram theorem thus guarantees the existence of a unique solution. The model is given in weak form but under the assumption that all bulk subdomains are convex, we have enough regularity to conclude that there is a corresponding strong form, which we present briefly, given by equations (3.2) – (3.3).

A fitted finite element method is presented in the paper, along with the a priori error estimate (3.9) that shows that the method actually converges.

The resulting linear system is split up into a bulk equation and an interface equation as described in equation (3.10). The bulk equation (3.12) is solved directly because of its block diagonal structure, which reduces the time and memory consumption for a direct solver. The interface equation (3.13) is a

heavier linear system, though, and may be too large for a direct method to handle. We have chosen to implement the preconditioned gradient method with the preconditioner based on domain decomposition that is described in section 3.3. We show that the PCG method with this preconditioner actually converges, whereafter we present some numerical examples.

The numerical examples are split into two parts with different purposes – first, we investigate the convergence of the fitted finite element method with regards to the mesh size h. Here, we do not separate the bulk and interface equations, but instead solely use a direct solver, since we have two relatively small test problems. The main focus lies on reducing h and seeing whether the anticipated rate of convergence is met. In the case of infinite interfaces, one obtains convex bulk subdomains, where we predict a linear convergence rate. The numerical example shows that the expected rate of convergence is met. In the case of finite interfaces, where one obtains non-convex subdomains, the expected rate of convergence is $h^{1/2}$. The obtained rate of convergence in the numerical example exceeds this, and is thus better than the estimation.

The second part of the numerical examples is the part with a large number of interfaces and implementation of the iterative method. Here, the main focus is to analyse how many iterations the PCG method needs before converging, and to see how the different parameters A_j and B_j affect this. The number of iterations before applying the preconditioner lie around 5000 to 10000. Using the preconditioner, the number of iterations reduce significantly – in our test cases the number of iterations vary between 24 and 87.

The parameter B_j describes the coupling between the bulk areas and the interfaces, with a higher B_j representing a stronger coupling. The numerical examples show that a stronger coupling requires more iterations.

The parameter A_j is either a constant 1 or piecewise constant on the different grid subintervals, uniformly distributed between 0.01 and 1. The system requires more iterations in the case of the uniformly distributed A_j.

4.2 Future work

There are several natural extensions to the research in the appended paper. In the domains we have examined so far, we do not allow slits. That is, at the end of every interface, there must be another interface that takes over. One natural continuation of our research is thus to examine the case where slits are allowed.

Another planned research topic is to use the *super localised orthogonal decomposi-*

tion (SLOD) method in order to solve the PDE problem (3.2)-(3.3). The SLOD method is based on the *localised orthogonal decomposition (LOD) method*, which is a variation of FEM that uses fine-scale correctors based on the problem. See (Målqvist and Peterseim, 2020). SLOD is a variation of the LOD method that converges even faster, see (Hauck and Peterseim, 2022).

Yet another extension is to alter the problem formulation itself. As it is now, we have examined elliptic problems, but it would also be interesting to take a look at parabolic problems posed on domains with a large number of interfaces.

Bibliography

Arrarás, A., Gaspar, F. J., Portero, L., and Rodrigo, C. (2019). Mixed-dimensional geometric multigrid methods for single-phase flow in fractured porous media. *SIAM Journal on Scientific Computing*, 41(5):B1082–B1114.

Asadzadeh, M. (2020). *An Introduction to the Finite Element Method for Differential Equations*. Wiley.

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

Brenner, S. and Scott, L. R. (2002). *The Mathematical Theory of Finite Element Methods*. Texts in Applied Mathematics. Springer New York.

Burman, E., Claus, S., Hansbo, P., Larson, M. G., and Massing, A. (2015). CutFEM: discretizing geometry and partial differential equations. *Internat. J. Numer. Methods Engrg.*, 104(7):472–501.

Burman, E., Hansbo, P., Larson, M. G., Larsson, K., and Massing, A. (2019). Finite element approximation of the Laplace–Beltrami operator on a surface with boundary. *Numerische Mathematik*, 141:141–172.

Demmel, J. W. (1997). *Applied Numerical Linear Algebra*. Society for Industrial and Applied Mathematics.

Dziuk, G. and Elliott, C. M. (2013). Finite element methods for surface PDEs. *Acta Numerica*, 22:289–396.

Fritz, M., Köppl, T., Oden, J. T., Wagner, A., Wohlmuth, B., and Wu, C. (2022). A 1d–0d–3d coupled model for simulating blood flow and transport processes in breast tissue. *International Journal for Numerical Methods in Biomedical Engineering*, 38(7):e3612.

Görtz, M., Hellman, F., and Målqvist, A. (2022). Iterative solution of spatial network models by subspace decomposition. Preprint. https://doi.org/10.48550/arXiv.2207.07488.

Greenbaum, A. (1997). *Iterative Methods for Solving Linear Systems*. Society for Industrial and Applied Mathematics.

Hansbo, A. and Hansbo, P. (2002). An unfitted finite element method, based on Nitsche's method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191(47):5537–5552.

Hauck, M. and Peterseim, D. (2022). Super-localization of elliptic multiscale problems. *Mathematics of Computation (MCOM)*.

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

Jaffre, J., Roberts, J. E., and Martin, V. (2006). Modeling fractures and barriers as interfaces for flow in porous media. *SIAM Journal of Scientific Computing*, 26(5):1667–1691.

Kornhuber, R. and Yserentant, H. (2016). Numerical homogenization of elliptic multiscale problems by subspace decomposition. *Multiscale Modeling and Simulation*, 14(3):1017–1036.

Larsson, S. and Thomée, V. (2009). *Partial Differential Equations with Numerical Methods*. Springer-Verlag Berlin, Heidelberg.

Målqvist, A. and Peterseim, D. (2020). *Numerical Homogenization by Localized Orthogonal Decomposition*. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Olshanskii, M. A., Reusken, A., and Grande, J. (2009). A finite element method for elliptic equations on surfaces. *SIAM Journal on Numerical Analysis*, (47):3339–3358.

Schwartz, P., Adalsteinsson, D., Colella, P., Arkin, A., and Onsum, M. (2005). Numerical computation of diffusion on a surface. *Proceedings of the National Academy of Sciences of the United States of America*, 102:11151–6.

Scott, L. R. and Zhang, S. (1990). Finite element interpolation of nonsmooth functions satisfying boundary conditions. *Mathematics of Computation*, 54:483–493.