8-2017

Discretizations & Efficient Linear Solvers for Problems Related to Fluid Flow

Ryan R. Grove
Clemson University, rgrove@g.clemson.edu

Discretizations & efficient linear solvers for problems related to fluid flow

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Mathematical Sciences

---

by
Ryan R. Grove
August 2017

Accepted by:
Dr. Timo Heister, Committee Chair
Dr. Leo Rebholz
Dr. Christopher Cox
Dr. Qingshan Chen

Abstract

Numerical solutions to fluid flow problems involve solving the linear systems arising from the discretization of the Stokes equation or a variant of it, which often have a saddle point structure and are difficult to solve. Geometric multigrid is a parallelizable method that can efficiently solve these linear systems especially for a large number of unknowns. We consider two approaches to solve these linear systems using geometric multigrid:

First, we use a block preconditioner and apply geometric multigrid as in inner solver to the velocity block only. We develop deal.II [6] tutorial step-56 [51] to compare the use of geometric multigrid to other popular alternatives. This method is found to be competitive in serial computations in terms of performance and memory usage.

Second, we design a special smoother to apply multigrid to the whole linear system. This smoother is analyzed as a Schwarz method using conforming and inf-sup stable discretization spaces. The resulting method is found to be competitive to a similar multigrid method using non-conforming finite elements that were studied by Kanschat and Mao [65]. This approach has the potential to be superior to the first approach.

Finally, expanding on the research done by Dannberg and Heister [32], we explore the analysis of a three-field Stokes formulation that is used to describe melt migra-

tion in the earth's mantle. Multiple discretizations were studied to find the best one to use in the ASPECT [12] software package. We also explore improvements to ASPECT's linear solvers for this formulation utilizing block preconditioners and algebraic multigrid.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Fluid dynamics is a broad and important field encompassing the study of natural convection within the mantle, the ocean, and the way in which air flows around the wings of a plane. The study of computational fluid mechanics is a topic of interest for engineers and mathematicians, since numerical simulation of fluid flow is a critical task in many applications within the industrial sector. With the need for numerical computations comes the need for numerical analysis, which gives a mathematical foundation that provides a way to know if computations are correct as well as insight on how to improve algorithms.

1.1 The Stokes Equation

The Stokes equation, an equation of top importance within the field of fluid mechanics, describes a creeping flow and is a prototype for many fluid dynamic computations. Let $\Omega \in \mathbb{R}^d$ be a bounded, connected domain (with dimension $d = 2, 3$) with smooth, piecewise boundary $\partial\Omega$. As in Benzi et al. [16], given a force $\mathbf{f} : \Omega \to \mathbb{R}^d$, we solve

for a velocity $\mathbf{u} : \Omega \to \mathbb{R}^d$ and a pressure $p : \Omega \to \mathbb{R}$ where

$$
\begin{aligned}
-\eta\Delta\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega, && (1.1)\\
\nabla \cdot \mathbf{u} &= 0 \text{ in } \Omega, && (1.2)\\
\mathbf{u} &= \mathbf{0} \text{ on } \partial\Omega, && (1.3)
\end{aligned}
$$

with viscosity $\eta > 0$. Physically, η can be thought of as frictional force that measures diffusion of momentum. It is caused by the molecular nature of fluid which creates resistance to shearing motions, thus taking flow's kinetic energy and converting it into heat [37]. Often for simulation of more complex flows such as Navier-Stokes equations, solving the Stokes problem is an important subproblem. This thesis aims to understand, analyze and develop efficient solvers for Stokes and is a potential stepping stone to having major impacts on numerical fluid dynamics computations.

1.2 A Saddle Point System

While using the finite element method, if you discretize and number your unknowns in a suitable way, then the discretization of the Stokes equation creates a saddle point system of the form

$$
\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix},
$$

which is what we obtain, as is obtained in Benzi et al. [16], for special cases of the discretized Stokes system (such as when we use Taylor Hood finite elements) [89].

The solution of saddle point systems of equations can require a large amount of time

to compute [56]. The size of the systems in realistic computations can become large enough that applying generic solvers for linear systems is inefficient, and to make it possible to solve this saddle point problem quickly, we need preconditioners that take advantage of our system's properties and structure [56].

In addition to computational fluid dynamics, saddle point problems come up in a variety of fields. For example, in mathematics, they also appear in linear optimization problems. In economics, it can be seen in solving inter-regional input-output systems [38]. It also makes an appearance in the elastic analysis and structural dynamics to determine internal forces, leading to the resulting stresses, strains, and displacements of a finite element model of a structure and a set of external loads in the area of electrical circuits and networks [17]. It can also be seen in finance [77], image reconstruction [54] and registration [53], along with many other areas.

1.3 Derivation of Stokes Equation

The Navier-Stokes equations describe the flow of a fluid that is both Newtonian and incompressible. As in Benzi et al. [16], given a force $\mathbf{f} : \Omega \to \mathbb{R}^d$, we seek a velocity $\mathbf{u} : \Omega \to \mathbb{R}^d$ and a pressure $p : \Omega \to \mathbb{R}$ such that

$$
\begin{aligned}
-\eta\Delta\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega, && (1.4)\\
\nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega, && (1.5)\\
\mathbf{u} &= \mathbf{0} \text{ on } \partial\Omega, && (1.6)
\end{aligned}
$$

where $\eta > 0$ is inversely proportional to Reynolds number Re as follows:

$$Re = \frac{\rho_{\text{ref}} L_{\text{ref}} U_{\text{ref}}}{\eta},$$

where ρ_{ref}, U_{ref}, and L_{ref}, are the reference density, velocity, and length from the nondimensionalization, respectively [56]. Re can be thought of as a dimensionless ratio of the driving from the boundary to the dampening from η [37]. When Re is small, this implies that constrained flows are present [37].

Mathematically, the condition of incompressibility is stated as $\nabla \cdot \mathbf{u} = 0$ and it is significant simplification that comes with a cost [37]. Physically, this constraint restricts the application of our model to problems where all of the relevant velocities occurring in the fluid are much less than the speed of sound (up to about 220 miles per hour) [37]. In terms of classical physics, the conservation of linear momentum is seen in Equation 1.4, while the conservation of mass, which is also sometimes called the incompressibility condition, is seen in Equation 1.5. To determine a unique pressure p, we also impose

$$\int_{\Omega} p \, \mathrm{dx} = 0. \tag{1.7}$$

In Equation 1.4 lies $(\mathbf{u}\cdot\nabla)\mathbf{u}$ which makes the Navier-Stokes system non-linear, but one popular approach of linearization is based on Picard's iteration [42], which is outlined in Benzi et al. [16] with existence and uniqueness proofs in Girault & Raviart [50] and one can find a proof for convergence of Picard's iteration in Karakashian [66].

As outlined in Benzi et al. [16], in a Picard iteration, there is an Oseen problem that

needs solved:

$$
\begin{aligned}
-\eta\Delta\mathbf{u} + (\mathbf{b}\cdot\nabla)\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega, && (1.8)\\
\nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega, && (1.9)\\
\mathbf{u} &= \mathbf{0} \text{ on } \partial\Omega, && (1.10)
\end{aligned}
$$

where $\mathbf{b}$ is known and divergence-free [16]. Discretization of Equations (1.8) through (1.10) using finite elements (see Elman et al. [42] and Quarteroni & Valli [89]) results in a saddle point system like in Section 1.2 [16, 89].

Our interest, for this thesis, is the case where $\mathbf{b} = \mathbf{0}$ which yields

$$
\begin{aligned}
-\eta\Delta\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega, && (1.11)\\
\nabla\cdot\mathbf{u} &= 0 \text{ in } \Omega, && (1.12)\\
\mathbf{u} &= \mathbf{0} \text{ on } \partial\Omega, && (1.13)
\end{aligned}
$$

which are collectively known as the Stokes equation. These equations describe a creeping flow which typically occurs in three settings [74]: small geometries such as in human capillaries, fluid flow moving through small pores, such as in filtration, and small velocities and large viscosities, such as in lubricant flow or mantle convection. The Stokes system is an important stepping stone for more complicated problems.

1.4 Motivation for Parallel Computing

The size of the systems that we are interested in within the research and industry sectors grows each year and computations with 100+ million unknowns are not un-

common. Some reasons for this are: complicated geometries that require resolving small features, turbulent flows that require resolving fine turbulent structures in the fluid, and large domains (such as in mantle convection). Problem sizes will be continuing to grow, and thus the memory requirements for solving these problems will also be increasing.

1.5 Introduction to Parallel Computing

An efficient way to solve these challenging problems is to use parallel computing which means our algorithms must also be designed to work efficiently in parallel. Parallel computing can help us by dividing the problem into multiple communicating subproblems capable of independently solving the problem on different processors. Due to the size and properties of the problems we are interested in solving, we turn to multigrid methods for preconditioners for our iterative solvers since they often scale linearly with respect to the number of unknowns [97, 56]. This is a significant advantage over other methods and becomes increasingly important as the number of unknowns becomes very large.

An example of this lies in our interest of geodynamics, particularly with the focus of melt migration in mantle convection. Without high-performance, massively parallel implementation, it would simply take too much time to create "high-resolution, 3-D, compressible, global mantle convection simulations coupled with melt migration" as done in Dannberg and Heister [32]. The existence of parallel computing allows research to be done more effectively in this field, as well as many others.

1.6 Highlights of Thesis Contributions

A central idea of this work is to provide a stepping stone to revolutionize the way people solve Stokes for large problems. To do this, we will be applying geometric multigrid as a preconditioner to the linear system resulting from discretizing the Stokes equation. We first apply this preconditioner on the velocity block, which is just a vector valued Laplace operator, and then we compare it to the approach where we apply it on the entire system matrix. For the latter we use additive Schwarz smoothers [44], which can be used in the multigrid to create an efficient and easily parallelizeable code.

In Chapter 3, we compare popular choices of preconditioners for the velocity block of the Stokes equation to geometric multigrid (GMG) in terms of performance and memory usage. The former includes UMFPACK [34], ILU, and algebraic multigrid (AMG) [93]. This has been partially investigated, but to our knowledge no code is available for the general public that implements GMG for the velocity block of Stokes so there is room for more detailed comparisons. We address this issue, so others can use our code as a template or starting point for their own research. The goal is to show that GMG is at least competitive in serial computations, because this will imply that it will outperform the other methods (especially UMFPACK and ILU) as our systems grow larger as well as in parallel computations.

Subtle choices can highly affect algorithm design results, time, and usability such as the choice of finite element or preconditioner. This, combined with our goal of solving Stokes as quickly and efficiently as possible, led us to extend the work of Kanschat and Mao [65] to include conforming inf-sup stable finite elements in Chapter 4. That is, we used their idea of applying GMG not just to the velocity block, as we do in

Chapter 3, but the entire system matrix instead and extended this idea to work with conforming inf-sup stable finite elements. Our aim is to achieve results that are much better than those from Chapter 3 in terms of iteration counts.

Finally, we consider an application in geophysics that requires the solution of a related three-field Stokes equation. The goal is to provide a theoretical foundation and extend the common preconditioning approaches for Stokes to this problem. Following the work done by Dannberg and Heister [32], given force $\mathbf{f} : \Omega \to \mathbb{R}^d$ and $\mathbf{g} : \Omega \to \mathbb{R}^d$, we seek a velocity $\mathbf{u} : \Omega \to \mathbb{R}^d$, a fluid pressure $p_f : \Omega \to \mathbb{R}$, and a compaction pressure $p_c : \Omega \to \mathbb{R}$ such that

$$
\begin{aligned}
-\nabla \cdot (\eta \nabla \mathbf{u}) + \nabla p_f + \nabla p_c &= \mathbf{f}, && (1.14)\\
\nabla \cdot \mathbf{u} - \nabla \cdot (k_D \nabla p_f) &= \mathbf{g}, && (1.15)\\
\nabla \cdot \mathbf{u} + \frac{1}{\epsilon} p_c &= 0. && (1.16)
\end{aligned}
$$

where $\eta > 0$ is the shear viscosity, $k_D \geq 0$ is the Darcy coefficient [33], and $\frac{1}{\epsilon} > 0$, where ϵ is the bulk viscosity. Parallel computing is required for mantle convection with melt migration due to the need of high resolution, higher dimension simulations [32], and this requires stable discretizations and efficient preconditioners that can be run in parallel.

In Chapter 5, we explore the analysis of three-field Stokes equation and try to improve existing solvers used in current competitive geoscience codes. Scientists in geoscience have seemingly been using this formulation without a complete mathematical understanding, since, no complete analysis or discussions of its discretization have been published. We investigate extending the solvers developed in earlier chapters of this thesis to the three-field Stokes equation. There are numerous researchers in the geo-

science community that the results of this chapter directly impact, as simulating flows using the three-field Stokes equation is a fundamental necessity in their research.

1.7 Overview

In Chapter 2, we build up mathematical tools needed to discuss finite element method (FEM) for the Poisson equation and we then extend this theory to the Stokes equation, which is the key focus of this thesis. We also introduce here our methods for solving Stokes-type systems, including linear solvers and Krylov methods [73].

In Chapter 3, we briefly introduce preconditioners and geometric multigrid (GMG) before explaining the deal.II tutorial step-56 [51] where we use GMG as a preconditioner on the velocity block to create an efficient linear solver for the Stokes equation and compare it to alternative approaches. In Chapter 4, we apply multigrid method as a preconditioner to the whole system instead of just the velocity block. In Chapter 5, we explore melt migration, which is an aspect of mantle convection described by the three-field Stokes equation. In Chapter 6, we make some concluding remarks.

Chapter 2

Mathematical Foundations

This chapter serves as a thorough mathematical foundation for this thesis. We build up mathematical tools needed to discuss finite element method for the Poisson equation, before applying these same ideas to the Stokes equation. To start, consider the bounded domain Ω with edge $\partial\Omega$ in $\mathbb{R}^d$, where $d = 2, 3$. Physically, this space Ω is the space in which our fluid will reside.

2.1 Function Spaces

The conservation laws of mass and momentum are the rules which fluid dynamics are bound [111]. When we take a step back and look at fluid flow on the "big scale", it would appear to an observer that the local differences in velocity are exerting a force upon the adjacent fluid, which alters flow as well as dissipating energy [74]. The first natural function space is the space of all velocity fields with finite total kinetic energy, the Hilbert space $L^2(\Omega)$, which is the space of functions that are "square integrable"

[74, 63]. We define $L^2(\Omega)$ on a scalar function $q : \Omega \to \mathbb{R}$ as

$$L^2(\Omega) = \left\{ q : \Omega \to \mathbb{R} | \int_\Omega |q|^2 \, dx < \infty \right\}.$$

For vector functions, $\mathbf{v} : \Omega \to \mathbb{R}^d, \mathbf{v} \in L^2(\Omega)^d$ if its components are in $L^2(\Omega)$, and we will write this as simply $\mathbf{v} \in L^2(\Omega)$. We have yet to show the physical importance of this $L^2(\Omega)$ space we have defined, namely that it can be seen as the set of all velocity fields with finite kinetic energy K. To see this, let ρ be constant density and $\mathbf{u}$ a velocity field, then

$$K = \frac{1}{2} \text{ mass} \times \text{velocity}^2 = \frac{1}{2}\rho \int_\Omega |\mathbf{u}|^2 \, dx,$$

as seen in Layton [74] and further explored in Doering & Gibbon [37]. We define $L^2(\Omega)$ norm for a scalar function $q : \Omega \to \mathbb{R}$ as $||q|| := (\int_\Omega |q|^2 \, dx)^{\frac{1}{2}}$. The norm $|| \cdot ||$ will further always denote the $L^2(\Omega)$ norm (other norms will have subscripts). For a vector $\mathbf{v} : \Omega \to \mathbb{R}^d$, the $L^2(\Omega)$ norm is written as

$$||\mathbf{v}|| = \int_\Omega \left(|\mathbf{v}|^2 \, dx \right)^{\frac{1}{2}},$$

where $\mathbf{v} = (v_1, \ldots, v_d)$ and $| \cdot |$ is the Euclidean norm. Physically, if there are no outside forces, then the $L^2(\Omega)$ norm being preserved is equivalent to the physical property of the total kinetic energy being conserved [37]. Furthermore, the previous ideas can be generalized for the $L^p(\Omega)$ function space, where $0 < p < \infty$, as

$$L^p(\Omega) = \left\{ x : \Omega \to \mathbb{R} \middle| \int_\Omega |x|^p \, dx < \infty \right\},$$

as seen in Layton [74], with norm

$$||\mathbf{v}||_p = (\int_\Omega |\mathrm{v}|^p \; dx)^{\frac{1}{p}}.$$

For our analysis, we will also need to define inner products in our function spaces. The $L^2(\Omega)$ inner products are defined, as seen in Layton [74], as:

$$\begin{aligned}(p,q) &:= \int_\Omega p(x)q(x) \; dx \text{ for } p,q \in L^2(\Omega), \\ (\mathbf{u},\mathbf{v}) &:= \int_\Omega \mathbf{u}(x)\mathbf{v}(x) \; dx \text{ for } \mathbf{u},\mathbf{v} \in L^2(\Omega)^d,\end{aligned}$$

and let subspace $L_0^2(\Omega) \subseteq L^2(\Omega)$ be defined, as seen in Layton [74], as

$$L_0^2(\Omega) = \left\{ q : \Omega \to \mathbb{R}, q \in L^2(\Omega) \middle| \int_\Omega q \; dx = 0 \right\}.$$

Complex patterns in fluids are created by large local changes in velocity (the first derivatives of $\mathbf{u}$) which cause a part of the fluid to exert forces or drags on adjacent parts of the fluid [74]. This fluid must then move out of the way of other parts of the fluid and the force required to do so must be finite. Thus, if the velocity is to be physically relevant, its gradient must be in $L_2(\Omega)^{d\times d}$ [74]. Therefore, we define

$$\begin{aligned}H^1(\Omega) &= \left\{ \mathbf{v} : \Omega \to \mathbb{R}^d \middle| \mathbf{v} \in L^2(\Omega)^d, \nabla\mathbf{v} \in L^2(\Omega)^{d\times d} \right\}, \\ H_0^1(\Omega) &= \left\{ \mathbf{v} : \Omega \to \mathbb{R}^d \middle| \mathbf{v} \in H^1(\Omega), \mathbf{v} = 0 \text{ on } \partial\Omega \right\},\end{aligned}$$

where $\nabla\mathbf{v}$ is the Jacobian of $\mathbf{v}$.

A Sobolov space $W^{m,p}(\Omega)$ has derivatives of order up to m in $L^p(\Omega)$, with integer m

and $1 \leq p \leq \infty$ [107, 78], and norm

$$||\mathbf{u}||_{W^{m,p}(\Omega)} = \left(\sum_{|j| \leq m} ||\nabla^j \mathbf{u}||^p_{L^p(\Omega)} \right)^{\frac{1}{p}}.$$

When $p = 2$, then $W^{m,2}(\Omega) = H^m(\Omega)$ with

$$(\mathbf{u}, \mathbf{v})_{H^m(\Omega)} = \sum_{|j| \leq m} (\nabla^j \mathbf{u}, \nabla^j \mathbf{v})$$

and thus $H^1(\Omega)$ is a Sobolov space and has the following inner product and norm definitions:

$$\begin{aligned} (\mathbf{u}, \mathbf{v})_{H^1} &= (\mathbf{u}, \mathbf{v}) + (\nabla \mathbf{u}, \nabla \mathbf{v}) \\ ||\mathbf{u}||_{H^1} &= (||\mathbf{u}||^2 + ||\nabla \mathbf{u}||^2)^{\frac{1}{2}}. \end{aligned}$$

There also exists the H^k norm and semi-norm, which, respectively, are

$$\begin{aligned} ||\mathbf{u}||_k &= (||\mathbf{u}||_{k-1} + \sum_{|j|=k} ||\nabla^j \mathbf{u}||^2)^{\frac{1}{2}}, \\ |\mathbf{u}|_k &= ||\nabla^k \mathbf{u}||. \end{aligned}$$

2.2 The Poisson Equation

Given a force $\mathbf{f} : \Omega \to \mathbb{R}^d$, we seek a solution $\mathbf{u} : \Omega \to \mathbb{R}^d$ of

$$-\Delta \mathbf{u} = \mathbf{f} \text{ in } \Omega, \tag{2.1}$$

$$\mathbf{u} = \mathbf{0} \text{ on } \partial\Omega, \tag{2.2}$$

where $\mathbf{f} \in L^2(\Omega)$. We require that $\mathbf{u}$ vanishes on the boundary of the domain because we want our boundary to represent fixed walls. Physically, when $\mathbf{u}$ is a velocity, the microscopic interactions occurring between the fluid and the wall are at least as strong as those between different parts of the fluid themselves, so the velocity vector field should be continuous at the wall [37].

2.3 Triangulation

Before we move into the discrete case, we need to decompose Ω. We want our triangulation $\mathcal{T}_h \subset \mathbb{R}^d$ to be conforming (squares need to be edge to edge), non-degenerate (the minimum angle of the square must be sufficiently large), and the boundary of the computational domain needs to be within the targeted error of the boundary of the real domain [74].

A triangulation Ω_h of Ω is made by subdividing Ω into a set $\mathcal{T}_h = \{Q_1, ..., Q_m\}$ of m non-overlapping quadrilateral cells Q_i in two and three dimensions, respectively, such that

$$\Omega_h = \bigcap_{Q_i \in \mathcal{T}_h} \overline{Q_i} = \overline{Q_1} \cap ... \cap \overline{Q_m},$$

and we define the mesh parameter (or mesh size [95]) to be

$$h = \max_{i=1,...,m} \mathrm{D}(Q_i),$$

where $\mathrm{D}(Q_i)$ is the cell Q_i's diameter.

2.4 Finite Element Spaces

An introduction of various mesh generation algorithms can be found in Ern & Guermond (see [43]) but all of our grid generation is done using deal.II's GridGenerator class [6].

Let $\mathcal{Q}_p$ be the Lagrange FE space with order p on the reference cell $\hat{K} = [0,1]^d$ as described by Heister [56]. For each element $K \in \mathcal{T}_h$, we define the bilinear mapping from the reference cell $\hat{K}$ to the cell K as $F_K : \hat{K} \to K$ [56]. We let $p = 1, 2, \ldots$ and define the space

$$Q_p := \{\mathbf{v} \in C(\Omega) | \mathbf{v}|_K \circ F_K \in \mathcal{Q}_p, K \in \mathcal{T}_h\}.$$

Continuous along the boundaries of each cell, a finite element function is defined to be the image of a polynomial function on $\hat{K}$ on each cell K as in Heister [56].

2.5 Finite Element Method for the Poisson Equation

In this section we seek to obtain convergence and estimates of the error, which show asymptotic convergence as $h \to 0$ after we bound the approximate solution in a physically relevant norm (by the problem data) [74].

The FEM is a numerical method used for solving partial differential equations (PDEs) in engineering and science and is well-known in for finding numerical solutions of differential and integral equations in the fields of math and engineering [63]. It was introduced in the late 1950s and early 1960s by engineers interested in numerical solutions of partial differential equations for structural engineering, where they had

structures subdivided into many small parts with simple behavior, which they named finite elements [63]. Its main competitors are the finite difference method, which is good for "simple" problems on "simple" geometries (see Wesseling [111]), and spectral methods [25]; neither of which will be discussed here. FEM is particularly advantageous for complex geometries, more complicated PDEs, and problems with variable or non-linear material properties [63]. Its solid mathematical foundation offers reliability and in many cases makes analyzing and estimating error in approximate solutions possible. In other methods, obtaining an estimate of such errors can be much harder [63].

Recalling our Poisson problem (2.1)-(2.2), the first step for finite element method is creating a variational formulation. This variational formulation is one reason that the FEM is robust; it is fundamentally different from the classical numerical methods for partial differential equations [74]. In the finite difference method, one replaces all derivatives with difference quotients that rely on unknown values at a finite number of points, to get your discrete problem [63].

To reformulate our differential equation as an equivalent variational problem, we first multiply both sides "by a test function $\mathbf{v} \in C_0^\infty(\Omega)^d$ and integrate over the domain" as in Strang and Fix [104], which yields

$$(-\nabla \cdot (\nabla \mathbf{u}), \mathbf{v}) = (\mathbf{f}, \mathbf{v}).$$

Then, we can use Green's Theorem, which is derived from the Divergence Theorem in Johnson [63], to get

$$(\nabla \mathbf{u}, \nabla \mathbf{v}) - \int_{\partial\Omega} (\nabla \mathbf{u} \cdot \mathbf{n}) \mathbf{v} dS = (\mathbf{f}, \mathbf{v}),$$

where $\mathbf{n}$ is the unit outward normal, and $\int_{\partial\Omega}(\nabla\mathbf{u}\cdot\mathbf{n})\mathbf{v}dS = 0$ since $\mathbf{v} = \mathbf{0}$ on $\partial\Omega$. Therfore, we now have: find $\mathbf{u} \in X := H_0^1(\Omega)$ such that

$$(\nabla\mathbf{u}, \nabla\mathbf{v}) = (\mathbf{f}, \mathbf{v}) \ \forall\mathbf{v} \in X \tag{2.3}$$

as our variational or weak form. Equation (2.3) is called a weak formulation of Equations (2.1) and (2.2) and the solution of Equation (2.3) is called a weak solution of Equations (2.1) and (2.2) [63]. It is important to remember that if $\mathbf{u}$ is a weak solution to Equations (2.1) and (2.2), it is not necessarily also a classical solution unless $\mathbf{u}$ is "sufficiently regular", where we refer to a solution being "sufficiently regular" if $\Delta\mathbf{u}$ is defined in a classical sense. However, the weak formulation has the mathematical advantage of it being relatively simple to prove existence in the weak formulation [63].

We want to find $\mathbf{u} \in X := H_0^1(\Omega)$ (note: $||\phi||_X = ||\nabla\phi||$ is a norm on $H_0^1(\Omega)$ [74]) such that $(\nabla\mathbf{u}, \nabla\mathbf{v}) = (\mathbf{f}, \mathbf{v})$ for all $\mathbf{v} \in X$. To do this, we first define the map $a(\mathbf{u}, \mathbf{v}) : X \times X \to \mathbb{R}$ to be $(\nabla\mathbf{u}, \nabla\mathbf{v})$ and note that it is bilinear, symmetric, and bounded [74, 56].

Theorem: Lax-Milgram [74]

Let $a(\mathbf{u}, \mathbf{v}) : X \times X \to \mathbb{R}$ be a bilinear form satisfying continuity

$$|a(\mathbf{u}, \mathbf{v})| \leq c_1||\mathbf{u}||_X||\mathbf{v}||_X \ \forall\mathbf{u}, \mathbf{v} \in X$$

and coercivity

$$a(\mathbf{u}, \mathbf{u}) \geq c_2||\mathbf{u}||_X \ \forall\mathbf{u} \in X$$

and let $l : X \to \mathbb{R}$ be a linear functional which is continuous as in $l(\mathbf{v}) \leq c_3||\mathbf{v}||_X \ \forall \mathbf{v} \in X$. Then there exists a unique $\mathbf{u} \in X$ s.t $a(\mathbf{u}, \mathbf{v}) = l(\mathbf{v}) \ \forall \mathbf{v} \in X$, furthermore $||\mathbf{u}||_X \leq \frac{c_3}{c_2}$. A proof of this theorem can be found in Axelsson & Barker [7]. For our problem, $l(\mathbf{v}) = (\mathbf{f}, \mathbf{v})$ is linear. So we check the continuity first:

$$a(\mathbf{u}, \mathbf{v}) = (\nabla \mathbf{u}, \nabla \mathbf{v}) \leq ||\nabla \mathbf{u}||||\nabla \mathbf{v}|| = ||\mathbf{u}||_X||\mathbf{v}||_X,$$

where the first inequality comes from the Cauchy-Schwarz inequality which is defined as $(\mathbf{u}, \mathbf{v}) \leq ||\mathbf{u}||||\mathbf{v}|| \ \forall \mathbf{u}, \mathbf{v} \in L^2(\Omega)$ [74], and thus we have continuity with $c_1 = 1$. Now we check coercivity:

$$a(\mathbf{u}, \mathbf{u}) = (\nabla \mathbf{u}, \nabla \mathbf{u}) = ||\nabla \mathbf{u}||||\nabla \mathbf{u}|| = ||\mathbf{u}||_X^2,$$

and thus we have coercivity of a with $c_2 = 1$. Now we check continuity of l:

$$l(\mathbf{v}) = (\mathbf{f}, \mathbf{v}) \leq ||\mathbf{f}||||\mathbf{v}|| \leq ||\mathbf{u}||_X C_{PF}||\nabla \mathbf{v}|| = C_{PF}||\mathbf{f}||||\mathbf{v}||_X,$$

where $C_{PF} > 0$, by Cauchy-Schwarz inequality since $f \in L^2(\Omega)$, and thus we have continuity of l where $c_3 = C_{PF}||f||$. Here we used the Poincare-Friedrichs inequality which states if you let a function space X be defined as $X := H_0^1(\Omega)$, then there exists a positive constant C_{PF} such that $||\mathbf{v}|| \leq C_{PF}||\nabla \mathbf{v}|| \ \forall \mathbf{v} \in X$, the proof of which can be found in Layton [74]. This is said to hold as long as Ω is bounded in some direction as discussed in Temam [107].

Thus, Lax-Milgram is satisfied and a unique solution to (2.3) exists. And since $\mathbf{u} \in X$ exists, we can choose $\mathbf{v} = \mathbf{u}$ to obtain

$$||\nabla \mathbf{u}||^2 = (\mathbf{f}, \mathbf{u}) \leq ||\mathbf{f}||C_{PF}||\nabla \mathbf{u}||,$$

which implies $||\nabla \mathbf{u}|| \leq C_{PF}||\mathbf{f}||$.

The next step of the FEM is to combine the Galerkin approximation and a good choice for finite dimensional space X_h [74]. The Galerkin method begins when we pick a finite dimensional subspace $X_h \subseteq X$ on Ω_h that vanishes on the boundary $\partial\Omega$ [95]. Then we want to find a Galerkin approximation $\mathbf{u}_h \in X_h$ such that $a(\mathbf{u}_h, \mathbf{v}_h) = F(\mathbf{v}_h)$ for all $\mathbf{v}_h \in X_h$ with the hope that u_h is an acceptable approximation to $\mathbf{u}$.

We choose values at node points of $\mathcal{T}_h$ to describe the functions in X_h which will be called the global degrees of freedom (dofs), as explained in Johnson [63]. We define $\mathbf{u}_h(\mathbf{x}) \in X_h$ to be the nodal interpolation of $\mathbf{u}(\mathbf{x})$ which is created by using a set of basis functions $\left\{\boldsymbol{\phi}_j\right\}_1^n$ (also known as trial functions [104]) where

$$\mathbf{u}_h(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \boldsymbol{\phi}_j(\mathbf{x}),$$

and thus each function of X_h can be written in this way [95], where undetermined coefficients α_j are just point values of $\mathbf{u}_h$. Since X_h is a closed subspace of the Hilbert space X, X_h is also a Hilbert space. Since $||\cdot||_{X_h} = ||\cdot||_X$, we can use the Lax-Milgram theorem for this discrete case with the exact same steps as the continuous case, that the problem of finding $\mathbf{u}_h \in X_h$ satisfying $a(\mathbf{u}_h, \mathbf{v}_h) = F(\mathbf{v}_h)$ for all $\mathbf{v}_h \in X_h$ is well-posed. There is a complete convergence theory for the Galerkin method thanks to Cea's lemma [74], which states that if $a(\cdot,\cdot) : X \times X \to \mathbb{R}$ is a continuous, coercive, bilinear form, $F : X \to \mathbb{R}$ is a bounded (continuous) linear functional, and $X_h \leq X$ be finite dimensional, then if $\mathbf{u} \in X$ solves: $a(\mathbf{u}, \mathbf{v}) = F(\mathbf{v})\ \forall \mathbf{v} \in X$ and $\mathbf{u}_h \in X_h$ solves $a(\mathbf{u}_h, \mathbf{v}_h) = F(\mathbf{v}_h)\ \forall \mathbf{v}_h \in X_h$, then

$$||\nabla(\mathbf{u} - \mathbf{u}_h)|| < (1 + \frac{c_1}{c_2}) \inf_{\mathbf{v}_h \in X_h} ||\nabla(\mathbf{u} - \mathbf{v}_h)||,$$

of which a proof can be found in Layton [74]. This convergence theory is critical because we need to understand and analyze how well $\mathbf{u}_h$ approximates $\mathbf{u}$ [97].

An alternate way to view $\mathbf{u}_h$ provided in Johnson [63] is to view $\mathbf{u}_h$ as the projection with respect to our defined $H^1(\Omega)$ inner product of the exact solution $\mathbf{u}$ on X_h.

2.6 Formal Definition of a Finite Element

Now that we have defined many pieces of a finite element, we give a formal definition of a finite element as a triple $(\mathcal{T}_h, P_{\mathcal{T}_h}, \Sigma)$, where

- $\mathcal{T}_h$ is a triangulation,
- $P_{\mathcal{T}_h}$ is a finite space of continuous functions defined on $\mathcal{T}_h$,
- Σ is a set of linear forms that map from $P_{\mathcal{T}_h} \to \mathbb{R}$. The elements of Σ are called degrees of freedom [85].

such that a function $v \in P_Q$ is unique and defined from the values of Σ as mentioned in Johnson [63]. See the work done by Bangerth et al. [11] for more information on an approach for how to use the FEM in parallel.

2.7 The deal.II Finite Element Library

All numerical simulations in this thesis were performed using deal.II [6]. It is open source and widely used by people who use finite elements and has been cited in many projects on many continents [56]. deal.II is coded in C++ and uses template

programming to make it possible to write unique programs in two-dimensional but then run them in three-dimensional with little to no extra effort.

Linear algebra libraries (PETSc [9] and Trilinos [58]), solvers, input and output, as well as MPI-based parallelization [11] all have support within deal.II [56, 6]. This all comes with an extensive collection of documentation as well as many tutorial programs (including step-56 [51] which is highlighted in Chapter 3) that incrementally highlights various aspects of the library while explaining both the mathematics and the deal.II implementation in detail [56].

Using the definition of finite element space as well as triangulation that was chosen above, the degrees of freedom need a global numbering, which is done by deal.II's DoFHandler class [6]. deal.II will be used to solve our linear systems using specific solvers and preconditioners (to be discussed in more detail later). Last but not least, deal.II provides many post processing options, including data output and analysis, and error estimation of the solution for adaptive mesh refinement [56, 6].

2.8 Finite Element Method of the Stokes Problem

We start with the variational form of the Stokes problem and proceed to show well-posedness in both continuous and discrete cases, before finishing with velocity and pressure error bounds. This is based on Layton [74].

To derive the variational formulation of the Stokes problem, let $(\mathbf{u}, p)$ be the classical solution of the Stokes problem and multiply (1.11) and (1.12) by functions

$\mathbf{v} \in C_0^\infty(\Omega)^d$ and $q \in C_0^\infty(\Omega)$ and integrate:

$$\begin{aligned}(-\eta\Delta\mathbf{u}, \mathbf{v}) + (\nabla p, \mathbf{v}) &= (\mathbf{f}, \mathbf{v}) \\ (\nabla \cdot \mathbf{u}, q) &= 0\end{aligned}$$

We balance derivatives using Green's Theorem, as we did in Section 2.5, to get

$$\begin{aligned}(-\eta\Delta\mathbf{u}, \mathbf{v}) &= (\eta\nabla\mathbf{u}, \nabla\mathbf{v}) - \int_{\partial\Omega} (\nabla\mathbf{u} \cdot \mathbf{n}) \cdot \mathbf{v} dS, \\ (\nabla p, \mathbf{v}) &= -(p, \nabla \cdot \mathbf{v}) + \int_{\partial\Omega} p(\mathbf{v} \cdot \mathbf{n}) dS,\end{aligned}$$

and note that both boundary integrals vanish due to boundary conditions. Thus we have the following: Find $\mathbf{u} \in X = \boldsymbol{H}_0^1(\Omega)$, $p \in Y = L_0^2(\Omega)$ such that

$$(\eta\nabla\mathbf{u}, \nabla\mathbf{v}) - (p, \nabla \cdot \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \; \forall \mathbf{v} \in X \tag{2.4}$$

$$(\nabla \cdot \mathbf{u}, q) = 0 \; \forall q \in Y \tag{2.5}$$

as the weak form.

2.8.1 Continuous Well-posedness

A well-posed problem has a guaranteed existence of a solution and uniqueness of that solution. This is also based on Layton [74].

Theorem

The problem (2.4) - (2.5) is well-posed if $\mathbf{f} \in H^{-1}(\Omega)$.

Proof

Define the bilinear form $a : X \times X \to \mathbb{R}$ by

$$a(\mathbf{u}, \mathbf{v}) \quad = \quad (\eta \nabla \mathbf{u}, \nabla \mathbf{v})$$

and also the form $b : X \times Y \to \mathbb{R}$ by

$$b(\mathbf{v}, q) \quad = \quad -\int_{\Omega} (\nabla \cdot \mathbf{v}) q \; d\Omega$$

Thus, (2.4) and (2.5) can be written as

$$a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) \quad = \quad \mathbf{l}(\mathbf{v}) \tag{2.6}$$

$$b(\mathbf{u}, q) \quad = \quad m(q) \tag{2.7}$$

where $\mathbf{l} : X \to \mathbb{R}$ and $l(\mathbf{v}) = (\mathbf{f}, v)$ and $m(q) = 0$.

Therefore, appropriate discretization of the Stokes system, including picking Taylor Hood [106] elements and rearranging the order of the unknowns (described in deal.II tutorial step-56 [51]), yields a saddle point problem whose form was described in Section 1.2.

Let us define the space

$$V = \{\mathbf{v} \in X | b(\mathbf{v}, q) = 0 \;\; \forall q \in Y\},$$

and then we decompose $X = V \oplus V^{\perp}$ and $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_1$ where $\mathbf{u}_0 \in V$ and $\mathbf{u}_1 \in V^{\perp}$.

Now restrict $q \in Y$, therefore (2.7) becomes

$$\begin{aligned} b(\mathbf{u}_0 + \mathbf{u}_1, q) &= m(q) \\ \implies b(\mathbf{u}_0, q) + b(\mathbf{u}_1, q) &= m(q) \\ \implies b(\mathbf{u}_1, q) &= m(q) \end{aligned}$$

since, $b(\mathbf{u}_0, q) = 0 \ \forall q \in Y$. Note that when this equation is solved that we will have $\mathbf{u}_1$ uniquely determined since X is the direct sum of V and $V^\perp$ and we know that for all $\mathbf{x} \in X$, there exists a unique $\mathbf{x}_0$ in V and $\mathbf{x}_1$ in $V^\perp$ such that $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1$. Using the following inf-sup condition [74, 89, 23], we achieve this:

$$\inf_{0 \neq q \in Y} \sup_{0 \neq \mathbf{v} \in X} \frac{b(\mathbf{v}, q)}{||\mathbf{v}||_X ||q||_Y} \geq \alpha > 0 \tag{2.8}$$

where the $X-$norm is the standard H^1 norm and the $Y-$norm is the standard L^2 norm. Now that we have found $\mathbf{u}_1$, we focus on solutions living in V. We want to find $\mathbf{u}_0$ on V such that (2.6) holds. So now we have the new problem

$$a(\mathbf{u}, \mathbf{v}) = \mathbf{l}(\mathbf{v}) \tag{2.9}$$

where $\mathbf{v} \in V$ and we need to complete the conditions of Lax-Milgram. Our energy norm is

$$||\mathbf{u}||_V = \sqrt{(\eta \nabla \mathbf{u}, \nabla \mathbf{u})} \tag{2.10}$$

The left hand side of Equation (2.9) is

$$a\left(\mathbf{u}, \mathbf{v}\right) = (\eta \nabla \mathbf{u}, \nabla \mathbf{v})$$

Now we need to ensure this satisfies Lax-Milgram so we check, continuity, coercivity, and boundedness of the right hand side as we did for the Poisson problem:

Continuity

Before we do this let's bound our norm (2.10) as follows:

$$||\mathbf{u}||_V \geq ||\sqrt{\eta}\nabla\mathbf{u}|| \tag{2.11}$$

Now, applying Cauchy-Schwarz to $|a(\mathbf{u},\mathbf{v})|$ yields

$$\begin{aligned} |a(\mathbf{u},\mathbf{v})| &\leq ||\sqrt{\eta}\nabla\mathbf{u}||||\nabla v|| \\ &\leq ||\sqrt{\eta}\nabla\mathbf{u}||\frac{1}{\sqrt{\eta_{\min}}}||\sqrt{\eta}\nabla\mathbf{v}|| \\ &\leq \frac{1}{\sqrt{\eta_{\min}}}||\mathbf{u}||_V||\mathbf{v}||_V \end{aligned}$$

Therefore, we have that

$$|a(\mathbf{u},\mathbf{v})| \leq c_1||\mathbf{u}||_V||\mathbf{v}||_V \tag{2.12}$$

where $c_1 = \frac{1}{\sqrt{\eta_{\min}}}$, therefore showing continuity.

Coercivity

Now, looking at our energy norm once again and letting $\mathbf{u} = \mathbf{v}$ yields

$$|a(\mathbf{u},\mathbf{u})| \geq ||\sqrt{\eta}\nabla\mathbf{u}||^2$$

which, thanks to how we picked our norm, can be upper bounded by

$$|a(\mathbf{u},\mathbf{u})| \geq c_2||\mathbf{u}||_V^2 \tag{2.13}$$

where $c_2 = 1$, therefore showing continuity.

Boundedness of Right Hand Side

Adding the right hand sides of equation (2.6)

$$\begin{aligned} \mathbf{l}(\mathbf{v}) &\leq ||\mathbf{l}||_{H^{-1}}||\nabla\mathbf{v}|| \\ &= ||\mathbf{l}||_{H^{-1}}\frac{1}{\eta_{\min}}||\eta\nabla\mathbf{v}|| \\ &\leq \frac{1}{\eta_{\min}}||\mathbf{l}||_{H^{-1}}||\mathbf{v}||_V. \end{aligned}$$

Therefore, we have that

$$F(v) \leq c_3||\mathbf{v}||_V \tag{2.14}$$

where $c_3 = \frac{1}{\eta_{\min}}||\mathbf{l}||_{H^{-1}}$, therefore showing boundedness of the right hand side as long as $\mathbf{l} \in H^{-1}(\Omega)$. Since continuity, coercivity, and boundedness of the right hand side is achieved, Lax-Milgram holds, and then there exists a unique $\mathbf{u}_0 \in V$ s.t $a(\mathbf{u}_0,\mathbf{v}) = l(\mathbf{v})\ \forall\mathbf{v} \in V$, furthermore $||\mathbf{u}_0||_V \leq \frac{c_3}{c_2} = \frac{1}{\eta_{\min}}||\mathbf{l}||_{H^{-1}}$.

Finishing the proof

Now that $\mathbf{u}_0$ and $\mathbf{u}_1$ are known, we can again consider (2.6) as follows

$$\begin{aligned} b(\mathbf{v}, p) &= \mathbf{l}(\mathbf{v}) - a(\mathbf{u}_0 + \mathbf{u}_1, \mathbf{v}) \\ &= h(\mathbf{v}). \end{aligned}$$

Therefore, our new problem is to find $p \in Y$ such that

$$b(\mathbf{v}, p) = h(\mathbf{v}).$$

Note that when this equation is solved that we will have p uniquely determined. Using the following inf-sup condition, we achieve this:

$$\inf_{0 \neq \mathbf{v} \in X} \sup_{0 \neq q \in Y} \frac{b(\mathbf{v}, q)}{||\mathbf{v}||_X ||q||_Y} \geq \alpha_2 > 0. \tag{2.15}$$

Now p is uniquely determined. We now have $\mathbf{u}_0$, $\mathbf{u}_1$, and p existing and being uniquely determined, therefore we have well-posedness as long as $\mathbf{l} \in H^{-1}(\Omega)$, which is true if $\mathbf{f} \in H^{-1}(\Omega)$.

2.8.2 Discrete Well-posedness

Now, we need to prove the same for the discrete case. The FE discretization needs FE spaces for both variables, namely $V_h \subset V$ and $Q_h \subset Q$, and $X_h = V_h \times Q_h$. This is referred to as a mixed finite element method because we seek independent approximations of both velocity $\mathbf{u}$ and pressure p [63].

When the well-posedness of a problem can be studied using the Lax-Milgram theo-

rem, such as the Poisson problem, the well-posedness of any of its finite dimensional approximations can also be treated by the Lax-Milgram theorem thanks to Cea's Lemma [49]. Unfortunately, additional conditions are required for the discrete well-posedness proof for Stokes. Intuitively, if (asymptotically) Q_h is "too large" compared to V_h, then we have too many constraints on velocity or velocity does not have enough degrees of freedom and the discrete solution may not converge [10]. A lot of theory for the construction of mixed finite element spaces is available (see [98, 23] and the references within).

Although there are many finite element spaces for the Stokes problem that can be found in literature, all of them have the same goal, which is satisfying our approximation properties, our discrete inf-sup condition, and wanting the corresponding linear systems of equations to be efficiently solved [109]. There is an extensive mathematical foundation available where the mathematical derivation of stable element pairs is sufficiently solved (for example, see Brezzi & Fortin [23]). One major result is that, in general, the discrete velocity space has to have a higher polynomial degree than the corresponding discrete pressure space [109].

Discrete Inf-Sup Condition (LBB Condition)

We can fulfil the conditions mentioned above by using Taylor Hood elements, where $X_h = Q_{k+1}$ and $Y_h = Q_k$. Using these finite element spaces, the LBB discrete inf-sup-condition is written as

$$\inf_{0 \neq q_h \in Y_h} \sup_{0 \neq \mathbf{v}_h \in X_h} \frac{b(\mathbf{v}_h, q_h)}{||\mathbf{v}_h||_{X_h} ||q_h||_{Y_h}} \geq c > 0 \tag{2.16}$$

where c_1 is a constant, the X_h-norm is the standard H^1 norm, and the Y_h-norm is the standard L^2 norm [10].

Discrete Ellipticity (Coercivity Hypothesis)

If we again use Taylor Hood elements as we did previously for the Laplace problem, where $X_h = Q_{k+1}$ and $Y_h = Q_k$, the coercivity hypothesis is written as

$$\forall \mathbf{v}_h \in X_h, a(\mathbf{v}_h, \mathbf{v}_h) \geq c_2 ||\mathbf{v}_h||_X^2 \tag{2.17}$$

where c_2 is a constant [49].

Theorem

If $\mathbf{u}_h \in X_h \subset X = \boldsymbol{H}_0^1 = \{\mathbf{u}_h \in H^1(\Omega), \mathbf{u}_h|_{\partial\Omega} = 0\}$ and $p_h \in Y_h \in Y = L^2$, then we have well-posedness of the following system:

$$\begin{aligned} (\eta \nabla \mathbf{u}_h, \nabla \mathbf{v}_h) - (p, \nabla \cdot \mathbf{v}_h) &= (\mathbf{f}, \mathbf{v}_h) \qquad && \forall \mathbf{v}_h \in X_h \\ (\nabla \cdot \mathbf{u}_h, q) &= 0 \qquad && \forall q_h \in Y_h \end{aligned}$$

The proof of which is omitted as it is exactly the same as the continuous case except it requires the above Discrete Inf-Sup Condition and Discrete Ellipticity conditions.

2.8.3 Convergence

Following Layton [74] and using finite element space Q_k as defined in Section 2.4, we will now continue through the analysis picking Taylor-Hood elements (recall this

means using $Q_k \times Q_{k-1}$ elements) to satisfy the discrete inf-sup condition.

2.8.3.1 Velocity Bound

The continuous and discretized weak forms of (2.18) become

$$(\eta\nabla\mathbf{u}, \nabla\mathbf{v}) - (p, \nabla\cdot\mathbf{v}) = (\mathbf{f}, \mathbf{v}) \ \forall\mathbf{v} \in X, \tag{2.18}$$

$$(\eta\nabla\mathbf{u}_h, \nabla\mathbf{v}_h) - (p_h, \nabla\cdot\mathbf{v}_h) = 0 \ \forall\mathbf{v}_h \in X_h. \tag{2.19}$$

Note that $(p_h, \nabla\cdot\mathbf{v}_h) = 0$. Restrict (2.18) to $\mathbf{v}_h \in X_h$ and subtract (2.19) from (2.18) and let $\mathbf{e} = \mathbf{u} - \mathbf{u}_h$ to get

$$(\eta\nabla\mathbf{e}, \nabla\mathbf{v}_h) - (p, \nabla\cdot\mathbf{v}_h) = 0 \tag{2.20}$$

Decompose $\mathbf{e} = (\mathbf{u} - \mathbf{w}_h) + (\mathbf{w}_h - \mathbf{u}_h) = \boldsymbol{\nu} + \boldsymbol{\phi}_h$, where $\mathbf{w}_h \in X_h$. Note that our trick is that $\mathbf{v}_h \in V_h$ and therefore $(\nabla\cdot\mathbf{v}_h, q_h) = 0 \ \forall q_h \in Y_h$.

Choosing $\mathbf{v}_h = \boldsymbol{\phi}_h$, note that

$$(\eta\nabla\boldsymbol{\phi}_h, \nabla\boldsymbol{\phi}_h) + (\eta\nabla\boldsymbol{\nu}, \nabla\boldsymbol{\phi}_h) = (p, \nabla\cdot\boldsymbol{\phi}_h),$$

and that from this we get

$$(\eta\nabla\boldsymbol{\phi}_h, \nabla\boldsymbol{\phi}_h) = (p, \nabla\cdot\boldsymbol{\phi}_h) - (\eta\nabla\boldsymbol{\nu}, \nabla\boldsymbol{\phi}_h).$$

Therefore, we have

$$\begin{aligned}
\eta_{\min}||\nabla\boldsymbol{\phi}_h||^2 &= \eta_{\min}(\nabla\boldsymbol{\phi}_h, \nabla\boldsymbol{\phi}_h) \\
&\leq (\eta\nabla\boldsymbol{\phi}_h, \nabla\boldsymbol{\phi}_h) \\
&\leq (p_f, \nabla\cdot\boldsymbol{\phi}_h) - (\eta\nabla\boldsymbol{\nu}, \nabla\boldsymbol{\phi}_h) \\
&\leq |(p_f - q_h, \nabla\cdot\boldsymbol{\phi}_h)| + |(\eta\nabla\boldsymbol{\nu}, \nabla\phi_h)| \\
&\leq c||p_f - q_h||||\nabla\boldsymbol{\phi}_h|| + \eta_{\max}||\nabla\boldsymbol{\nu}||||\nabla\boldsymbol{\phi}_h||.
\end{aligned}$$

This implies that

$$\eta_{\min}||\nabla\boldsymbol{\phi}_h|| \leq c||p - q_h|| + \eta_{\max}||\nabla\boldsymbol{\nu}||.$$

Therefore,

$$\begin{aligned}
||\nabla(\mathbf{u} - \mathbf{u}_h)|| &= ||\nabla(\mathbf{u} - \mathbf{w}_h)|| + ||\nabla(\mathbf{w}_h - \mathbf{u}_h)|| \\
&\leq ||\nabla(\mathbf{u} - \mathbf{w}_h)|| + \frac{1}{\eta_{\min}}\left(c||p - q_h|| + \eta_{\max}||\nabla(\mathbf{u} - \mathbf{w}_h)||\right) \\
&\leq \left(1 + \frac{\eta_{\max}}{\eta_{\min}}\right)\inf_{\mathbf{w}_h\in X_h}||\nabla(\mathbf{u} - \mathbf{w}_h)|| + \frac{c}{\eta_{\min}}\inf_{q_h\in Y_h}||p - q_h|| \\
&= \left(1 + \frac{\eta_{\max}}{\eta_{\min}}\right)\inf_{\mathbf{w}_h\in X_h}||\nabla\boldsymbol{\nu}|| + \frac{c}{\eta_{\min}}\inf_{q_h\in Y_h}||p - q_h|| \\
&\leq \left(1 + \frac{\eta_{\max}}{\eta_{\min}}\right)h^k|\mathbf{u}|_{k+1} + \frac{c}{\eta_{\min}}h^{l+1}|p|_{l+1},
\end{aligned}$$

where in the last inequality we assume that $(X_h, Y_h) = (Q_k, Q_l)$.

2.8.3.2 Pressure Bound

We again use the continuous and discretized weak forms of (2.18) to get

$$-(\eta\nabla\mathbf{u},\nabla\mathbf{v}_h)-(p,\nabla\cdot\mathbf{v}_h) = (\mathbf{f},\mathbf{v}_h)\ \forall\mathbf{v}_h\in V_h, \tag{2.21}$$

$$-(\eta\nabla\mathbf{u}_h,\nabla\mathbf{v}_h)-(p_h,\nabla\cdot\mathbf{v}_h) = (\mathbf{f},\mathbf{v}_h)\ \forall\mathbf{v}_h\in V_h. \tag{2.22}$$

Subtracting the two yields:

$$-(\eta\nabla(\mathbf{u}-\mathbf{u}_h),\nabla\mathbf{v}_h)-(p-p_h),\nabla\cdot\mathbf{v}_h)=0. \tag{2.23}$$

Decompose $p-p_h=(p-r_h)+(r_h-p_h)$ where $r_h\in Y_h$. Then,

$$(p_h-r_h),\nabla\cdot\mathbf{v}_h)=(\eta\nabla(\mathbf{u}-\mathbf{u}_h),\nabla\mathbf{v}_h)+(p-r_h,\nabla\cdot\mathbf{v}_h). \tag{2.24}$$

Divide by $||\nabla\mathbf{v}_h||\neq 0$. Then,

$$\begin{aligned}\frac{(p_h-r_h),\nabla\cdot\mathbf{v}_h)}{||\nabla\mathbf{v}_h||} &= \frac{(\eta\nabla(\mathbf{u}-\mathbf{u}_h),\nabla\mathbf{v}_h)}{||\nabla\mathbf{v}_h||}+\frac{(p-r_h,\nabla\cdot\mathbf{v}_h)}{||\nabla\mathbf{v}_h||}\\ &\leq \frac{||\eta\nabla(\mathbf{u}-\mathbf{u}_h)||||\nabla\mathbf{v}_h||}{||\nabla\mathbf{v}_h||}+\frac{||p-r_h||||\nabla\mathbf{v}_h||}{||\nabla\mathbf{v}_h||}\\ &\leq ||\eta\nabla(\mathbf{u}-\mathbf{u}_h)||+||p-r_h||.\end{aligned}$$

We once again use the inf-sup condition (2.8) and take the supremum to get

$$\beta||p_h-r_h||\leq||\eta\nabla(\mathbf{u}-\mathbf{u}_h)||+||p-r_h||.$$

By the triangle inequality,

$$
\begin{aligned}
||p - p_h|| &\leq ||p - r_h|| + ||r_h - p_h|| \\
&\leq ||p - r_h|| + \beta^{-1}||\eta\nabla(\mathbf{u} - \mathbf{u}_h)|| + \beta^{-1}||p - r_h|| \\
&\leq (1 + \beta^{-1})||p - r_h|| + \beta^{-1}||\eta\nabla(\mathbf{u} - \mathbf{u}_h)|| \\
&\leq (1 + \beta^{-1}) \inf_{r_h \in Y_h} ||p - r_h|| + \beta^{-1}\eta_{\max} \sup_{\mathbf{u}_h \in X_h} ||\nabla(\mathbf{u} - \mathbf{u}_h)|| \\
&\leq (1 + \beta^{-1})h^{l+1}|p|_{l+1} + \beta^{-1}\eta_{\max}\left(\left(1 + \frac{\eta_{\max}}{\eta_{\min}}\right)h^k|\mathbf{u}|_{k+1} + \frac{c}{\eta_{\min}}h^{l+1}|p|_{l+1}\right) \\
&\leq h^{l+1}|p|_{l+1}\left[(1 + \beta^{-1}) + \beta^{-1}\eta_{\max}\frac{c}{\eta_{\min}}\right] + h^k|\mathbf{u}|_{k+1}\left[\beta^{-1}\eta_{\max}\left(1 + \frac{\eta_{\max}}{\eta_{\min}}\right)\right].
\end{aligned}
$$

Thus, we have shown convergence for Taylor Hood finite elements.

2.9 Grad-Div Stabilization

If inf-sup stables elements are chosen, the LBB condition mentioned in the previous section creates a bond between the velocity and pressure unknowns, thus the polynomial degree of the approximation to the pressure is less than the polynomial degree of the approximation to the velocity which means that the pressure may not get resolved when using lower order polynomials and thus an additional term in the model is required for suppressing the related instability [83].

The grad-div stabilization (see [84]) puts numerical dissipation into the method and thus, by just adding it to a method, it is possible that problems that were once turbulent become stable [56]. One can think of grad-div stabilization "as a stabilization

procedure of least-square type" [83]. It results from adding

$$-\tau\nabla(\nabla \cdot \mathbf{u}) = 0$$

to the continuous Stokes equations (1.11), (1.12) and (1.13) yielding the term

$$\tau(\nabla \cdot \mathbf{u}_h, \nabla \cdot \mathbf{v}_h)$$

in the variational formulation, where $\tau \geq 0$. The advantage of using this term are that you are able to penalize the numerical scheme when discrete mass conservation is not met [75, 114]. The disadvantage is that you are changing the energy balance of the numerical scheme [114]. This stabilization will be used in Chapter 4.

2.10 Linear Solvers for Stokes

There are many different approaches to solving Stokes-type systems [57] including, for example, Uzawa type methods, which rely on the Uzawa algorithm as explored in Bramble et al. [22] as well as Temam [107]. Arrow-Hurwicz type recursive methods are described in Temam [107]. Guermond et al. [52] used three types of fractional step (or projection) methods. Volker [110] used finite element discretizations of higher order and special multigrid methods. Elman [39] introduced a preconditioner for the linearized Navier-Stokes equations that is particularly useful when the mesh size or viscosity vanishes. Niet and Wubs [35] compared "two preconditioners for the saddle point problem: one based on the augmented Lagrangian approach and another involving artificial compressibility". Elman et al. [41] examined a preconditioning operator that was proposed by Kay and Loghin [67], and explored its behavior in the

Navier-Stokes equations. Numerical aspects of nonlinear and linear iteration schemes have been studied in detail (see, for example Turek [109]).

2.11 Krylov Methods

We now introduce the Krylov methods [73], which are quick iterative methods that solve linear systems. Krylov methods solve $A\mathbf{x} = b$ for $\mathbf{x} \in \mathbb{R}^n$ with matrix $A \in \mathbb{R}^{n \times n}$ and vector $\mathbf{b} \in \mathbb{R}^n$ [56, 73]. Krylov methods use an iterative process to calculate $\mathbf{x}_m$ to approximate $\mathbf{x}$ that begins with an initial solution $\mathbf{x}_0$ [56, 73]. This approximate solution is created in the affine subspace $\mathbf{x}_0 + K_m$ of the solution space $\mathbb{R}^n$, where

$$K_m(A, \mathbf{v}) = \text{span}\left\{\mathbf{v}, A\mathbf{v}, A^2\mathbf{v}, ..., A^{m-1}\mathbf{v}\right\} \subseteq \mathbb{R}^n$$

is the Krylov space of order m for matrix A and vector $\mathbf{v}$ [56]. One of the biggest advantages of Krylov methods is that they do not require the elements of the A matrix and instead only need to perform matrix-vector products. Krylov methods use scalar products and matrix-vector multiplications that are easily parallelized [56]. Note that if you are using a Krylov solver for parallel computing, that you also require a preconditioner that is able to handle parallel computing [56].

Typical Krylov-space methods are the Conjugate Gradient (CG) method (explained in Johnson [63] and originally from Hestenes & Stiefel [59]), GMRES (explained in Saad & Schultz [96]), and BiCGSTAB (explained in Van der Vorst [36]) [109]. Most of these Krylov-space methods were first introduced in Krylov [73].

2.11.1 GMRES

The Generalized Minimal RESidual algorithm (GMRES) is an efficient solver that creates a unique iterate in the affine subspace $\mathbf{x}_0 + K_m$ where the residual's Euclidean norm is minimized [102]. The main convergence properties of GMRES can be found in Silvester et al. [102] and the proofs of them can be found in the resources therein. For more intricate information on GMRES, the GMRES method is extensively compared to many methods in Valli & Quarteroni [89].

The GMRES method has the disadvantage that as the number of iterations k is increased, the number of necessary stored vectors is scaling with respect to k and the number of multiplications needed is scaling with respect to $\frac{1}{2}k^2N$, where N is the most steps taken before the process terminates [96]. To remedy this, we can restart the algorithm every $m \in \mathbb{Z}^+$ steps. The practical implementation of this is described in Saad & Schultz [96].

Letting M be a preconditioner, we will describe a GMRES algorithm for a sample problem $AM^{-1}(M\mathbf{x}) = \mathbf{b}$, but first, it is important to point out that we don't need elements of AM^{-1}, and instead can solve $M\mathbf{x} = v$ whenever that operation is required, as mentioned above. Therefore, it is important when using GMRES that it is simple to calculate $M^{-1}\mathbf{v}$ for any vector $\mathbf{v}$ [94]. GMRES with right preconditioning for $AM^{-1}(M\mathbf{x}) = \mathbf{b}$ is defined as follows [94, 87] :

1. Choose $\mathbf{x}_0$ and a dimension m of the Krylov space. Define an $(m+1) \times m$ matrix H_m and initialize all its entries $H_{i,j}$ to be zero.

2. Arnoldi process:

   (a) Compute $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0, \beta = ||\mathbf{r}_0||_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$.

(b) For $j = 1, ..., m$ do

Compute $\mathbf{z}_j := M^{-1}\mathbf{v}_j$

Compute $\mathbf{w} := A\mathbf{z}_j$

For $i = 1, ..., j$ do

$h_{i,j} : (\mathbf{w}, \mathbf{v}_i)$ and $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_i$

Compute $h_{j+1,j} = ||\mathbf{w}||_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/hj + 1, j.$

(c) Define $V_m := [\mathbf{v}_1, ..., \mathbf{v}_m]$.

3. Compute $\mathbf{x}_m = \mathbf{x}_0 + M^{-1}V_m\mathbf{y}_m$ where $\mathbf{y}_m = \text{argmin}_{\mathbf{y}}||\beta\mathbf{e}_1 - H_m\mathbf{y}||_2$ and $\mathbf{e}_1 = [1, 0, ..., 0]^T$.

4. If satisfied stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and restart the Arnoldi process.

The Arnoldi process creates an orthogonal basis of the preconditioned Krylov subspace using a modified Gram-Schmidt process [94, 95].

2.11.2 FGMRES

Now, we describe the FGMRES algorithm for when one is using preconditioners that are not linear operators, such as when the precondtioner is using another iterative solver within [56]. If the preconditioner is redefined as you go such that $\mathbf{z}_j := M_j^{-1}\mathbf{v}_j$ instead of $\mathbf{z}_j := M^{-1}\mathbf{v}_j$, then we would define our approximate solution as $\mathbf{x}_m = \mathbf{x}_0 + Z_m\mathbf{y}_m$, in which $Z_m = [\mathbf{z}_1, \ldots, \mathbf{z}_m]$, instead of $\mathbf{x}_m = \mathbf{x}_0 + M^{-1}V_m\mathbf{y}_m$, where $\mathbf{y}_m$ is computed the same in both cases. This is known as the flexible variant of the right preconditioned algorithm, defined as follows [94, 87]:

1. Choose $\mathbf{x}_0$ and a dimension m of the Krylov spaces. Define an $(m + 1) \times m$

matrix H_m and initialize all its entries $H_{i,j}$ to zero.

2. Arnoldi process:

   (a) Compute $\mathbf{r}_0 = b - A\mathbf{x}_0, \beta = ||\mathbf{r}_0||_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$.

   (b) For $j = 1, ..., m$ do

   Compute $\mathbf{z}_j := M_j^{-1}\mathbf{v}_j$

   Compute $\mathbf{w} := A\mathbf{z}_j$

   For $i = 1, ..., j$ do

   $h_{i,j} : (\mathbf{w}, \mathbf{v}_i)$ and $\mathbf{w} := \mathbf{w} - h_{i,j}\mathbf{v}_i$

   Compute $h_{j+1,j} = ||\mathbf{w}||_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/hj + 1, j$.

   (c) Define $Z_m := [\mathbf{z}_1, ..., \mathbf{z}_m]$.

3. Compute $\mathbf{x}_m = \mathbf{x}_0 + M^{-1}Z_m\mathbf{y}_m$ where $\mathbf{y}_m = \text{argmin}_{\mathbf{y}}||\beta\mathbf{e}_1 - H_m\mathbf{y}||_2$ and $\mathbf{e}_1 = [1, 0, ..., 0]^T$.

4. If satisfied stop, else set $\mathbf{x}_0 \leftarrow x_m$ and restart the Arnoldi process.

When compared to the right preconditionfed version, the flexible variant requires the storage of $\mathbf{z}_j$ and that the solution update requires $\mathbf{z}_j$ which effectively doubles the memory requirement [95, 94].

2.12 Amdahl's Law

We want to be able to predict the theoretical speed-up of a program when using multiple processors. Amdahl [3] analyzed parallel scalability and following Heister

[56] we explain Amdahl's Law as

$$\text{speedup}(n) = \frac{1}{1 - E_p + \frac{E_p}{n}},$$

which describes the speedup of the serial part $E_s = 1 - E_p$ of the program with n processors where $E_p \in [0, 1]$ is a perfectly parallelized fraction [56].

Chapter 3

Geometric Multigrid for Stokes

There are two main approaches to solving Stokes: you either form the Schur complement [55] (such as in Furuichi et al. [48], Murphy et al. [80] and the references therein) or attack the block system directly (such as in Silvester and Wathen [103]). If you choose the latter and wish to use multigrid [108], you have the choice of applying multigrid on the whole system at once, as we do in Chapter 4, or by only applying multigrid on the velocity block, as discussed in this chapter which is done in ASPECT [15] and deal.II step-56 [51].

In this chapter, we compare popular choices of preconditioners for the velocity block of the Stokes equation to geometric multigrid (GMG) in terms of performance and memory usage. The former include UMFPACK, ILU, and algebraic multigrid (AMG). This has been partially investigated, but since (to our knowledge) no code is available for the general public that implements GMG for the velocity block of Stokes so there is room for more detailed comparisons. We address this issue, so others can use our code as a template for their own research. The main objective of this chapter is to show GMG is at least competitive in serial computations, because this will imply that

it will outperform the other methods (especially UMFPACK and ILU) as our systems grow larger and in parallel computations due to the properties that GMG possesses.

In this chapter, we will use FGMRES with geometric multigrid as a preconditioner for the velocity block, and we will show in the results section that this provides a better approach than the linear solvers used in the deal.II tutorial step-22 [71]. Fundamentally, this is because only with multigrid it is possible to get $O(n)$ solve time, where n represents the amount of unknowns in the linear system, as discussed in Section 3.1.1. Using the Timer class of deal.II [6], we collect some statistics to compare set-up times, solve times, and number of iterations. We also compute errors to make sure what we have implemented is correct.

This tutorial was contributed by Ryan Grove and Timo Heister to the deal.II finite element library [6]. As written in the step-56 tutorial [51], "This material is based upon work partially supported by National Science Foundation grant DMS1522191 and the Computational Infrastructure in Geodynamics initiative (CIG), through the National Science Foundation under Award No. EAR-0949446 and The University of California-Davis. The Isaac Newton Institute for Mathematical Sciences in Cambridge, England deserves special thanks for support and hospitality during the programme Melt in the Mantle where work on this tutorial was undertaken. This work was supported by EPSRC grant no EP/K032208/1."

The full commented and uncommented programs can be found in the online deal.II manual under the step-56 tutorial [51].

3.1 Preconditioner

The number of iterations needed by a Krylov method depends on the eigenvalue spectrum of the matrix involved [56]. A small number of iterations is required for eigenvalues clustered away from zero. By preconditioning the linear system for the matrices where this is not the case with a linear, regular operator P^{-1}, we hope to create an improved eigenvalue spectrum, where P^{-1} will be an approximation of A^{-1}.

3.1.1 Geometric Multigrid (GMG)

At the present time, the computer power available to us enables very accurate simulations using well over a billion degrees of freedom. For problem sizes much larger than this, we are in need of numerical algorithms having optimal time complexity. The challenge we face is finding an iterative way of solving high dimensional linear systems that is efficient. This is achieved by multigrid methods due to their ability to effectively reduce both the smooth and oscillatory error using a subtle interplay of smoothing and coarse grid correction steps [97, 60, 24].

A multigrid method is an iterative method where a collection of successively coarser finite element grids can be either used as a solver or as a preconditioner to an iterative solver [63]. The motivation behind multigrid comes from noting simple iterative methods are quite effective at reducing the high frequency error, but do not reduce the low frequency error very well. These simple iterative methods of which multigrid is a part of are referred to as smoothers because they smoothen the error's high frequency part [1]. If utilized a solver, these methods are optimal as they are able to compute a discrete solution to a system of PDEs in $O(n)$ work where n is the number of unknowns. Likewise, multigrid can be applied as a preconditioner of an

iterative solver and keep its $O(n)$ time, and thus is often paired with $O(1)$ solvers. For example, in deal.II [6], a multigrid v-cycle is used as a preconditioner for another iterative solver where the goal is to precondition the system so that the iterative solver converges in $O(1)$ number of iterations, and thus when it is combined with a multigrid preconditioner which does $O(n)$ work, an $O(n)$ overall solver is created [29].

In an algebraic multigrid (AMG) scheme, no information about the grid on which the governing PDEs are discretized is used at all [112]. However, the GMG scheme creates a hierarchy of meshes that cover the computational domain and coarser grids are created based upon the geometric location of dofs. Furthermore, the coarser grids are predetermined which allows the implementer to pick intergrid transfer operators specifically for his or her problem [112]. A disadvantage of GMG is it must be implemented differently for each new problem. As stated in Heys et al [60], "the goal in GMG is to use a relaxation strategy to reduce the oscillatory errors on a given grid and rely on predetermined interpolation to effectively represent the remaining smooth error components on coarser levels".

Let a conforming coarse mesh $\mathcal{T}_0$ be given. The mesh hierarchy is defined recursively in the sense that cells of $\mathcal{T}_\ell$ are obtained by taking each cell of $\mathcal{T}_{\ell-1}$ and splitting it into congruent children and in this sense we refer to the meshes as being nested, where the index ℓ refers to the mesh level. We define the mesh size h_ℓ as the maximum of diameters of the cells of $\mathcal{T}_\ell$. Due to the refinement process, we have $h_\ell = 2^{-\ell} h_0$ on quadrilateral cells [65]. These meshes are conforming by construction [65].

3.1.1.1 Function Spaces for each Level

Following the work of Clevenger [29] and Janssen and Kanschat [62], we let $\mathbf{u}, \mathbf{v} \in V \subseteq H_0^1(\Omega)$ and $a(\mathbf{u}, \mathbf{v})$ be a bilinear form, and consider finding $\mathbf{u}$ for

$$a(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in V \tag{3.1}$$

where $\mathbf{f} \in L^2(\Omega)$. Then, consider the discrete space $V_\ell \subset V$ to be the set of continuous, piecewise functions on $\mathcal{T}_\ell$ that vanish on $\partial\Omega$. Note that for any level $\ell \geq 1$, since the meshes in $\{\mathcal{T}_\ell\}_{\ell \geq 0}$ are nested, we have that $V_{\ell-1} \subset V_\ell$. For each ℓ, for $\mathbf{u}_\ell \in V_\ell$, we want to find $\mathbf{u}_\ell$ such that

$$a_\ell(\mathbf{u}_\ell, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \quad \forall \mathbf{v} \in V_\ell \tag{3.2}$$

For $\mathbf{v}$, $\mathbf{w} \in V_\ell$, the inner product space for level l is $(\mathbf{v}, \mathbf{w})_\ell = h_\ell^2 \sum_{i=1}^{n_\ell} \mathbf{v}(p_i)\mathbf{w}(p_i)$, where the p_i are the support points of $\mathcal{T}_\ell$. Let $A_\ell : V_\ell \to V_\ell$ s.t. for $\mathbf{v}$, $\mathbf{w} \in V_\ell$, then $(A_\ell \mathbf{v}, \mathbf{w})_\ell = a_\ell(\mathbf{v}, \mathbf{w})$, where A_ℓ is symmetric positive definite operator that represents the bilinear form $a_\ell(\cdot, \cdot)$. Denote the energy norm as $\| \cdot \|_E = \sqrt{a_\ell(\cdot, \cdot)}$.

We can now write the discretized equation (3.2) as

$$A_\ell \mathbf{u}_\ell = \mathbf{f}_\ell \tag{3.3}$$

where $\mathbf{u}_\ell \in V_\ell$ and $\mathbf{f}_\ell \in V_\ell$ s.t. $(\mathbf{f}_\ell, \mathbf{v})_\ell = (\mathbf{f}, \mathbf{v})$ $\forall \mathbf{v} \in V_\ell$. Using a finite basis for V_ℓ, we get a linear system where $\mathbf{u}_\ell$ is a coefficient vector of size n_ℓ.

3.1.1.2 The V-cycle Algorithm

The combination of using both a fine grid and a coarse grid in the solution process requires the definition of transfer operators between the levels [61]. Again following the work of Clevenger [29] and Janssen and Kanschat [62], we let $I_{\ell-1}^{\ell} : V_{\ell-1} \to V_{\ell}$ be the coarse-to-fine grid operators where $I_{\ell-1}^{\ell}\mathbf{v} = \mathbf{v} \quad \forall \mathbf{v} \in V_{\ell-1}$ and the fine-to-coarse grid operators $I_{\ell}^{\ell-1} : V_{\ell} \to V_{\ell-1}$ which is defined as the transpose of $I_{\ell-1}^{\ell}$ with respect to $(\cdot, \cdot)_{\ell}$. Using a finite basis for V_{ℓ}, these operators can be expressed as rectangular matrices and $I_{\ell-1}^{\ell}$ is the transpose matrix scaled by a constant.

We can now define our algorithm for ℓth level of a multigrid v-cycle. Let $B_{\ell}^{-1}\mathbf{d}_{\ell}$ be the approximate solution of $A_{\ell}\mathbf{x}_{\ell} = \mathbf{d}_{\ell}$. Let S_{ℓ} be a set of smoothing operators and let $B_0 = A_0$ and set $\mathbf{x}_{\ell}^{(0)} = 0$. Define B_{ℓ}^{-1} in the following way:

(i) After just a few iterations of an iterative method, the error tends to smoothen by quite a bit, and this observation motivated an idea in multigrid to apply a few iterations of a simple iterative method on the fine grid in hope to dampen high frequency errors [61]. Thus, for each level, given an approximate solution, we apply a series of smoothing steps of a simple iterative method of our choosing on the residual, since just projecting onto a coarser mesh won't preserve the residual well. We call this presmoothing, where first you compute $\mathbf{x}_{\ell}^{(m_{\ell})}$, for $i = 1, ..., m_{\ell}$ you have that $\mathbf{x}_{\ell}^{(i)} = \mathbf{x}_{\ell}^{(i-1)} + S_{\ell}\left(\mathbf{d}_{\ell} - A_{\ell}\mathbf{x}_{\ell}^{(i-1)}\right)$.

(ii) Then, we map the current error out to a coarser grid because an approximation to the remaining smooth error is able to be more efficiently computed on a coarser grid than a finer one. We apply the v-cycle operator $B_{\ell-1}^{-1}$ to the residual on lower level, and this is where the recursion in this method comes in. We continue this smoothing and mapping to coarser grids recursively, and when we

arrive at the coarsest mesh we want to arrive at, we directly solve the system at a much less cost than on the finest level. Afterwards, this smoothened residual must be interpolated up to the finer grid and then be added to the current fine grid approximation in order to correct it [61]. This whole step is written as $\mathbf{x}_\ell^{(m_\ell+1)} = \mathbf{x}_\ell^{(m_\ell)} + I_{\ell-1}^{\ell} B_{\ell-1}^{-1} I_{\ell}^{\ell-1} \left(\mathbf{d}_\ell - A_\ell \mathbf{x}_\ell^{(m_\ell)}\right)$.

(iii) We then apply a second series of smoothing steps to reduce the residual. This is called postsmoothing, where first you compute $\mathbf{x}_\ell^{(2m_\ell+1)}$, for $i = m_\ell+2, ..., 2m_\ell+1$ you have that $\mathbf{x}_\ell^{(i)} = \mathbf{x}_\ell^{(i-1)} + S_\ell \left(d_\ell - A_\ell \mathbf{x}_\ell^{(i-1)}\right)$.

Set $B_\ell^{-1}\mathbf{d}_\ell = \mathbf{x}_\ell^{(2m_\ell+1)}$. This is known as the multigrid v-cycle, a graphical representation of such is visible in Figure 3.1 and the hierarchy of meshes is pictured in Figure 3.2, where B is called the multigrid preconditioner [29].

To show the convergence of the v-cycle algorithm, it suffices to show the algorithm is contraction with contraction number less than 1 and independent of the level (see [29] and [62] for proofs and explanations). A significant factor of the efficiency of the multigrid method comes within the smoothing step, where the high frequency components of the error, which correspond to large eigenvalues, are significantly reduced [63]. Then, the solution is projected down onto a coarser grid and the low frequency content of the solution is significantly deflated by the coarse grid correction on the less expensive coarse grid in the sense that low frequency error is reduced to high frequency error by the approximate removal of the low frequency components from the error [1]. In summary, we have significantly reduced the error in each multigrid step [63].

An overview of the multigrid algorithm and everything needed for multigrid can be

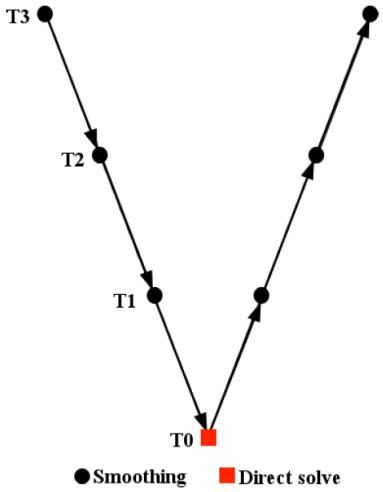

Figure 3.1: Typical V-Cycle from Clevenger [29]

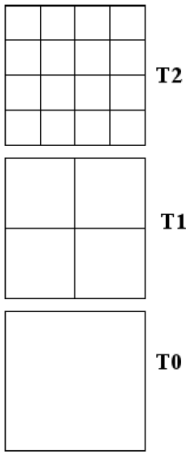

Figure 3.2: Hierarchy of Meshes from Clevenger [29]

found in Schöberl [97]. Finally, once again following the work of Clevenger [29] and Janssen and Kanschat [62], it should be shown the total work of the multigrid v-cycle with global refinement is $\mathcal{O}(n_L)$, where L is the maximum level. Let W_ℓ be the amount of work in the ℓth level v-cycle. Let $n_\ell = \dim V_\ell$ and note that for any level $\ell \geq 1$, since the meshes in $\{\mathcal{T}_\ell\}_{\ell \geq 0}$ are nested, we have that $n_{\ell-1} = C_\ell n_\ell$ where $C_\ell < 1$. Let $C_{\max} = \max_{0 \leq \ell \leq L} C_\ell$. As worked out in Clevenger [29], if we pick smoothers and level transfers that are each $\mathcal{O}(n_\ell)$ we can write

$$\begin{aligned} W_\ell &\leq C(2m)n_\ell + W_{\ell-1} \\ &\leq C(n_\ell + n_{\ell-1} + n_{\ell-2} + \cdots + n_1) \\ &= C\left(1 + \sum_{i=0}^{\ell-1} \prod_{j=i+1}^{\ell} C_j\right) n_\ell \\ &\leq C\left(\sum_{i=0}^{\ell} (C_{\max})^i\right) n_\ell \\ &\leq C\frac{C_{\max}}{1 - C_{\max}} n_\ell \quad \text{since } C_{\max} < 1 \\ &\leq C n_\ell \end{aligned} \tag{3.4}$$

Thus, $W_\ell \in \mathcal{O}(n_L)$ [29, 62].

3.1.2 Multigrid Methods for Saddle Point Problems

Unfortunately, saddle point problems are typically difficult to solve due to indefiniteness and poor spectral properties and thus the multigrid methods discussed above need to be used cleverly to be efficient in solving saddle point problems [28].

3.1.2.1 Block Preconditioning

As stated at the beginning of this chapter, our goal here is to compare several solution approaches. While step-22 [71] solves the linear system using a Schur complement approach in two separate steps, we instead attack the block system at once using FMGRES with an efficient preconditioner, in the spirit of the approach outlined in the Results section of step-22 [71].

As written in the step-56 tutorial [51], "The weak form of the discrete Stokes equations naturally leads to the following linear system for the nodal values of the velocity and pressure fields:

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} U \\ P \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}.$$

The idea is as follows: if we find a block preconditioner P such that the matrix

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} P^{-1}$$

is simple, then an iterative solver with that preconditioner will converge in a few iterations." Notice that we are doing right preconditioning here. Using the Schur complement $S = BA^{-1}B^T$, we find that

$$P^{-1} = \begin{pmatrix} A & B^T \\ 0 & -S \end{pmatrix}^{-1}$$

is a good choice. Let $\widetilde{A^{-1}}$ be an approximation of A^{-1} and $\widetilde{S^{-1}}$ of S^{-1}, we see

$$P^{-1} = \begin{pmatrix} A^{-1} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & S^{-1} \end{pmatrix} \approx \begin{pmatrix} \widetilde{A^{-1}} & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} I & B^T \\ 0 & -I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & \widetilde{S^{-1}} \end{pmatrix}.$$

Since P is aimed to be a preconditioner only, we shall use the approximations on the right in the equation above [51]. As discussed in the deal.II tutorial step-22 [71], $-M_p^{-1} =: \widetilde{S^{-1}} \approx S^{-1}$, where M_p is the pressure mass matrix and is solved approximately by using CG with ILU as a preconditioner. For our work on deal.II tutorial step-56 [51], $\widetilde{A^{-1}}$ is can be obtained by one of multiple methods: solving a linear system with CG and ILU as preconditioner, just using one application of an ILU, solving a linear system with CG and GMG as a preconditioner (as described in deal.II tutorial program step-16 [64]), or just performing a single V-cycle of GMG. That means that we only apply GMG to the velocity block which is just a vector valued Laplace operator analogous to our study of the Poisson problem in Chapter 2. On the contrary, in Chapter 4 we apply the preconditioner to the entire system matrix instead of just the velocity block!

As a comparison, instead of FGMRES, we also use the direct solver UMFPACK on the whole system to compare our results with. If you want to use a direct solver (like UMFPACK), the system needs to be invertible. To avoid the one dimensional null space given by the constant pressures, we fix the first pressure unknown to zero. This is not necessary for the iterative solvers.

A vast amount of research is being conducted on block preconditioning. Chan & Jin [27] investigated the solution of block system by the preconditioned conjugate gradient method. Elman et al. [40] explored the topic of automatically generating "a block preconditioner for solving the incompressible Navier–Stokes equations".

Notay [82] analyzed bock preconditioners for symmetric saddle point matrices. Bai [8] constructed "block-counter-diagonal and block-counter-tridiagonal preconditioning matrices to precondition Krylov subspace methods". Cao [26] looked at "applying preconditioned Krylov subspace methods to the solution of large saddle point-type systems with singular top-left blocks". Block preconditioning is also used for magnetostatic problems as explored in Perugia & Simoncini [86]. Klawonn [69] used "block-triangular preconditioners for a class of saddle point problems with a penalty term".

3.1.3 Slightly Modified Stokes Problem

Let $\mathbf{u} \in H_0^1 = \{\mathbf{u} \in H^1(\Omega), \mathbf{u}|_{\partial\Omega} = 0\}$ and $p \in L_0^2 = \{p \in L^2(\Omega), \int_\Omega p = 0\}$. The Stokes equations that we consider are read as follows in non-dimensionalized form as found in step-56 [51]:

$$\begin{aligned} -2\text{div}\frac{1}{2}\left[(\nabla\mathbf{u}) + (\nabla\mathbf{u})^T\right] + \nabla p &= f \\ -\nabla\cdot\mathbf{u} &= 0 \end{aligned}$$

Note that we are using the deformation tensor instead of $\Delta\mathbf{u}$ (a detailed description of the difference between the two can be found in step-22 [71], but in summary, the deformation tensor is more physical as well as more expensive).

3.1.4 Reference Solution

The test problem is a manufactured solution (see deal.II tutorial step-7 [13] for details), and we choose $\mathbf{u} = (u_1, u_2, u_3) = (2\sin(\pi x), -\pi y\cos(\pi x), -\pi z\cos(\pi x))$ and

$p = \sin(\pi x)\cos(\pi y)\sin(\pi z)$. For the velocity, we have Dirichlet boundary conditions over the whole boundary of the domain $\Omega = [0,1] \times [0,1] \times [0,1]$. To enforce the boundary conditions we can just use our reference solution.

If you look up in the deal.II manual [6] what is needed to create a class derived from `Function<dim>`, you will find this class has numerous `virtual` functions, including `Function::value()`, `Function::vector_value()`, `Function::value_list()`, etc., all of which can be overloaded. Different parts of deal.II will require different ones of these particular functions. This can be confusing at first, but luckily the only thing you actually have to implement is `value()`. The other virtual functions in the `Function` class have default implementations inside that will call your implementation of `value` by default.

Notice our reference solution fulfills $\nabla \cdot \mathbf{u} = 0$. In addition, the pressure is chosen to have a mean value of zero. For the `Method of Manufactured Solutions` of step-7 [13], we need to find $\mathbf{f}$ such that:

$$\mathbf{f} = -2\mathrm{div}\frac{1}{2}\left[(\nabla \mathbf{u}) + (\nabla \mathbf{u})^T\right] + \nabla p.$$

Using the reference solution above, we obtain:

$$\begin{aligned}\mathbf{f} &= (2\pi^2 \sin(\pi x), -\pi^3 y \cos(\pi x), -\pi^3 z \cos(\pi x)) \\ &\quad +(\pi \cos(\pi x)\cos(\pi y)\sin(\pi z), -\pi \sin(\pi y)\sin(\pi x)\sin(\pi z), \pi \cos(\pi z)\sin(\pi x)\cos(\pi y))\end{aligned}$$

3.1.5 Computing Errors

Because we do not enforce the mean pressure to be zero for our numerical solution in the linear system, we need to post process the solution after solving. To do this we use the `VectorTools::compute_mean_value()` function to compute the mean value of the pressure to subtract it from the pressure.

3.1.6 DoFHandlers

The way we implement geometric multigrid here only executes it on the velocity variables (i.e., the A matrix described above) but not the pressure. One could implement this in different ways, including one in which one considers all coarse grid operations as acting on 2×2 block systems where we only consider the top left block. Alternatively, we can implement things by considering a linear system on the velocity part of the overall finite element discretization. The latter is the way we want to use here.

To implement this, we created a separate, second `DoFHandler` for just the velocities. We then built linear systems for the multigrid preconditioner based on only this second `DoFHandler`, and simply transferred the first block of (overall) vectors into corresponding vectors for the entire second `DoFHandler`. To make this work, we had to assure the order that the velocity dofs are ordered in the two `DoFHandler` objects is the same. This is in fact the case by first distributing degrees of freedom on both, and then using the same sequence of `DoFRenumbering` operations on both.

3.1.7 Differences from Step-22

The main difference between step-56 [51] and step-22 [71] is the utilization of block solvers instead of the Schur Complement approach used in step-22 [71]. Details of this approach can be found under the `Block Schur complement preconditioner` subsection of the `Possible Extensions` section of step-22 [71]. For the preconditioner of the velocity block, we borrow a class from `ASPECT` [72] called `BlockSchur↩ Preconditioner` that has the option to solve for the inverse of A or just apply one preconditioner sweep for it instead, which provides us with an expensive and cheap approach, respectively.

3.2 Results

We now examine convergence rates, timings, and memory usage and discuss our findings before making some conclusions about this chapter. All calculations were made using step-56 [51] unless otherwise stated.

3.2.1 Errors

We first run the code and confirm the finite element solution converges with the correct rates as predicted by the error analysis of mixed finite element problems. Given sufficiently smooth exact solutions u and p, the errors of the Taylor-Hood element $Q_k \times Q_{k-1}$ should be

$$\|\mathbf{u} - \mathbf{u}_h\|_0 + h(\|\mathbf{u} - \mathbf{u}_h\|_1 + \|p - p_h\|_0) \leq Ch^{k+1}(\|\mathbf{u}\|_{k+1} + \|p\|_k)$$

Table 3.1: Errors for 3D Computations

	L2 Velocity	**Reduction**	**L2 Pressure**	**Reduction**	**H1 Velocity**	**Reduction**
3 global refinements	0.000670888	-	0.0036533	-	0.0414704	-
4 global refinements	8.38E-005	8.0	0.00088494	4.1	0.0103781	4.0
5 global refinements	1.05E-005	8.0	0.000220253	4.0	0.00259519	4.0

Table 3.2: Timing Results for 3D Computations

	General			**GMG**					**ILU**					**UMFPACK**	
		Timings		*Timings*		*Iterations*			*Timings*		*Iterations*			*Timings*	
Cycle	DoFs	Setup	Assembly	Setup	Solve	Outer	I_A	I_S	Setup	Solve	Outer	I_A	I_S	Setup	Solve
0	15468	0.1s	0.3s	0.3s	1.3s	21	67	22	0.3s	0.6s	21	180	22	2.65s	2.8s
1	112724	1.0s	2.4s	2.6s	14s	21	67	22	2.8s	15.8s	21	320	22	236s	237s
2	859812	9.0s	20s	20s	101s	20	65	21	27s	268s	21	592	22	-	-

see for example Ern & Guermond [43], Section 4.2.5 p195. This is indeed what we observe in table 3.1, using the $Q_2 \times Q_1$ element as an example (this is what is done in the code, but is easily changed in `main()`).

3.2.2 Timing Results

Let us compare the direct solver approach using UMFPACK to the two methods in which we choose $\widetilde{A^{-1}} = A^{-1}$ and $\widetilde{S^{-1}} = S^{-1}$ by solving linear systems with A, S using CG. The preconditioner for CG is then either ILU or GMG. Table 3.2 summarizes solver iteration and timings (where I_A, I_S are the number of inner solves for A, S, respectively), while table 3.4 summarizes virtual memory (VM) peak usage.

Additional timing results were found using the file `Files / step-56_amg.prm` from the Github repository named `dissertation` of user `rrgrove6`. The only difference between this and the standard step-56 is the additional implementation of AMG so comparisons can be made between GMG and AMG.

Similar to the results written in the step-56 tutorial [51], we can see from tables 3.2 and 3.4 that:

Table 3.3: Additional Timing Results

	AMG				
	Timings		*Iterations*		
DoFs	Setup	Solve	Outer	I_A	I_S
15468	.307s	5.67s	21	330	22
112724	2.95s	47s	21	265	22
859812	26.73s	549s	20	353	21

Table 3.4: Virtual Memory Peak (kB)

DoFs	**GMG**	**ILU**	**UMFPACK**
15468	4805	4783	5054
112724	5441	5125	11288
859812	10641	8307	-

1. UMFPACK uses large amounts of memory, especially in 3d. Also, UMFPACK timings do not scale favorably with problem size.

2. Because we are using inner solvers for A and S, ILU, AMG, and GMG require the same number of outer iterations.

3. The number of inner iterations for A increases for ILU with refinement, leading to worse than linear scaling in solve time. In contrast, the number of inner iterations for A stays constant with GMG leading to nearly perfect scaling in solve time.

4. Although the number of inner iterations for A appear constant for both AMG and GMG with refinement, the number of inner iterations for A for AMG is about 5 times that of GMG.

5. Although the number of inner iterations for A appear constant for both AMG and GMG with refinement, the number of inner iterations for A for AMG is about 5 times that of GMG.

6. As the number of unknowns increases, it can already be seen that GMG has smaller solve times (and comparable setup times) compared to the other methods.

3.2.3 Conclusions

In this chapter, we have shown that applying GMG to the velocity block while solving Stokes is competitive in serial computations in terms of performance and memory usage to UMFPACK, ILU, and AMG. This implies that it will outperform the other methods (especially UMFPACK and ILU) as our systems grow larger and in parallel computations. Additionally, GMG can be parallelized like AMG so it is much more competitive than UMFPACK or ILU for bigger problems.

This work is in a good state to serve as a template or starting point for the research of others, as everything has been well documented and the code has been made available to everyone.

Chapter 4

Schwarz smoothers for conforming inf-sup stable discretizations of the Stokes equations

As in Benzi et al. [16], let $\Omega \in \mathbb{R}^d$ be a bounded, connected domain (with dimension $d = 2, 3$) with smooth, piecewise boundary $\partial\Omega$. Given a force $\mathbf{f} : \Omega \to \mathbb{R}^d$, we solve for a velocity $\mathbf{u} : \Omega \to \mathbb{R}^d$ and a pressure $p : \Omega \to \mathbb{R}$ where

$$
\begin{aligned}
-\eta\Delta\mathbf{u} + \nabla p &= \mathbf{f} \text{ in } \Omega && (4.1)\\
\nabla \cdot \mathbf{u} &= 0 \text{ in } \Omega && (4.2)\\
\mathbf{u} &= \mathbf{0} \text{ on } \partial\Omega, && (4.3)
\end{aligned}
$$

with viscosity $\eta > 0$. The analysis and numerical results for a multigrid method with subspace correction smoother that performs very efficiently on divergence-conforming discretizations with interior penalty is considered in Kanschat & Mao [65]. For their

multigrid method for the Stokes system, Kanschat and Mao [65] used Raviart-Thomas (RT) elements [90] where:

1. $\nabla \cdot \boldsymbol{V}_h = Q_h$
2. $\boldsymbol{V}_{h,0}^{\text{div}} \subset \boldsymbol{V}_{h,1}^{\text{div}} \subset \cdots \subset \boldsymbol{V}_{h,L}^{\text{div}}$.

where $\boldsymbol{V}_{h,l}^{\text{div}} = \{\mathbf{u}_h \in \boldsymbol{V}_l, (\nabla \cdot \mathbf{u}_h, q_h) = 0 \; \forall q_h \in Q_l\}$. We wish to extend this work to include conforming inf-sup stable elements including Taylor Hood and $Q_{k+1} \times DGP_k$ elements [5]. A DGP_k finite element is a discontinuous finite element based on Legendre polynomials of degree k that we plan to use so that we have a discontinuous pressure in order to achieve cell-wise mass conservation [14].

The DGP_k finite element implements $\frac{(p+1)(p+2)}{2}$ polynomials of degree p in $2D$. For example, in $2D$, the element DGP_1 would represent the span of the functions $\{1, x, y\}$, which is in contrast to the element DGQ_1 that is formed by the span of $\{1, x, y, xy\}$ and thus it is immediately clear that the DGP_k element can not be continuous on quadrilaterals [14]. More information about them can be found in Arndt [5].

The work in this chapter, unlike the work in Chapter 3, will apply the GMG preconditioner to the entire system matrix of the discretized Stokes equation. The goal of this work is to get better iteration counts than we did when we just applied the GMG preconditioner to the velocity block in Chapter 3. If we do see smaller iteration counts, and if someone in future work finds an efficient way to handle patch-based smoothers, then this work could be a stepping stone towards revolutionizing fluid flow solvers.

This chapter, in its entirety, was jointly done with Daniel Arndt from Heidelberg University in Heidelberg, Germany.

4.1 Smoothers

The smoothers that concern us for a given space are defined to be the additive or multiplicative iterative schemes that depend on the decomposition of the space [21].

4.1.1 New Function Spaces and Finite Elements

As described in Arnold et al. [30], the Hilbert space $H^{\text{div}}(\Omega)$ consists of square-integrable vector fields on a domain Ω with square-integrable divergence. In a variety of variational formulations of systems of PDEs, the $H^{\text{div}}(\Omega)$ function space naturally arises [30]. It is defined as

$$H^{\text{div}}(\Omega) = \left\{ \mathbf{v} \in L^2(\Omega)^d | \nabla \cdot \mathbf{v} \in L^2(\Omega) \right\},$$

as seen in Kanschat & Mao [65]. The inner product in $H^{\text{div}}(\Omega)$ is given by

$$\Lambda(\mathbf{u}, \mathbf{v}) = (\mathbf{u}, \mathbf{v}) + (\nabla \cdot \mathbf{u}, \nabla \cdot \mathbf{v}),$$

where $(\cdot, \cdot)$ is used to denote the inner product in $L^2(\Omega)$ as it is in Arnold et al. [30]. We now want to associate our inner product Λ with a linear operator $\mathbf{\Lambda}$ that maps $H^{\text{div}}(\Omega)$ isometrically onto its dual space. Thus, $\mathbf{\Lambda}$ is defined as it is in Arnold et al. [30] as

$$(\mathbf{\Lambda u}, \mathbf{v}) = \Lambda(\mathbf{u}, \mathbf{v}) \text{ for all } \mathbf{v} \in H^{\text{div}}.$$

More information about $H^{\text{div}}(\Omega)$, Λ, and $\mathbf{\Lambda}$ can be found in Arnold, Falk, & Winther [30]. We now use Raviart-Thomas elements as they conform in this function space as well as the fact that Kanschat & Mao [65] use them in their paper. Also, in

subsequent sections we will use Raviart-Thomas elements to explain the additive Schwarz smoother.

Raviart-Thomas elements are not the only elements that are commonly used while solving Stokes; Crouzeix-Raviart elements are used in multigrid for the Stokes problem in Braess & Verfürth [19]. They have become quite popular as they are able to overcome the difficulty of the construction of suitable prolongation and restriction operators for the transfer from coarse to fine grids and vice versa by constructing easily computable L^2-projections based on suitable quadrature rules [19].

4.1.2 The Additive Schwarz Smoother

In [100], Schwarz created an iterative method to solve classical BVPs for harmonic functions consisting of successively solving a similar problem in subdomains while alternating from one to the other (see also [76]).

As is done in Arnold et al. [30], given a finite element subspace V_h of $H^{\text{div}}(\Omega)$, we determine a positive-definite symmetric operator $\mathbf{\Lambda}_h : V_h \to V_h$ by

$$(\mathbf{\Lambda}_h \mathbf{u}, \mathbf{v}) = \Lambda(\mathbf{u}, \mathbf{v}) \text{ for all } v \in V_h.$$

Then for any $\mathbf{f} \in V_h$, the equation

$$\Lambda_h \mathbf{u} = \mathbf{f}$$

admits a unique solution $\mathbf{u} \in V_h$ [30]. Now we want to "define a v-cycle preconditioner $\mathbf{\Theta}_h$ for the operator $\mathbf{\Lambda}_h$ using an additive Schwarz smoother formed by summing solutions to local problems in a neighborhood of each mesh vertex" as is done in

Arnold et al. [30].

Let $\mathcal{T}_H = \{\Omega_j\}_{j=1}^J$ be a triangulation of our domain Ω that has mesh size H, and let $\mathcal{T}_h$ be the refined $\mathcal{T}_H$ with mesh size $h < H$ [30]. Let $\left\{\Omega'_j\right\}_{j=1}^J$ cover Ω where for each j we have that Ω'_j is a union of squares in $\mathcal{T}_h$ and $\Omega_j \subset \Omega'_j$ [30]. For $j = 1, 2, \ldots, J$, set $\boldsymbol{V}_j = \left\{\mathbf{v} \in \boldsymbol{V} : \mathbf{v} \equiv 0 \text{ on } \Omega \setminus \Omega'_j\right\}$ [30].

Let $\boldsymbol{V}_h$ be the Raviart-Thomas space with respect to $\mathcal{T}_h$ where the discrete Helmholtz Decomposition is satisfied. Following the work in Arnold et al. [30], the decomposition $\boldsymbol{V}_h = \sum_{j=0}^{J} \boldsymbol{V}_j$ leads us to the definiton of an additive Schwarz preconditioner $\boldsymbol{\Theta}_h : \boldsymbol{V}_h \to \boldsymbol{V}_h$. $\boldsymbol{\Theta}_h$ is defined as

$$\boldsymbol{\Theta}_h = \sum_{j=0}^{J} \boldsymbol{P}_j \boldsymbol{\Lambda}_h^{-1}, \tag{4.4}$$

where $\boldsymbol{P}_j : \boldsymbol{V}_h \to \boldsymbol{V}_j$ is the $H^{\text{div}}(\Omega)$ orthogonal projection (see [30] for more information, as well as proofs on the effectiveness of the preconditioner and bounds on $P := \boldsymbol{\Theta}_h \boldsymbol{\Lambda}_h = \sum_{j=0}^{J} \boldsymbol{P}_j$).

Additive Schwarz smoothers require damping for model problems [1]. The advantage of additive smoothers is that they are easily parallelizeable since the sum in Equation 4.4 can be done in parallel.

4.2 Background

There has been a great amount of research done that relates to this work. Motivated by the index reduction technique of minimal extension, a remodelling of the Navier Stokes flow equations is proposed and analyzed using Taylor Hood and Crouzeix-Raviart finite elements in Altmann & Heiland [2]. A GMG method using a constrained Braess–Sarazin smoother, using only partial regularity assumption,

for saddle point systems using stable finite element pairs is analyzed in Chen [28]. A "p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations" using a Jacobi smoother is presented in Fidkowsi et al. [46]. For the compressible Euler equations, solutions of "high-order accurate discontinuous Galerkin discretizations of non-linear systems of conservation laws on unstructured grids" are found using the spectral hp-multigrid method in Nastase & Mavriplis [81]. Algebraic multigrid with higher-order finite elements for elliptic partial differential equations, including Stokes, are explored in Heys et al. [60]. The "performance of the multigrid method applied to spectral element discretizations of the Poisson and Helmholtz equations using smoothers based on finite element discretizations, overlapping Schwarz methods, and point-Jacobi are considered in conjunction with conjugate gradient and GMRES acceleration techniques" in Fischer & Lottes [47]. An algorithm for parallelizing the Gauss-Seidel multigrid smoother for distributed memory computers is analyzed in Adams [1]. A coupled multigrid method for generalized Stokes flow problems using Taylor Hood elements is explored in Takacs [105]. Additive Schwarz-type iteration methods for saddle point problems as smoothers in a multigrid method including looking into Crouzeix-Raviart mixed finite element for the Stokes equations is done in Schöberl & Zulehner [99]. "A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems" is presented in Klawonn & Pavarino [70].

A significant amount of the literature on domain decomposition Schwarz methods are on SPD problems in Hilbert spaces which methods rely on the SPD properties of the underlying problem and the Hilbert space structures. Feng & Lorton [45] have introduced additive Schwarz methods "for nonsymmetric and indefinite linear systems arising from continuous and discontinuous Galerkin approximations of general

nonsymmetric and indefinite elliptic partial differential equations" and use convection-diffusion equations to show that their framework is successful. Their framework allows applications of Schwarz methods to "general nonsymmetric and indefinite elliptic partial differential equations", such as Stokes [45].

4.3 Assumptions and Definitions

Following Kanschat and Mao [65], we consider for the domain Ω an admissible partition $\mathcal{T}_h$ defining a hierarchical partitioning $(\mathcal{T}_l)_{0\leq l\leq L}$ where $\mathcal{T}_l$ consists of the cells on level l.

As written in Dallmann et al. [31], for a simplex $T \in \mathcal{T}_h$ or a quadrilateral/hexahedron T in $\mathbb{R}^d$, let $\hat{T}$ be the reference unit simplex or the unit cube $(-1,1)^d$. The bijective reference mapping $F_T \colon \hat{T} \to T$ is affine for simplices and multi-linear for quadrilaterals/hexahedra [31]. Let P_k and Q_k with $k \in \mathbb{N}_0$ be the set of polynomials of degree $\leq k$ and of polynomials of degree $\leq k$ in each variable respectively [31]. Moreover, we set

$$R_k(\hat{T}) := \begin{cases} P_k(\hat{T}) & \text{on simplices } \hat{T} \\ Q_k(\hat{T}) & \text{on quadrilaterals/hexahedra } \hat{T}, \end{cases}$$

as was done in Dallmann et al. [31]. Although our analysis holds for all of the above, we will only be interested in quadrilaterals in this thesis. Define

$$\begin{aligned} Y_{l,-k}(\mathcal{T}_l) &:= \{\boldsymbol{v}_h \in L^2(\Omega) : \ \boldsymbol{v}_h|_T \circ F_T \in R_k(\hat{T}) \ \forall T \in \mathcal{T}_l\}, \\ Y_{l,k}(\mathcal{T}_l) &:= Y_{h,-k} \cap W^{1,2}(\Omega). \end{aligned}$$

For convenience, we write $\boldsymbol{V}_l = \mathbb{R}_{k_u}(\mathcal{T}_l)$ instead of $\boldsymbol{V}_l = [Y_{h,k_u}]^d \cap \boldsymbol{V}$ and $Q_l = \mathbb{R}_{\pm k_p}(\mathcal{T}_l)$ instead of $Q_l = Y_{h,\pm k_p}(\mathcal{T}_l) \cap Q$. Furthermore, we define

$$\boldsymbol{V}_h := \mathbb{R}_{k_u}(\mathcal{T}_h) = \bigcup_{0 \le l \le L} \boldsymbol{V}_l, \qquad Q_h := \mathbb{R}_{\pm k_p}(\mathcal{T}_h) = \bigcup_{0 \le l \le L} Q_l.$$

and use the notation $X_l = \boldsymbol{V}_l \times Q_l$.

4.3.1 Inf-Sup Stability (LBB Condition) with levels

Following Kanschat and Mao [65], we let $\boldsymbol{V}_l$ and Q_l be finite element spaces satisfying a discrete inf-sup-condition

$$\inf_{q \in Q_l \setminus \{0\}} \sup_{\boldsymbol{v} \in \boldsymbol{V}_l \setminus \{\boldsymbol{0}\}} \frac{(\nabla \cdot \boldsymbol{v}, q)}{\|\nabla \boldsymbol{v}\|_{\boldsymbol{V}_l} \|q\|_{Q_l}} \ge \beta > 0 \tag{4.5}$$

with a constant β independent of l [65].

In particular, this means by the closed range theorem that the space of weakly divergence-free solutions is not trivial:

$$\boldsymbol{V}_l^{\mathrm{div}} := \{\boldsymbol{v}_h \in \boldsymbol{V}_l : (\nabla \cdot \boldsymbol{v}_h, q_h) = 0 \quad \forall q_h \in Q_l\} \neq \{\boldsymbol{0}\}$$

Assumption 1 (Interpolation Operators). There are quasi-interpolation operators $j_u \colon \boldsymbol{V} \to \boldsymbol{V}_l$ and $j_p \colon Q \to Q_l$ such that for all $T \in \mathcal{T}_l$, for all $\boldsymbol{w} \in \boldsymbol{V} \cap [W^{s,2}(\Omega)]^d$ with $1 \le s \le k_u + 1$:

$$\|\boldsymbol{w} - j_u \boldsymbol{w}\|_{0,M} + h_l \|\nabla(\boldsymbol{w} - j_u \boldsymbol{w})\|_{0,M} \le C h_l^s \|\boldsymbol{w}\|_{W^{s,2}(\omega_M)}, \tag{4.6}$$

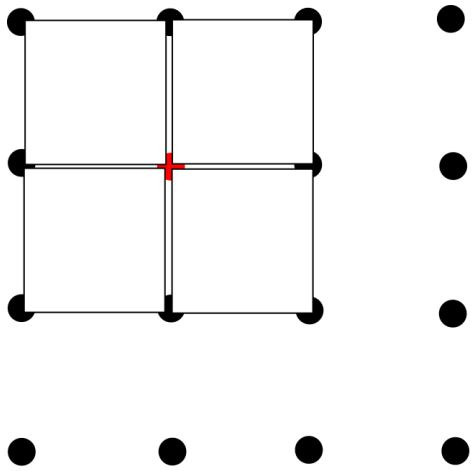

Figure 4.1: A patch $\Omega_{l,v}$ of cells $\mathcal{T}_{l,v}$ sharing an inner vertex

where M is a cell and for all $q \in Q \cap H^s(T)$ with $1 \leq s \leq k_p + 1$:

$$\|q - j_p q\|_{0,M} + h_l \|\nabla(q - j_p q)\|_{0,M} \leq C h_l^s \|q\|_{W^{s,2}(\omega_M)}, \tag{4.7}$$

on a suitable patch $\omega_M \supset T$. Moreover, let

$$\|\boldsymbol{v} - j_u \boldsymbol{v}\|_{L^\infty(M)} \leq C h_l |\boldsymbol{v}|_{W^{1,\infty}(M)} \quad \forall \boldsymbol{v} \in [W^{1,\infty}(M)]^d.$$

4.3.2 Patches

Let $\mathcal{N}_l$ be the set of inner vertices in the triangulation $\mathcal{T}_l$ and let $\mathcal{T}_{l,v}$ be the set of cells in $\mathcal{T}_l$ sharing the vertex v [65]. The set of cells around a particular inner vertex is called a patch $\Omega_{l,v}$ as seen in Figure 4.1 [65]. For Raviart-Thomas elements, all boundary patch dofs are set to zero [65]. For the conforming inf-sup stable elements, we use homogeneous Dirichlet boundary conditions for the velocity and nothing for the pressure [65].

4.3.3 Discrete Spaces on Patches

The discrete spaces we will be using are defined to be

$$\begin{aligned}
\boldsymbol{V}_{l,v} &= \{\mathbf{u}_h \in [Q_{k+1}(K)]^d \forall K \in \Omega_{l,v}, \mathbf{u}_h \in C(\Omega_{l,v}), \mathbf{u}_h|_{\partial\Omega_{l,v}} = \mathbf{0}\}, \\
Q_{l,v} &= \{p_h \in Q_k(K) \forall K \in \Omega_{l,v}, p_h \in C(\Omega_{l,v})\}, \\
\boldsymbol{X}_{l,v} &= \boldsymbol{V}_{l,v} \times Q_{l,v}.
\end{aligned}$$

Furthermore, $\boldsymbol{V}_{l,v}^{\text{div}}$ is the space of weakly divergence-free functions on that patch:

$$\boldsymbol{V}_{l,v}^{\text{div}} = \{\mathbf{u}_h \in V_{l,v} : (\nabla \cdot \mathbf{u}_h, q_h) = 0 \ \forall q_h \in Q_{l,v}\}.$$

4.4 Stokes, Perturbed Primal and Perturbed Dual Problem

Following Kanschat and Mao [65], our plan is to eliminate the pressure by considering a perturbed, discrete Stokes problem in weak formulation

$$\begin{aligned}
\alpha(\boldsymbol{u}_l, v_h) + \nu(\nabla \boldsymbol{u}_l, \nabla \boldsymbol{v}_h) + \tau_{gd}(\nabla \cdot \boldsymbol{u}_h - \epsilon p_h, \nabla \cdot \boldsymbol{v}_h - \epsilon q_h) \\
-(p_l, \nabla \cdot \boldsymbol{v}_h) + (\nabla \cdot \boldsymbol{u}_l, q_h) - \epsilon(p_l, q_h) = (\boldsymbol{f}, \boldsymbol{v}_h)
\end{aligned} \tag{4.8}$$

where $\alpha(\boldsymbol{u}_l, v_h)$ is a reaction term. Defining the operator $\mathcal{A}_l : X \times X \to X^*$ by

$$\begin{aligned}
\mathcal{A}_l((\boldsymbol{u}_l, p_l), (\boldsymbol{v}_h, q_h)) :=& \alpha(\boldsymbol{u}_l, v_h) + \nu(\nabla \boldsymbol{u}_l, \nabla \boldsymbol{v}_h) \\
&+ \tau_{gd}(\nabla \cdot \boldsymbol{u}_l - \epsilon p_l, \nabla \cdot \boldsymbol{v}_h - \epsilon q_h) \\
&+ (p_l, \nabla \cdot \boldsymbol{v}_h) + (\nabla \cdot \boldsymbol{u}_l, q_h) - \epsilon(p_l, q_h).
\end{aligned}$$

This problem can be written as $\mathcal{A}_l((\boldsymbol{u}_l, p_l), (\boldsymbol{v}_h, q_h)) = (\boldsymbol{f}, \boldsymbol{v}_h)$ for all $(\boldsymbol{v}_h, q_h) \in \boldsymbol{V}_l \times Q_l$.

Testing the perturbed Stokes problem with $(0, q_l)$, we observe

$$-\tau_{gd}(\nabla \cdot \boldsymbol{u}_l - \epsilon p_l, \epsilon q_h) + (\nabla \cdot \boldsymbol{u}_l, q_h) - \epsilon(p_l, q_h) = 0.$$

and therefore $\epsilon p_l = \pi_{Q_h} \nabla \cdot \boldsymbol{u}_l$.

Hence, for $\epsilon > 0$ the Stokes problem can be rewritten as

$$\begin{aligned} A_l(\widetilde{\boldsymbol{u}}_l, \boldsymbol{v}_h) &:= \alpha(\widetilde{\boldsymbol{u}}_l, \boldsymbol{v}_h) + \nu(\nabla \widetilde{\boldsymbol{u}}_l, \nabla \boldsymbol{v}_h) \\ &\quad + \tau_{gd}(\nabla \cdot \pi_{Q_h}^{\perp}(\nabla \cdot \widetilde{\boldsymbol{u}}_l), \nabla \cdot \pi_{Q_h}^{\perp}(\boldsymbol{v}_h)) \\ &\quad + \frac{1}{\epsilon}(\pi_{Q_h} \nabla \cdot \widetilde{\boldsymbol{u}}_l, \pi_{Q_h} \nabla \cdot \boldsymbol{v}_h). \\ A_l(\widetilde{\boldsymbol{u}}_l, \boldsymbol{v}_h) &= (\boldsymbol{f}, \boldsymbol{v}_h) \end{aligned}$$

for all $\boldsymbol{v}_h \in \boldsymbol{V}_l$.

Lemma 1. *Let $(\boldsymbol{u}_l, p_l)$ be the solution to the perturbed problem in two variables and $\widetilde{\boldsymbol{u}}_l$ the solution to the perturbed problem in one variable, Equation 4.8. Then it holds*

$$\boldsymbol{u}_l = \widetilde{\boldsymbol{u}}_l \qquad \textit{and} \qquad \epsilon p_l = \pi_{Q_h}(\nabla \cdot \boldsymbol{u}_l) = \pi_{Q_h}(\nabla \cdot \widetilde{\boldsymbol{u}}_l)$$

Proof. Following Kanschat and Mao [65], testing the two variable solution with $(0, q_l)$

gives

$$\begin{aligned}
0 &= \mathcal{A}_l\left(\begin{pmatrix}\boldsymbol{u}_l\\ p_l\end{pmatrix},\begin{pmatrix}0\\ q_l\end{pmatrix}\right)\\
&= -(\nabla\cdot\boldsymbol{u}_l, q_l) + \epsilon(p_l, q_l) - \epsilon\tau_{gd}(\nabla\cdot\boldsymbol{u}_l - \epsilon p_l, q_l)\\
&= -(1+\epsilon\tau_{gd})(\nabla\cdot\boldsymbol{u}_l - \epsilon p_l, q_l)
\end{aligned}$$

and hence $\epsilon p_l = \pi_{Q_h}(\nabla\cdot\boldsymbol{u}_l)$. Testing the two variable solution with $(\boldsymbol{v}_l, 0)$ gives

$$\begin{aligned}
(f,\boldsymbol{v}_l) &= \mathcal{A}_l\left(\begin{pmatrix}\boldsymbol{u}_l\\ p_l\end{pmatrix},\begin{pmatrix}\boldsymbol{v}_l\\ 0\end{pmatrix}\right)\\
&= \alpha(\boldsymbol{u}_l,\boldsymbol{v}_l) + \nu(\nabla\boldsymbol{u}_l,\nabla\boldsymbol{v}_l) + (\nabla\cdot\boldsymbol{v}_l, p_l) + \tau_{gd}(\nabla\cdot\boldsymbol{u}_l - \epsilon p, \nabla\cdot\boldsymbol{v}_l)\\
&= \alpha(\boldsymbol{u}_l,\boldsymbol{v}_l) + \nu(\nabla\boldsymbol{u}_l,\nabla\boldsymbol{v}_l) + \epsilon^{-1}(\pi_{Q_h}(\nabla\cdot\boldsymbol{u}_l), \nabla\cdot\boldsymbol{v}_l)\\
&\quad + \tau_{gd}(\pi_{Q_h}(\nabla\cdot\boldsymbol{u}_l - \epsilon p), \nabla\cdot\boldsymbol{v}_l)\\
&= \alpha(\boldsymbol{u}_l,\boldsymbol{v}_l) + \nu(\nabla\boldsymbol{u}_l,\nabla\boldsymbol{v}_l) + \epsilon^{-1}(\pi_{Q_h}(\nabla\cdot\boldsymbol{u}_l), \pi_{Q_h}(\nabla\cdot\boldsymbol{v}_l))\\
&\quad + \tau_{gd}(\pi^{\perp}_{Q_h}(\nabla\cdot\boldsymbol{u}_l), \nabla\cdot\boldsymbol{v}_l - \pi_{Q_h}(\nabla\cdot\boldsymbol{v}_l))\\
&= \alpha(\boldsymbol{u}_l,\boldsymbol{v}_l) + \nu(\nabla\boldsymbol{u}_l,\nabla\boldsymbol{v}_l) + \epsilon^{-1}(\pi_{Q_h}(\nabla\cdot\boldsymbol{u}_l), \pi_{Q_h}(\nabla\cdot\boldsymbol{v}_l))\\
&\quad + \tau_{gd}(\pi^{\perp}_{Q_h}(\nabla\cdot\boldsymbol{u}_l), \pi^{\perp}_{Q_h}(\nabla\cdot\boldsymbol{v}_l)).
\end{aligned}$$

and therefore $\boldsymbol{u}_l = \widetilde{\boldsymbol{u}}_l$. ■

Testing the perturbed Stokes problem with $\widetilde{\boldsymbol{u}}_l$ gives

$$(\boldsymbol{f},\widetilde{\boldsymbol{u}}_l) = \alpha\|\widetilde{\boldsymbol{u}}_l\|_0^2 + \nu\|\nabla\widetilde{\boldsymbol{u}}_l\|_0^2 + \tau_{gd}\|\pi^{\perp}_{Q_h}(\nabla\cdot\widetilde{\boldsymbol{u}}_l)\|_0^2 + \frac{\|\pi_{Q_h}(\nabla\cdot\widetilde{\boldsymbol{u}}_l)\|_0^2}{\epsilon}. \tag{4.9}$$

Hence, the perturbed bilinear form is $\boldsymbol{X}_l$-elliptic.

4.5 Estimates

The bilinear form a_l that represents the weak Laplace operator is defined as

$$a_l(\boldsymbol{u}, \boldsymbol{v}) := \nu(\nabla \boldsymbol{u}, \nabla \boldsymbol{v})$$

For $\boldsymbol{u}_l \in \boldsymbol{V}_l$ define $\boldsymbol{u}_l^0 \in \boldsymbol{V}_l^{\text{div}}$ as projection of $\boldsymbol{u}_l$ onto $\boldsymbol{V}_l^{\text{div}}$ with respect to a_l, i.e.

$$a_l(\boldsymbol{u}_l^0, \boldsymbol{v}_l) = a_l(\boldsymbol{u}_l, \boldsymbol{v}_l) \quad \forall \boldsymbol{v}_l \in \boldsymbol{V}_l^{\text{div}}.$$

Then define $\boldsymbol{u}_l^\perp$ by $\boldsymbol{u}_l^\perp := \boldsymbol{u}_l - \boldsymbol{u}_l^0$.

Lemma 2.

$$\frac{\alpha}{d}\|\nabla \cdot \boldsymbol{u}_l^\perp\|_0^2 \leq a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp) \leq \frac{\nu}{\gamma_l^2}\|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_l^\perp)\|_0^2$$

Proof. We proof the inequalities separately:

1. Due to ellipticity of the bilinear form $a_l(\cdot, \cdot)$ there exists β satisfying

   $$\beta\|\nabla \boldsymbol{u}_l^\perp\|_0^2 \leq a_l(\boldsymbol{u}_l^\perp, u^\perp).$$

   And this gives

   $$\|\nabla \cdot \boldsymbol{u}_l^\perp\|_0^2 \leq d\|\nabla \boldsymbol{u}_l^\perp\|_0^2 \leq \frac{d}{\beta} a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp).$$

   since you can bound the squared norm of the divergence by d times the squared norm of the gradient.

2. Choose $q = \pi_{Q_h}(\nabla \cdot \boldsymbol{u}_l^\perp)$, then the inf-sup stability implies that there is $\boldsymbol{v}_l \in \boldsymbol{V}_l$

satisfying

$$\gamma_l \|\nabla \boldsymbol{v}_l\| \leq \|q\|_0, \qquad q = \nabla \cdot \boldsymbol{v}_l.$$

Then we have

$$(\nabla \cdot \boldsymbol{u}_l^\perp, q_h) = (\nabla_{Q_h} \nabla \cdot \boldsymbol{u}_l^\perp, q_h) = (q, q_h) = (\nabla \cdot \boldsymbol{v}_l, q_h).$$

Let $\boldsymbol{v}_l = \boldsymbol{v}_l^0 + \boldsymbol{v}_l^\perp$ be the local Helmholtz decomposition. We then see

$$(\nabla \cdot \boldsymbol{v}, q_h) = (\nabla \cdot \boldsymbol{v}_l^\perp, q_h) = (\nabla \cdot \boldsymbol{u}_l^\perp, q_h)$$

and $\boldsymbol{v}_l^\perp = \boldsymbol{u}_l^\perp + a$ for some $a \in \boldsymbol{V}_l^{\mathrm{div}}$. We can conclude

$$\begin{aligned} a_l(v, \boldsymbol{v}) &= a_l(v^0, v^0) + a_l(\boldsymbol{v}_l^\perp, \boldsymbol{v}_l^\perp) \\ &\geq a_l(\boldsymbol{v}_l^\perp, \boldsymbol{v}_l^\perp) = a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp) + a_l(a, a) \\ &\geq a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp). \end{aligned}$$

Finally, the inequality we want to prove holds due to

$$a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp) \leq a_l(v, \boldsymbol{v}) = \nu \|\nabla \boldsymbol{v}\|_0^2 \leq \frac{\nu}{\gamma_l^2} \|q\|_0^2 \leq \frac{\nu}{\gamma_l^2} \|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_l^\perp)\|_0^2.$$

■

4.6 Convergence of the Perturbation

Let $(\boldsymbol{u}, p)$ be the solution to the continuous Stokes problem and $(\boldsymbol{u}_\epsilon, p_\epsilon)$ the solution to the continuous perturbed problem. Similarly, $(\boldsymbol{u}_h, p_h)$ and $(\boldsymbol{u}_l, p_l)$ are the solutions to the discretized problem and its perturbation.

Lemma 3. *It holds*

$$\alpha\|\boldsymbol{u} - \boldsymbol{u}_l\|_0^2 + \nu\|\nabla(\boldsymbol{u} - \boldsymbol{u}_l)\|_0^2 + \|p - p_l\|_0^2 \lesssim \epsilon^2 + h^{2k_p+2} + h^{2k_u}. \tag{4.10}$$

Proof. Use [101] for the continuous and discrete part separately to get

$$\alpha\|\boldsymbol{u_h} - \boldsymbol{u}_l\|_0^2 + \nu\|\nabla(\boldsymbol{u}_h - \boldsymbol{u}_l)\|_0^2 + \|p_h - p_l\|_0^2 \lesssim \epsilon^2.$$

Furthermore, standard theory yields

$$\alpha\|\boldsymbol{u} - \boldsymbol{u}_h\|_0^2 + \nu\|\nabla(\boldsymbol{u} - \boldsymbol{u}_h)\|_0^2 + \|p - p_h\|_0^2 \lesssim h^{2k_p+2} + h^{2k_u}$$

and finally

$$\begin{aligned}
&\alpha\|\boldsymbol{u} - \boldsymbol{u}_l\|_0^2 + \nu\|\nabla(\boldsymbol{u} - \boldsymbol{u}_l)\|_0^2 + \|p - p_l\|_0^2 \\
&\lesssim \alpha\|\boldsymbol{u} - \boldsymbol{u}_h\|_0^2 + \nu\|\nabla(\boldsymbol{u} - \boldsymbol{u}_h)\|_0^2 + \|p - p_h\|_0^2 \\
&\quad + \alpha\|\boldsymbol{u}_h - \boldsymbol{u}_l\|_0^2 + \nu\|\nabla(\boldsymbol{u}_h - \boldsymbol{u}_l)\|_0^2 + \|p_h - p_l\|_0^2 \\
&\lesssim \epsilon^2 + h^{2k_p+2} + h^{2k_u}.
\end{aligned}$$

■

4.7 Definition of Multigrid Algorithms and the Smoothers

Recalling our previously defined patches, the subspace $X_{\ell,v} = V_{\ell,v} \times Q_{\ell,v}$ consists of the functions in X_ℓ with support in $\Omega_{\ell,v}$. As written in Kanschat and Mao [65], "note that this implies homogeneous slip boundary conditions on $\partial\Omega_{\ell,v}$ for the velocity subspace $V_{\ell,v}$ and zero mean value on $\Omega_{\ell,v}$ for the pressure subspace $Q_{\ell,v}$."

Using our operator A, we define its patchwise counterpart $A_{l,v}$ by

$$A_{l,v} := I_{l,v}^T A_l I_{l,v}$$

where $I_{l,v} : \boldsymbol{V}_{l,v} \to \boldsymbol{V}_l$ denotes the embedding of $\boldsymbol{V}_{l,v}$ into $\boldsymbol{V}_l$ and $I_{l,v}^T : \boldsymbol{V}_l \to \boldsymbol{V}_{l,v}$ the L^2 projection into $\boldsymbol{V}_{l,v}$. Finally, the smoothing operator $R_l : V_l \mapsto V_{l,v}$ on level l is given by

$$R_l := \eta \sum_v I_{l,v} A_{l,v}^{-1} I_{l,v}^T$$

where $\eta \in (0, 1]$ is a scaling factor, and R_l is L^2 symmetric and positive definite [65]. We define the operator $P_{l,v} : V_l \to V_{l,v}$ by

$$P_{l,v} := A_{l,v}^{-1} I_{l,v}^T A_l.$$

Due to $P_{l,v}v \in V_{l,v}$ and

$$\begin{aligned} A_l(I_{l,v}P_{l,v}v, I_{l,v}P_{l,v}v) &= A_l(I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l v, I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l v) \\ &= v^T A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l v \\ &= v^T A_l I_{l,v}A_{l,v}^{-1}A_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l v \\ &= v^T A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l v \\ &= A_l(I_{l,v}P_{l,v}v, v), \end{aligned}$$

$P_{l,v}$ is an orthogonal projection to $V_{l,v}$ w.r.t to A_l.

Now, we are prepared to define the multigrid preconditioner $B_l : \boldsymbol{V}_l \times Q_l \to \boldsymbol{V}_l \times Q_l, (f_l, g_l) \mapsto B_l(f_l, g_l)$ we want to use: We define $B_{0,\epsilon} = A_{0,\epsilon}^{-1}$ and for $l \geq 1$ we consider the algorithm analogous to Section 3.1.1.2 and the formulas of Kanschat and Mao [65] as follows

1. Pre-smoothing: Begin with $(\boldsymbol{u}_0, p_0) = (0, 0)$ and let

$$\begin{pmatrix} u_i \\ p_i \end{pmatrix} = \begin{pmatrix} u_{i-1} \\ p_{i-1} \end{pmatrix} + R_l \left(\begin{pmatrix} f_l \\ g_l \end{pmatrix} - A_l \begin{pmatrix} u_{i-1} \\ p_{i-1} \end{pmatrix} \right), \quad i = 1, \ldots, m(l) \tag{4.11}$$

   where $m(l)$ is the number of smoothing steps.

2. Coarse grid correction:

$$\begin{pmatrix} u_{m(l)+1} \\ p_{m(l)+1} \end{pmatrix} = \begin{pmatrix} u_{m(l)} \\ p_{m(l)} \end{pmatrix} + I_{l-1}^T B_{l-1} I_l \left(\begin{pmatrix} f_l \\ g_l \end{pmatrix} - A_l \begin{pmatrix} u_{m(l)} \\ p_{m(l)} \end{pmatrix} \right) \tag{4.12}$$

3. Post-smoothing:

$$\begin{pmatrix} u_i \\ p_i \end{pmatrix} = \begin{pmatrix} u_{i-1} \\ p_{i-1} \end{pmatrix} + R_l \left(\begin{pmatrix} f_l \\ g_l \end{pmatrix} - A_l \begin{pmatrix} u_{i-1} \\ p_{i-1} \end{pmatrix} \right), \quad i = m(l)+2, \ldots, 2m(l)+1 \tag{4.13}$$

4. Assign:

$$B_l \begin{pmatrix} f_l \\ g_l \end{pmatrix} = \begin{pmatrix} u_{2m(l)+1} \\ p_{2m(l)+1} \end{pmatrix} \tag{4.14}$$

The V-cycle algorithm is then given by

$$\begin{pmatrix} u_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} u_k \\ p_k \end{pmatrix} + B_L \left(\begin{pmatrix} f \\ 0 \end{pmatrix} - A_{L,\epsilon} \begin{pmatrix} u_k \\ p_k \end{pmatrix} \right). \tag{4.15}$$

For the standard V-cycle, $m(l)$ per level l is constant

$$m_{standard}(l) = m(L) = k,$$

while for the variable V-cycle, $m(l)$ is halved per level

$$m_{variable}(l) = k \cdot 2^{l-L},$$

as done in Kanschat & Mao [65].

4.8 Equivalence of Smoothers for the Perturbed Primary and the Dual Problem

Again following the work of Kanschat and Mao [65], we seek to show the equivalence of smoothers for our perturbed primary and dual problems.

Theorem 2. $X_l := \{(u,p) \in \boldsymbol{V}_l \times Q_l : \pi_{Q_h}(\nabla \cdot u) = \epsilon p\}$

1. *$\mathcal{R}_l$ preserves X_l. For*

$$\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix} := \mathcal{R}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}$$

it holds

$$\hat{u}_l = R_l \boldsymbol{u}_l.$$

2. *$\mathcal{I}_l : X_{l-1,\epsilon} \to X_{l-1,\epsilon}$. For*

$$\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix} := \mathcal{I}_l^T \mathcal{I}_{l-1} \begin{pmatrix} u_{l-1} \\ p_{l-1} \end{pmatrix}$$

it holds

$$\hat{u}_l = \mathcal{I}_l^T \mathcal{I}_{l-1} u_{l-1}.$$

3. The coarse grid solution operator maps X_l into $X_{l-1,\epsilon}$:

$$X_{l-1,\epsilon} \ni \begin{pmatrix} \hat{u}_{l-1} \\ \hat{p}_{l-1} \end{pmatrix} := \mathcal{A}_{l-1,\epsilon}^{-1} \mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}$$

and it holds

$$\hat{u}_{l-1} = A_{l-1,\epsilon}^{-1} \mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \boldsymbol{u}_l.$$

Proof. 1. $(u,p) \in X_l$ is equivalent to

$$\mathcal{A}_l \left(\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} 0 \\ q \end{pmatrix} \right) = 0 \quad \forall q \in Q_l \tag{4.16}$$

Therefore, we check this property for the smoothed pair for all q_w that have support on just one cell w:

$$\begin{aligned}
&\mathcal{A}_l \left(\mathcal{R}_l \begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right) \\
&= \eta \sum_v \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{\substack{v \cap N(w) \neq \emptyset \\ v \not\ni w}} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&\quad + \eta \sum_{v \ni w} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix}.
\end{aligned}$$

For the second term, we get

$$
\begin{aligned}
&\eta \sum_{v \ni w} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \left(\mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \begin{pmatrix} 0 \\ q_w \end{pmatrix}^T \mathcal{I}_{l,w}^T \mathcal{I}_{l,v} \underbrace{\mathcal{I}_{l,v}^T \mathcal{A}_l \mathcal{I}_{l,v}}_{\mathcal{A}_{l,v}} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \begin{pmatrix} 0 \\ q_w \end{pmatrix}^T \mathcal{I}_{l,w}^T \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} = \eta \sum_{v \ni w} \begin{pmatrix} 0 \\ q_w \end{pmatrix}^T \mathcal{I}_{l,w}^T \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \begin{pmatrix} 0 \\ q_w \end{pmatrix}^T \mathcal{I}_{l,w}^T \begin{pmatrix} u \\ p \end{pmatrix} = \eta \sum_{v \ni w} (q_w, p).
\end{aligned}
$$

For the first term, we have in case of discontinuous ansatz spaces that $\mathcal{I}_{l,w}^T \mathcal{A}_l \mathcal{I}_{l,w}^T$ just contains face terms and only the velocity-velocity block contain these. In particular, we have

$$
\begin{aligned}
&\eta \sum_{\substack{v \cap N(w) \neq \emptyset \\ v \not\ni w}} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{\substack{v \cap N(w) \neq \emptyset \\ v \not\ni w}} \begin{pmatrix} 0 & \mathcal{I}_{l,w,p} q_w \end{pmatrix} \begin{pmatrix} \star & 0 \\ 0 & 0 \end{pmatrix} \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} = 0.
\end{aligned}
$$

Similarly, we have for discontinuous ansatz spaces ($\sum_v \mathcal{I}_{l,v}\mathcal{I}_{l,v}^T = \mathcal{I}$))

$$
\begin{aligned}
0 = \mathcal{A}_l \left(\begin{pmatrix} u \\ p \end{pmatrix}, \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right) &= \eta \sum_v \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{\substack{v \cap N(w) \neq \emptyset \\ v \not\ni w}} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&\quad + \eta \sum_{v \ni w} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix} \\
&= \eta \sum_{v \ni w} \left(\mathcal{I}_{l,w} \begin{pmatrix} 0 \\ q_w \end{pmatrix} \right)^T \mathcal{I}_{l,v} \mathcal{A}_{l,v} \mathcal{I}_{l,v}^T \begin{pmatrix} u \\ p \end{pmatrix}.
\end{aligned}
$$

Provided the first property holds, we want to prove $\hat{u}_l = R_l \boldsymbol{u}_l$:

$$
\begin{aligned}
(\mathcal{A}_l \hat{u}_l, w) &= \left(\mathcal{A}_l \begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix}, \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \left(\mathcal{A}_l \sum_v \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \sum_{\substack{v \cap N(w) \neq \emptyset \\ v \not\ni w}} \left(\mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&\quad + \eta \sum_{v \ni w} \left(\mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right)
\end{aligned}
$$

The second term gives

$$\begin{aligned}
&\eta \sum_{v \ni w} \left(\mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \sum_{v \ni w} \left(\mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,v} \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \sum_{v \ni w} \left(\mathcal{I}_{l,v}^T \mathcal{A}_l \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \sum_{v \ni w} \left(\mathcal{I}_{l,v}^T \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} \begin{pmatrix} w \\ 0 \end{pmatrix} \right) \\
&= \eta \sum_{v \ni w} (\mathcal{I}_{l,v}^T \boldsymbol{u}_l, \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} w) \\
&= \eta \sum_{v \ni w} (\mathcal{I}_{l,v}^T \mathcal{A}_l \mathcal{I}_{l,v} A_{l,v}^{-1} \mathcal{I}_{l,v}^T \boldsymbol{u}_l, \mathcal{I}_{l,v}^T \mathcal{I}_{l,w} w) \\
&= \eta \sum_{v \ni w} (\mathcal{A}_l \mathcal{I}_{l,v} A_{l,v}^{-1} \mathcal{I}_{l,v}^T \boldsymbol{u}_l, \mathcal{I}_{l,w} w)
\end{aligned}$$

2. Define the prolongated variables by $\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix} := \mathcal{I}_l^T \mathcal{I}_{l-1} \begin{pmatrix} u_{l-1} \\ p_{l-1} \end{pmatrix}$ and since this is just an inclusion mapping we have $\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix} = \begin{pmatrix} u_{l-1} \\ p_{l-1} \end{pmatrix}$ and in particular

$$\begin{aligned}
\mathcal{A}_l \left(\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix}, \begin{pmatrix} 0 \\ q_l \end{pmatrix} \right) &= \mathcal{A}_l \left(\begin{pmatrix} u_{l-1} \\ p_{l-1} \end{pmatrix}, \begin{pmatrix} 0 \\ q_l \end{pmatrix} \right) \\
&= (\nabla \cdot u_{l-1}, q_l) - (\epsilon p_{l-1}, q_l).
\end{aligned}$$

This means $\mathcal{A}_l \left(\begin{pmatrix} \hat{u}_l \\ \hat{p}_l \end{pmatrix}, \begin{pmatrix} 0 \\ q_l \end{pmatrix} \right) = 0$ for all $q_l \in Q_{l-1}$. What about $q_l \in Q_l \setminus Q_{l-1}$?

Next, we want to show $\hat{u}_l = \mathcal{I}_l(u_{l-1})$:

3. For the first property, we calculate

$$\begin{aligned}
&\mathcal{A}_{l-1,\epsilon} \left(\mathcal{A}_{l-1,\epsilon}^{-1} \mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \begin{pmatrix} 0 \\ q_{l-1} \end{pmatrix} \right) \\
&= \left(\mathcal{I}_{l-1}^T \mathcal{I}_{l-1} \mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \begin{pmatrix} 0 \\ q_{l-1} \end{pmatrix} \right) \\
&= \left(\mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \begin{pmatrix} 0 \\ q_{l-1} \end{pmatrix} \right) \\
&= \left(\mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \underbrace{\mathcal{I}_l^T \mathcal{I}_{l-1} \begin{pmatrix} 0 \\ q_{l-1} \end{pmatrix}}_{\in 0 \times Q_l} \right) = 0.
\end{aligned}$$

Finally, we show $\hat{u}_{l-1} = A_{l-1,\epsilon}^{-1} \mathcal{I}_{l-1}^T \mathcal{I}_l \mathcal{A}_l \boldsymbol{u}_l$.

We first notice

$$
\begin{aligned}
(A_{l-1,\epsilon}\hat{u}_{l-1}, \boldsymbol{v}_{l-1}) &= \left(\mathcal{A}_{l-1,\epsilon} \begin{pmatrix} \hat{u}_{l-1} \\ \hat{p}_{l-1} \end{pmatrix}, \begin{pmatrix} v_{l-1} \\ 0 \end{pmatrix} \right) \\
&= \left(\mathcal{A}_{l-1,\epsilon}\mathcal{A}_{l-1,\epsilon}^{-1}\mathcal{I}_{l-1}^T\mathcal{I}_l\mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \begin{pmatrix} v_{l-1} \\ 0 \end{pmatrix} \right) \\
&= \left(\mathcal{I}_{l-1}^T\mathcal{I}_l\mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \begin{pmatrix} v_{l-1} \\ 0 \end{pmatrix} \right) \\
&= \left(\mathcal{A}_l \begin{pmatrix} \boldsymbol{u}_l \\ p_l \end{pmatrix}, \underbrace{\mathcal{I}_l^T\mathcal{I}_{l-1} \begin{pmatrix} v_{l-1} \\ 0 \end{pmatrix}}_{\in(\boldsymbol{V}_l \times 0)} \right) \\
&= (\mathcal{A}_l\boldsymbol{u}_l, \mathcal{I}_l^T\mathcal{I}_{l-1}v_{l-1}) \\
&= (\mathcal{I}_{l-1}^T\mathcal{I}_l\mathcal{A}_l\boldsymbol{u}_l, v_{l-1}).
\end{aligned}
$$

This means $A_{l-1,\epsilon}\hat{u}_{l-1} = \mathcal{I}_{l-1}^T\mathcal{I}_l\mathcal{A}_l\boldsymbol{u}_l$ and therefore

$$\hat{u}_{l-1} = A_{l-1,\epsilon}^{-1}\mathcal{I}_{l-1}^T\mathcal{I}_l\mathcal{A}_l\boldsymbol{u}_l.$$

■

4.9 Smoother Properties

In this section we rely heavily on the work of Widlund and Toselli [113] and Feng and Karakashian [44].

Proposition 1. *If R_l satisfies the conditions*

$$A_l((I_l - R_l A_l)\boldsymbol{w}, \boldsymbol{w}) \geq 0, \quad \forall \boldsymbol{w} \in \boldsymbol{V}_l$$

and

$$(R_l^{-1}[I_l - P_{l-1}]\boldsymbol{w}, [I - P_{l-1})\boldsymbol{w}) \leq \beta_l A_l([I_l - P_{l-1}]\boldsymbol{w}, [I_l - P_{l-1}]\boldsymbol{w}), \quad \forall \boldsymbol{w} \in \boldsymbol{V}_l$$

where $\beta_l = \mathcal{O}(\gamma_l^{-1})$, then it holds

$$0 \leq A_l([I_l - \mathcal{B}_l A_l)\boldsymbol{w}, \boldsymbol{w}) \leq \delta A_l(\boldsymbol{w}, \boldsymbol{w}), \quad \forall \boldsymbol{w} \in \boldsymbol{V}_l$$

where $\delta = \frac{\hat{C}}{1+\hat{C}}$ and $\hat{C}$ is defined below.

The proof of which is omitted as it is standard multigrid theory [18, 20].

Lemma 4. *Let $\eta \leq 2^{-dim}$, then*

$$A_l((I_l - R_l A_l)\boldsymbol{w}, \boldsymbol{w}) \geq 0, \quad \forall \boldsymbol{w} \in \boldsymbol{V}_l.$$

Proof. Following Kanschat & Mao [65], consider $\boldsymbol{w} \in \boldsymbol{V}_l$ with support on a single cell K. We can decompose the term in question into

$$\begin{aligned}
&A_l((I_l - R_l A_l)\boldsymbol{w}, \boldsymbol{w}) \\
&= A_l(\boldsymbol{w}, \boldsymbol{w}) - A_l((R_l A_l)\boldsymbol{w}, \boldsymbol{w}) \\
&= A_l(\boldsymbol{w}, \boldsymbol{w}) - \eta \sum_v A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, \boldsymbol{w}) \\
&= A_l(\boldsymbol{w}, \boldsymbol{w}) - \eta \sum_v A_l(I_{l,v} P_{l,v}^T \boldsymbol{w}, \boldsymbol{w})
\end{aligned}$$

Using the projection property of $P_{l,v}$, we notice for the first part

$$\begin{aligned}(A_l I_{l,v} P_{l,v} \boldsymbol{w}, \boldsymbol{w}) &= A_l(I_{l,v} P_{l,v} \boldsymbol{w}, I_{l,v} P_{l,v} \boldsymbol{w}) \\ &= A_l(I_{l,v} P_{l,v} \boldsymbol{w}, I_{l,v} P_{l,v} \boldsymbol{w})|_{\boldsymbol{V}_{l,v}} \\ &= (A_l I_{l,v} P_{l,v} \boldsymbol{w}, \boldsymbol{w})|_{\boldsymbol{V}_{l,v}}\end{aligned}$$

due to the fact that the support of both arguments is restricted to $\boldsymbol{V}_{l,v}$.

With this, we observe

$$\begin{aligned}&A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w})|_{\boldsymbol{V}_{l,v}} \\ &= A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, \boldsymbol{w}) \\ &= A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, \boldsymbol{w})|_{\boldsymbol{V}_{l,v}} \\ &\leq (A_l(\boldsymbol{w}, \boldsymbol{w})|_{\boldsymbol{V}_{l,v}})^{\frac{1}{2}} (A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w})|_{\boldsymbol{V}_{l,v}})^{\frac{1}{2}}\end{aligned}$$

For the last term, we can estimate

$$\begin{aligned}&(A_l - A_l)(I_{l,v} P_{l,v}^T \boldsymbol{w}, I_{l,v} P_{l,v}^T \boldsymbol{w}) + A_l(I_{l,v} P_{l,v}^T \boldsymbol{w}, I_{l,v} P_{l,v}^T \boldsymbol{w}) \\ &= (\pi_{Q_h}^{\perp}(\nabla \cdot I_{l,v} P_{l,v}^T \boldsymbol{w}), \pi_{Q_h}^{\perp}(\nabla \cdot I_{l,v} P_{l,v}^T \boldsymbol{w})) + A_l(I_{l,v} P_{l,v}^T \boldsymbol{w}, I_{l,v} P_{l,v}^T \boldsymbol{w}) \\ &\leq (\pi_{Q_h}^{\perp}(\nabla \cdot I_{l,v} P_{l,v}^T \boldsymbol{w}), \pi_{Q_h}^{\perp}(\nabla \cdot I_{l,v} P_{l,v}^T \boldsymbol{w})) + A_l(I_{l,v} P_{l,v}^T \boldsymbol{w}, I_{l,v} P_{l,v}^T \boldsymbol{w}) \\ &\leq A_l(I_{l,v} P_{l,v}^T \boldsymbol{w}, I_{l,v} P_{l,v}^T \boldsymbol{w})\end{aligned}$$

and hence

$$A_l(I_{l,v} A_{l,v}^{-1} I_{l,v}^T A_l \boldsymbol{w}, \boldsymbol{w}) \leq A_l(\boldsymbol{w}, \boldsymbol{w})|_{\boldsymbol{V}_{l,v}}$$

Therefore, it holds

$$A_l((I_l - R_l A_l)\boldsymbol{w}, \boldsymbol{w}) \geq A_l(\boldsymbol{w}, \boldsymbol{w}) - \eta \sum_v A_l(\boldsymbol{w}, \boldsymbol{w})|_{\boldsymbol{V}_{l,v}}.$$

A vertex is part of a maximum of 2^{dim} cells and hence the last term is non-negative for $\eta \leq 2^{-dim}$. ■

The point of the next lemma is that it provides us with easier way to prove Lemma 6 (namely, the first inequality in the proof of Lemma 6). We only have to bound the sum of the local (energy) norms of a decomposition (we can choose) by the global (energy) norm.

Lemma 5. *It holds*

$$\eta \sum_v (R_l^{-1}\boldsymbol{u}, \boldsymbol{u}) = \inf_{\substack{\boldsymbol{u}_v \in \boldsymbol{V}_{l,v_v} \\ \sum_v I_{l,v}\boldsymbol{u}_v = u}} \sum_v A_l(I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)$$

Proof. Following Kanschat & Mao [65], we compute

$$
\begin{aligned}
&\eta(R_l^{-1}\boldsymbol{u},\boldsymbol{u}) = \eta\sum_v (R_l^{-1}\boldsymbol{u}, I_{l,v}\boldsymbol{u}_v) = \eta\sum_v (A_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u},\boldsymbol{u}_v)\\
&= \eta\sum_v (I_{l,v}^T A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u},\boldsymbol{u}_v) = \eta\sum_v (A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, I_{l,v}\boldsymbol{u}_v)\\
&\le \eta^{\frac12}\left(\eta\sum_v (A_l I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u})\right)^{\frac12}\left(\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)\right)^{\frac12}\\
&= \eta^{\frac12}\left(\eta\sum_v (I_{l,v}A_{l,v}^{-1}A_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, R_l^{-1}\boldsymbol{u})\right)^{\frac12}\left(\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)\right)^{\frac12}\\
&= \eta^{\frac12}\left(\eta\sum_v (I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, R_l^{-1}\boldsymbol{u})\right)^{\frac12}\left(\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)\right)^{\frac12}\\
&= \eta^{\frac12}(R_l R_l^{-1}\boldsymbol{u}, R_l^{-1}\boldsymbol{u})^{\frac12}\left(\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)\right)^{\frac12}\\
&= \eta^{\frac12}(u, R_l^{-1}\boldsymbol{u})^{\frac12}\left(\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v)\right)^{\frac12}\\
&\le \sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v).
\end{aligned}
$$

For $\boldsymbol{u}_v = \eta A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}$ it holds

$$
\sum_v I_{l,v}\boldsymbol{u}_v = \sum_v \eta I_{l,v}A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u} = R_l R_l^{-1}\boldsymbol{u} = \boldsymbol{u}
$$

and

$$
\begin{aligned}
\sum_v (A_l I_{l,v}\boldsymbol{u}_v, I_{l,v}\boldsymbol{u}_v) &= \sum_v (A_l I_{l,v}\eta A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, I_{l,v}\eta A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u})\\
&= \eta\sum_v (I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l I_{l,v}\eta A_{l,v}^{-1}I_{l,v}^T R_l^{-1}\boldsymbol{u}, R_l^{-1}\boldsymbol{u})\\
&= \eta(I_{l,v}A_{l,v}^{-1}I_{l,v}^T A_l u, R_l^{-1}\boldsymbol{u}) = \eta(u, R_l^{-1}\boldsymbol{u}).
\end{aligned}
$$

■

Lemma 6.

$$(R_l^{-1}[I_l - P_{l-1}]\boldsymbol{w}, [I - P_{l-1})\boldsymbol{w}) \leq \beta_l A_l([I_l - P_{l-1}]\boldsymbol{w}, [I_l - P_{l-1}]\boldsymbol{w}), \quad \forall \boldsymbol{w} \in \boldsymbol{V}_l$$

Proof. Using Lemma 5 we are left with showing

$$\sum_v (A_l I_{l,v} \boldsymbol{u}_v, I_{l,v} \boldsymbol{u}_v) \leq \beta_l A_l([I_l - P_{l-1}]\boldsymbol{w}, [I_l - P_{l-1}]\boldsymbol{w})$$

for a decomposition $(\boldsymbol{u}_v)_v$ of $[I_l - P_{l-1}]\boldsymbol{w}$, i.e.

$$\boldsymbol{u} := [I_l - P_{l-1}]\boldsymbol{w} = \sum_v I_{l,v} \boldsymbol{u}_v.$$

We choose the decomposition defined in Section 4.10 and note

$$\begin{aligned} \boldsymbol{u}_0 = P_{l-1} u &= P_{l-1}[I_l - P_{l-1}]\boldsymbol{w} \\ &= [P_{l-1} I_l - P_{l-1}^2]\boldsymbol{w} = 0. \end{aligned}$$

Therefore it holds $\boldsymbol{u}_j \in \boldsymbol{V}_{l,j}, j = 1, \ldots, J$ and Theorem 3 gives

$$\sum_{j=1}^{J} a_l(\boldsymbol{u}_j, \boldsymbol{u}_j) = \sum_{j=0}^{J} a_l(\boldsymbol{u}_j, \boldsymbol{u}_j) \lesssim a_l(\boldsymbol{u}, \boldsymbol{u}).$$

Now, for the problem in one variable, we get

$$
\begin{aligned}
&\sum_v A_L(\boldsymbol{u}_v, \boldsymbol{u}_v) \\
&= \sum_v a_l(\boldsymbol{u}_v, \boldsymbol{u}_v) + \alpha(\boldsymbol{u}_v, \boldsymbol{u}_v) + \epsilon^{-1}\|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_v)\|_0^2 + \tau_{gd}\|\pi_{Q_h}^{\perp}(\nabla \cdot \boldsymbol{u}_v)\|_0^2 \\
&\leq C a_l(\boldsymbol{u}, \boldsymbol{u}) + C\alpha(\boldsymbol{u}, \boldsymbol{u}) + \sum_v \epsilon^{-1}\|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_v)\|_0^2 + \tau_{gd}\|\pi_{Q_h}^{\perp}(\nabla \cdot \boldsymbol{u}_v)\|_0^2 \\
&= C a_l(\boldsymbol{u}, \boldsymbol{u}) + \sum_v \epsilon^{-1}\|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_v^{\perp})\|_0^2 + \tau_{gd}\|\nabla \cdot \boldsymbol{u}_v^{\perp}\|_0^2 \\
&\leq C a_l(\boldsymbol{u}, \boldsymbol{u}) + \sum_v \epsilon^{-1} a_l(\boldsymbol{u}_v^{\perp}, \boldsymbol{u}_v^{\perp}) + \tau_{gd}\|\nabla \cdot \boldsymbol{u}_v^{\perp}\|_0^2 \\
&\leq C a_l(\boldsymbol{u}, \boldsymbol{u}) + C\epsilon^{-1} a_l(\boldsymbol{u}_l^{\perp}, \boldsymbol{u}_l^{\perp}) + \sum_v \tau_{gd}\|\nabla \cdot \boldsymbol{u}_v^{\perp}\|_0^2 \\
&\leq C a_l(\boldsymbol{u}, \boldsymbol{u}) + C\epsilon^{-1}\frac{\nu}{\gamma_l^2}\|\pi_{Q_h}(\nabla \cdot \boldsymbol{u}_l^{\perp})\|_0^2 + C\frac{\tau_{gd}}{\nu} a_l(\boldsymbol{u}, \boldsymbol{u}) \\
&\leq C\frac{\tau_{gd}}{\nu} A_L(\boldsymbol{u}_v, \boldsymbol{u}_v)
\end{aligned}
$$

For the step

$$\sum_v a_l(\boldsymbol{u}_v^{\perp}, \boldsymbol{u}_v^{\perp}) \leq C a_l(\boldsymbol{u}_l^{\perp}, \boldsymbol{u}_l^{\perp})$$

we need to have some justification. This is the main difficulty and the reason we follow Feng and Karakashian [44], and although progress has been made, this analysis is not yet complete.

■

4.10 Domain Decomposition for Continous Lagrange Elements

Following Widlund and Toselli [113] and Feng and Karakashian [44], we attempt to find a suitable domain decomposition for the proof of Lemma 6. For a Cartesian mesh on level l, i,e. $\mathcal{T}_l$, consider the (inner) patches $(\Omega_{l,j})_{j=1,\dots,J}$. Then, we can decompose the space $\boldsymbol{V}_l$ as follows: Let $(x_{j,i})_{i=1\dots k}$ be the support points of $V_{l,j}$, Assume that there exist non-negative C^1-functions $\{\theta_j\}_{j=1}^J$ such that

- $\sum_j \theta_j \equiv 1$ in $\mathcal{T}_l \setminus \partial\mathcal{T}_l$
- $\theta_j = 0$ in $\overline{\mathcal{T}_l \setminus \Omega_{l,j}}$
- $\|\nabla\theta_j\|_{L^\infty} \leq \frac{1}{h_l}$

where $N(x_{i,j})$ is the number of patches that contain the support point $x_{j,i}$

$$N(x_{i,j}) := \left|\{(\Omega_{l,j})_{j=1,\dots,J} : x_{i,j} \in \Omega_{l,j} \setminus \partial\Omega_{l,j}\}\right|.$$

Let $\Pi_{V_{l,j}}$ be the Lagrange interpolation operator onto $V_{l,j}$ and define for $\boldsymbol{v}_l \in \boldsymbol{V}_l$

$$v_{l,0} := I_{l,v}^T P_{l-1} I_{l,v} \boldsymbol{v}_l \qquad\qquad \boldsymbol{v}_{l,j} := \Pi_{V_{l,j}}(\theta_j \boldsymbol{v}_l - v_{l,0} \in V_{l,j}.$$

By construction, it holds $\sum_j \boldsymbol{v}_{l,j} = \boldsymbol{v}_l$ and $\boldsymbol{v}_{l,j}$ defines a decomposition of $\boldsymbol{v}_l$.

4.10.1 Estimates

Following Feng and Karakashian [44] and Widlund and Toselli [113], we prove the following error projection

Lemma 7.

$$a_l(P_{l-1}\boldsymbol{u}, P_{l-1}\boldsymbol{u}) \leq a_l(\boldsymbol{u}, \boldsymbol{u}), \tag{4.17}$$

$$\|\boldsymbol{u} - P_{l-1}\boldsymbol{u}\|_{0,\Omega}^2 \lesssim a_l(\boldsymbol{u}, \boldsymbol{u}) h_{l-1}^2 \left(\frac{\nu + \tau_{gd}}{\tau_{gd}}\right)^2. \tag{4.18}$$

Proof. Due to the fact that P_{l-1} is a projection operator with respect to a_l, which is the same for each level, the first claim holds true. For the second claim, consider the auxiliary problem:

Find $(\boldsymbol{\phi}_u, \phi_p) \in H_0^1(\Omega) \times L^2(\Omega)$ such that

$$\nu(\nabla\boldsymbol{\phi}_u, \nabla\boldsymbol{\psi}_u) - (\nabla\cdot\boldsymbol{\phi}_u, \psi_p) - (\nabla\cdot\boldsymbol{\psi}_u, \phi_p) - \epsilon(\phi_p, \psi_p) = (\boldsymbol{u} - P_{l-1}\boldsymbol{u}, \boldsymbol{\psi}_u)$$

for all $(\boldsymbol{\psi}_u, \psi_p) \in H_0^1(\Omega) \times L^2(\Omega)$.

This problem has a unique solution $(\boldsymbol{\phi}_u, \phi_p) \in H^2(\Omega) \times H^1(\Omega)$ (provided Ω is convex) and it holds

$$\nu\|\boldsymbol{\phi}_u\|_{2,\Omega}^2 + \|\phi_p\|_{1,\Omega}^2 \lesssim \|\boldsymbol{u} - P_{l-1}\boldsymbol{u}\|_{0,\Omega}^2.$$

Now, we can conclude

$$\begin{aligned}
\|\boldsymbol{u} - P_{l-1}\boldsymbol{u}\|_{0,\Omega}^2 &= a_l(\phi, \boldsymbol{u} - P_{l-1}\boldsymbol{u}) \\
&\leq \inf_{v \in X_{l-1}} a_l(\phi - v, \boldsymbol{u} - P_{l-1}\boldsymbol{u}) \\
&\leq a_l(\boldsymbol{u} - P_{l-1}\boldsymbol{u}, \boldsymbol{u} - P_{l-1}\boldsymbol{u})^{\frac{1}{2}} \inf_{v \in X_{l-1}} a_l(\phi - v, \phi - v)^{\frac{1}{2}} \\
&\lesssim a_l(\boldsymbol{u}, \boldsymbol{u})^{\frac{1}{2}} h_{l-1}((\nu + \tau_{gd})\|\boldsymbol{\phi}_u\|_{2,\Omega} + \|\phi_p\|_{1,\Omega}) \\
&\lesssim a_l(\boldsymbol{u}, \boldsymbol{u})^{\frac{1}{2}} h_{l-1} \frac{\nu + \tau_{gd}}{\nu} \|\boldsymbol{u} - P_{l-1}\boldsymbol{u}\|_{0,\Omega} \\
&\lesssim a_l(\boldsymbol{u}, \boldsymbol{u}) h_{l-1}^2 \left(\frac{\nu + \tau_{gd}}{\nu}\right)^2.
\end{aligned}$$

■

Theorem 3. *For any $\boldsymbol{v}_l \in \boldsymbol{V}_l$ consider the decomposition $\boldsymbol{v}_{l,j}$ defined above. For this decomposition it holds*

$$\sum_{j=0}^{J} a_l(\boldsymbol{v}_{l,j}, \boldsymbol{v}_{l,j}) \lesssim a_l(\boldsymbol{v}_l, v_l)$$

provided $\tau_{gd} \lesssim \min\{\nu, \epsilon^{-1}\}$.

Proof. $w_u := u - \boldsymbol{u}_0$ $w_p := p - p_0$

$$\begin{aligned}
a_l(\boldsymbol{v}_{l,j}, \boldsymbol{v}_{l,j}) &= \sum_{K_i \in \Omega_{l,j}} \nu \|\nabla \boldsymbol{v}_{l,j}\|_{0,K_i}^2 + \tau_{gd}\|\nabla \cdot \boldsymbol{v}_{l,j} - \epsilon p_{l,j}\|_{0,K_i}^2 + \epsilon \|p_{l,j}\|_{0,K_i}^2 \\
&\leq \sum_{K_i \in \Omega_{l,j}} \nu \|\nabla \boldsymbol{v}_{l,j}\|_{0,K_i}^2 + 2\tau_{gd}\|\nabla \cdot \boldsymbol{v}_{l,j}\|_{K_i}^2 + 2(\tau_{gd}\epsilon^2 + \epsilon)\|p_{l,j}\|_{0,K_i}^2
\end{aligned}$$

Let $\overline{\theta}_{i,j}$ be the average of θ_j over K_i. It holds that

$$\|\theta_j - \overline{\theta}_{i,j}\|_{L^\infty(K_j)} \leq c h_j h_l^{-1}.$$

Using this inequality we can bound the first term by

$$\begin{aligned}\|\nabla \boldsymbol{v}_{l,j}\|^2_{0,K_i} &\leq 2\|\nabla \Pi_{K_i}(\overline{\theta_{i,j}} w_u)\|^2_{0,K_i} + 2\|\nabla \Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_u\|^2_{0,K_i} \\ &\leq 2\|\nabla w_u\|^2_{0,K_i} + c h_j^{-2} \|\Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_u\|^2_{0,K_i}\end{aligned}$$

and the last one by

$$\begin{aligned}\|p_{l,j}\|^2_{0,K_i} &\leq 2\|\Pi_{K_i}(\overline{\theta_{i,j}} w_p)\|^2_{0,K_i} + 2\|\nabla \Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_p\|^2_{0,K_i} \\ &\leq 2\|w_p\|^2_{0,K_i} + c\|\Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_p\|^2_{0,K_i}.\end{aligned}$$

Now,

$$\begin{aligned}\|\Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_u\|^2_{0,K_i} &\leq c h_j^d \|\theta_i - \overline{\theta_{i,j}}\|^2_{L^\infty(K_i)} \|w_u\|^2_{L^\infty(K_i)} \\ &\leq c\|\theta_i - \overline{\theta_{i,j}}\|^2_{L^\infty(K_i)} \|w_u\|^2_{0,K_i} \\ \|\Pi_{K_i}(\theta_i - \overline{\theta_{i,j}}) w_p\|^2_{0,K_i} &\leq c h_j^d \|\theta_i - \overline{\theta_{i,j}}\|^2_{L^\infty(K_i)} \|w_p\|^2_{L^\infty(K_i)} \\ &\leq c\|\theta_i - \overline{\theta_{i,j}}\|^2_{L^\infty(K_i)} \|w_p\|^2_{0,K_i}.\end{aligned}$$

and this gives

$$\begin{aligned}\|\nabla \boldsymbol{v}_{l,j}\|^2_{0,K_i} &\leq 2\|\nabla w_u\|^2_{0,K_i} + c h_l^{-2} \|w_u\|^2_{0,K_i} \\ \|p_{l,j}\|^2_{0,K_i} &\leq c\|w_p\|^2_{0,K_i}.\end{aligned}$$

Since the divergence-free subspaces are not nested we here cannot do more than using the same estimate for the divergence stabilization and we have

$$a_l(\boldsymbol{v}_{l,j}, \boldsymbol{v}_{l,j}) \lesssim \sum_{K_i \in \Omega_{l,j}} (\nu + \tau_{gd})(\|\nabla w_u\|^2_{0,K_i} + h_l^{-2}\|w_u\|^2_{0,K_i}) + (\tau_{gd}\epsilon^2 + \epsilon)\|w_p\|^2_{0,K_i}.$$

Finally, using Lemma 7 we estimate

$$\begin{aligned}
&\sum_j a_l(\boldsymbol{v}_{l,j}, \boldsymbol{v}_{l,j}) \\
&\lesssim \sum_j \sum_{K_i \in \Omega_{l,j}} \Big[(\nu + \tau_{gd})(\|\nabla w_u\|^2_{0,K_i} + h_l^{-2} \left(\frac{\nu + \tau_{gd}}{\nu}\right)^2 h_{l-1}^2 a_l(\boldsymbol{u}, \boldsymbol{u})) \\
&\qquad\qquad + (\tau_{gd}\epsilon^2 + \epsilon)\|w_p\|^2_{0,K_i} \Big] \\
&\lesssim \left(1 + \left(\frac{\nu + \tau_{gd}}{\nu}\right)^2 \frac{h_{l-1}^2}{h_l^2} + \frac{\tau_{gd}\epsilon^2 + \epsilon}{\epsilon}\right) a_l(\boldsymbol{u}, \boldsymbol{u}) \\
&\lesssim \left(\frac{\nu + \tau_{gd}}{\nu} + \frac{(\nu + \tau_{gd})^3}{\nu^2} \frac{h_{l-1}^2}{h_l^2} + \tau_{gd}\epsilon\right) a_l(\boldsymbol{u}, \boldsymbol{u}) \\
&\lesssim a_l(\boldsymbol{u}, \boldsymbol{u}).
\end{aligned}$$

provided $\tau_{gd} \leq \min\{\nu, \epsilon^{-1}\}$. ■

4.11 Numerical Results

- We use GMRES with the $\mathcal{B}_{l,\epsilon}$ preconditioner we defined.
- Instead of an additive smoother $\mathcal{R}_{l,\epsilon} := \eta \sum_v \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T$, we show the use of a multiplicative smoother $\mathcal{R}_{l,\epsilon} := \eta \prod_v \mathcal{I}_{l,v} \mathcal{A}_{l,v}^{-1} \mathcal{I}_{l,v}^T$. The multiplicative smoother, unlike the additive smoother, cannot be used in parallel.
- We use the file `Files/polynomial.prm` from the Github repository named `dissertation` of user `rrgrove6`.

The test problem that we used is

$$\begin{aligned}
-\eta\Delta\mathbf{u} + \nabla p &= -\eta\Delta\mathbf{u}_{\text{ref}} + \nabla p_{\text{ref}} \qquad (4.19) \\
\nabla \cdot \mathbf{u} &= 0 \qquad (4.20)
\end{aligned}$$

with the reference solution $\mathbf{u}(x,y) = (\sin^2(\pi x)\sin(2\pi y), -\sin^2(\pi y)\sin(2\pi x))$ and $p(x,y) = \sin(\pi x)\cos(\pi y)$ in 2D and a similar one in 3D.

In the following tables, Q_{Bubble} refers to the "implementation of a scalar Lagrange finite element that yields the finite element space of continuous, piecewise polynomials of degree p in each coordinate direction plus some bubble enrichment space spanned by $(2x_j - 1)^{p-1}\Pi_{i=0}^{\text{dim}-1}(x_i(1-x_i))$" as described in the deal.II manual [6].

Table 4.1: Iteration counts in 2D with $\nu =$ `1e-6` and nondistorted mesh using additive smoother with smoother relaxation term of .25 for all elements

	$Q_2 \times Q_1$			$Q_2 \times DGP_1$			$Q_{\text{Bubble}} \times Q_1$			$Q_2 \times Q_1 + DG_0$		
	γ			γ			γ			γ		
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	2	2	2	2	2	2	3	4	4	2	2	2
1	15	14	28	13	12	24	31	30	62	15	15	28
2	54	49	281	18	18	70	90	88	1000f	53	50	193
3	1000f	1000f	1000f	19	19	236	1000f	1000f	1000f	1000f	1000f	1000f
4	1000f	1000f	1000f	20	18	459	1000f	1000f	1000f	1000f	1000f	1000f

Table 4.2: Iteration counts in 2D with $\nu =$ `1e-6` and nondistorted mesh using additive smoother with smoother relaxation term of .25 for $Q_2 \times DGP_1$ elements and .0625 for all other elements

	$Q_2 \times Q_1$			$Q_2 \times DGP_1$			$Q_{\text{Bubble}} \times Q_1$			$Q_2 \times Q_1 + DG_0$		
	γ			γ			γ			γ		
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	2	2	2	2	2	2	3	4	4	2	2	2
1	20	20	41	13	12	24	37	37	79	24	21	43
2	33	34	254	18	18	70	69	65	1000f	49	47	265
3	53	47	1000f	19	19	236	189	175	1000f	98	83	1000f
4	62	52	1000f	20	18	459	263	244	1000f	182	166	1000f

4.11.1 Interpretation of Numerical Results

From these numerical results, it is clear that local Schwarz smoothers are applicable for inf-sup conforming elements and that we achieve comparable results to

Table 4.3: Iteration counts in 2D with $\nu =$ `1e-6` and nondistorted mesh using multiplicative smoother with smoother relaxation term of 1.0 for all elements

	$Q_2 \times Q_1$			$Q_2 \times DGP_1$			$Q_{\text{Bubble}} \times Q_1$			$Q_2 \times Q_1 + DG_0$		
		γ			γ			γ			γ	
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	1	1	1	1	1	1	1	1	1	1	1	1
1	6	6	16	3	3	9	18	17	38	7	6	16
2	9	8	49	5	5	32	28	34	97	21	19	58
3	10	9	138	6	5	89	37	40	553	65	60	381
4	11	9	282	6	5	195	38	41	-	-	-	-

Table 4.4: Iteration counts in 2D with $\nu =$ `1e-6` and nondistorted mesh using additive smoother with smoother relaxation term of .25 for $Q_3 \times DGP_2$ elements and .0625 for all other higher order elements

	$Q_3 \times Q_2$			$Q_3 \times DGP_2$			$Q_{\text{Bubble}}(3) \times Q_2$			$Q_3 \times Q_2 + (DG_0?)$		
		γ			γ			γ			γ	
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	2	2	2	2	2	2	9	9	8	2	2	2
1	25	24	21	16	16	12	43	44	70	30	28	22
2	42	39	43	18	17	19	81	78	276	58	53	45
3	46	42	60	18	17	30	154	139	1000f	93	84	72
4	48	43	65	28	16	39	167	148	1000f	163	143	94

Table 4.5: Iteration counts in 2D with $\nu =$ `1e-6` and nondistorted mesh using multiplicative smoother with smoother relaxation term of 1.0 for all higher order elements

	$Q_3 \times Q_2$			$Q_3 \times DGP_2$			$Q_{\text{Bubble}}(3) \times Q_2$			$Q_3 \times Q_2 + (DG_0?)$		
		γ			γ			γ			γ	
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	1	1	1	1	1	1	1	1	1	1	1	1
1	5	5	5	3	3	3	16	16	27	7	8	6
2	9	9	10	4	4	6	32	35	44	17	16	12
3	12	11	18	4	3	8	39	41	76	31	28	22
4	13	11	31	3	3	8	46	44	156	57	50	37

Table 4.6: Iteration counts in 2D with $\nu = 1$ and nondistorted mesh using additive smoother with smoother relaxation term of .25 for all elements

	$Q_2 \times Q_1$			$Q_2 \times DGP_1$			$Q_{\text{Bubble}} \times Q_1$			$Q_2 \times Q_1 + DG_0$		
		γ			γ			γ			γ	
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	2	2	2	2	2	2	4	4	4	2	2	2
1	14	14	14	11	11	10	35	35	35	14	14	13
2	52	52	48	13	13	12	150	150	193	40	40	41
3	1000f	1000f	488	14	14	13	1000f	1000f	1000f	1000f	1000f	1000f
4	1000f	1000f	1000f	15	15	14	1000f	1000f	1000f	1000f	1000f	1000f

Table 4.7: Iteration counts in 3D with $\nu =$ `1e-6` and nondistorted mesh using additive smoother with smoother relaxation term of .25 for all elements

	$Q_2 \times Q_1$			$Q_2 \times DGP_1$			$Q_{\text{Bubble}} \times Q_1$			$Q_2 \times Q_1 + DG_0$		
		γ			γ			γ			γ	
GR	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0	0.0	1.e-6	1.0
0	2	2	2	2	2	2	2	5	2	2	2	2
1	35	34	477	21	20	72	183	177	1000f	38	32	194
2	1000f	1000f	1000f	30	38	426	1000f	1000f	1000f	1000f	1000f	1000f

the Raviart-Thomas elements studied in Kanschat & Mao [65]. We also see that $Q_k \times DGP_{k-1}$ elements perform better than $Q_k \times Q_{k-1}$ elements.

4.12 Conclusions

Our goal was to extend Kanschat's work to include include $Q_{k+1} \times DGP_k$ elements but we also look at numerical results for Taylor Hood ($Q_{k+1} \times Q_k$), $Q_{\text{Bubble}}(k+1) \times Q_k$, and $Q_{k+1} \times Q_k + DG_0$ elements, that is, we wanted to show that Schwarz methods can be used as multigrid smoother for the Stokes equations using conforming and inf-sup stable discretization spaces, and that the iteration counts are sufficiently small. We have strong numerical evidence to support that we can do this for the $Q_k \times DGP_{k-1}$

elements, but the analysis is not complete, as we need justification for the step

$$\sum_v a_l(\boldsymbol{u}_v^\perp, \boldsymbol{u}_v^\perp) \leq C a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp)$$

which is the main difficulty. By applying the GMG preconditioner to the entire system matrix for Stokes, we hoped to get much better numbers for our iteration counts than we did when we just applied the GMG preconditioner to the velocity block of Stokes in Chapter 3. This seems to be the case, and if someone, in the future, finds an efficient way to handle patch-based smoothers (as right now there it is just too expensive build all of the local inverses), then this work could be a stepping stone towards revolutionizing fluid flow solvers.

Chapter 5

Three-field Stokes

5.1 Introduction

Inspired from the work of Keller et al [68], Rhebergen et al [92, 91] and Dannberg and Heister [32], we explore the analysis of the three-field Stokes equations (5.1), (5.2), and (5.3), as described in Chapter 1. Scientists in geoscience have seemingly been using this formulation without a complete mathematical understanding, since, no complete analysis or discussions of its discretization have been published. We investigate extending the solvers developed in earlier chapters of this thesis to the three-field Stokes equation to try to improve existing solvers used in current competitive geoscience codes. There are numerous researchers in the geoscience community that the results of this chapter directly impact, as simulating flows using the three-field Stokes equation is a fundamental necessity in their research.

5.2 ASPECT

The Open Source code ASPECT (Advanced Solver for Problems in Earth's ConvecTion) [12] implements "state of the art algorithms for high-Rayleigh number flows such as those in the Earth's mantle" [72]. While working on this thesis we provided numerous contributions to the ASPECT library including the implementation of an advection only solver.

5.3 Introduction to Melt

As said by Dannberg and Heister [32], "mantle convection and melt migration are important processes for our understanding of the physics of Earth's interior and how it is linked to observations at the surface". It is important to have a simple physical model that can be solved by standard methods which can describe the generation of a partially molten rock, and the separation of the melt from this rock [79]. In other words, when a heated porous rock rises and the pressure on the rock reduces, this allows melt to generate in its pores. That is why Dannberg and Heister [32] use the original formulation of the partial differential equations that model the two-phase flow of the Earth's mantle that were derived by McKenzie [79], which addresses the compressibility of both individual phases making this formulation consistent for higher pressures as well.

5.4 Strong Form

Given forces $\mathbf{f} : \Omega \to \mathbb{R}^d$ and $\mathbf{g} : \Omega \to \mathbb{R}^d$, we seek a velocity $\mathbf{u} : \Omega \to \mathbb{R}^d$, a fluid pressure $p_f : \Omega \to \mathbb{R}$, and a compaction pressure $p_c : \Omega \to \mathbb{R}$ where $p_c = (1-\phi)(p_s - p_f)$ such that

$$
\begin{aligned}
-\nabla \cdot (\eta \nabla \mathbf{u}) + \nabla p_f + \nabla p_c &= \mathbf{f} && (5.1)\\
\nabla \cdot \mathbf{u} - \nabla \cdot (k_D \nabla p_f) &= \mathbf{g} && (5.2)\\
\nabla \cdot \mathbf{u} + \frac{1}{\epsilon} p_c &= 0 && (5.3)
\end{aligned}
$$

where $\eta > 0$ is the shear viscosity, $k_D \geq 0$ is the Darcy coefficient, and $\frac{1}{\epsilon} > 0$ (so that we recover Stokes if $k_D = 0$), where ϵ is the bulk viscosity. Note that although $\eta, \epsilon > 0$ are not constants, they are bounded, and thus we assume that $0 < \eta_{\min} \leq \eta \leq \eta_{\max}$ and $0 < \epsilon_{\min} \leq \epsilon \leq \epsilon_{\max}$, respectively.

5.5 Assumptions

For our analysis, we need $\mathbf{u}$ to be $\mathbf{0}$ on the boundary

$$\mathbf{u}|_{\partial\Omega} = \mathbf{0}, \tag{5.4}$$

p_f to be mean zero throughout the domain

$$\int p_f \, d\Omega = 0, \tag{5.5}$$

as well as a boundary condition for p_f (there are multiple ways to do this; we pick it to make the analysis easier):

$$\nabla p_f \cdot n = 0. \tag{5.6}$$

5.6 The k_D Cases

In our analysis of the three-field form, we found that vanishing k_D changes the nature of the PDE, so we will split our analysis up into three cases: $k_D = 0$, $k_D > 0$, and $k_D \geq 0$.

5.7 Case 1: $k_D = 0$ everywhere

The first case we look at is when $k_D = 0$ everywhere (that is, there is no melt), and in this case, we want to recover Stokes flow of the solid both analytically and computationally.

5.7.1 Wellposedness (Continuous)

Let $\mathbf{u} \in X = \boldsymbol{H}_0^1 = \{\mathbf{u} \in H^1(\Omega), \mathbf{u}|_{\partial\Omega} = 0\}$ and $p_f \in Y = L^2(\Omega)$. Letting $\phi = k_D = 0$, we reduce equations (5.1) and (5.2) to:

$$\begin{aligned} -\nabla \cdot (\eta \nabla \mathbf{u}) + \nabla p_f + \nabla p_c &= \mathbf{f} \\ \nabla \cdot \mathbf{u} \cancel{- \nabla \cdot (k_D \nabla p_f)} &= \mathbf{g} \\ \nabla \cdot \mathbf{u} + \frac{1}{\epsilon} p_c &= 0 \end{aligned}$$

But, since $p_c = p_s - p_f$ when $\phi = 0$, we get

$$\begin{aligned} -\nabla \cdot (\eta \nabla \mathbf{u}) + \nabla p_s &= \mathbf{f} + \epsilon \nabla \mathbf{g} \\ \nabla \cdot \mathbf{u} &= 0 \end{aligned}$$

When there is no melt, $\mathbf{g}$ is typically $\mathbf{0}$ and then $p_s = p_f$ since p_c would then have to become zero. Therefore, we have

$$-\nabla \cdot (\eta \nabla \mathbf{u}) + \nabla p_f = \mathbf{f} \tag{5.7}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{5.8}$$

The analysis for this case was done entirely in Chapter 2.

5.8 Case 2: $k_D > 0$ non-constant

We now turn our attention to the case where $k_D > 0$, or in other words, there is melt everywhere. In this case there is no need for an inf-sup condition and thus different finite element choices can be made!

5.8.1 Well-posedness

Letting $u \in X = H_0^1 = \{\mathbf{u} \in H^1(\Omega), \mathbf{u}|_{\partial\Omega} = \mathbf{0}\}$, $p_f \in Y = H_*^1 = \{p_f \in H^1(\Omega), \nabla p_f \cdot n = 0, \int_\Omega p_f = 0\}$, and $p_c \in Z = L^2$, we test equations (5.1), (5.2), and (5.3) with

$\mathbf{v}, q_f, q_c$, integrate by parts, and balance derivatives to get:

$$\begin{aligned}
(\eta\nabla\mathbf{u}, \nabla\mathbf{v}) - (p_f, \nabla\cdot\mathbf{v}) - (p_c, \nabla\cdot\mathbf{v}) &= (\mathbf{f}, \mathbf{v}) && (5.9)\\
(\nabla\cdot\mathbf{u}, q_f) + (k_D\nabla p_f, \nabla q_f) &= (\mathbf{g}, q_f) && (5.10)\\
(\nabla\cdot\mathbf{u}, q_c) + \frac{1}{\epsilon}(p_c, q_c) &= 0 && (5.11)
\end{aligned}$$

Note that since X, Y, and Z are all Hilbert spaces, then $W = X \bigotimes Y \bigotimes Z$ is also a Hilbert space. Now, we want

$$a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right) : W \to \mathbb{R}$$

where the energy norm is

$$||(\mathbf{u}, p_f, p_c)||_W = \sqrt{|\eta\mathbf{u}|_1^2 + ||p_f||^2 + |k_D p_f|_1^2 + ||\frac{1}{\epsilon}p_c||^2}. \qquad (5.12)$$

Adding the left hand sides of equations (5.9), (5.10), and (5.11) gives

$$\begin{aligned}
a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right) &= (\eta\nabla\mathbf{u}, \nabla\mathbf{v}) - (p_f, \nabla\cdot\mathbf{v})\\
&\quad -(p_c, \nabla\cdot\mathbf{v}) + (\nabla\cdot\mathbf{u}, q_f)\\
&\quad +(k_D\nabla p_f, \nabla q_f) + (\nabla\cdot\mathbf{u}, q_c)\\
&\quad +(\frac{1}{\epsilon}p_c, q_c).
\end{aligned}$$

Notice that if we flipped the sign on $-(p_f, \nabla\cdot\mathbf{v})$ and $-(p_c, \nabla\cdot\mathbf{v})$ we would have gotten a saddle point problem. We now try to satisfy the following theorem:

Theorem 2.6 (Banach-Necas-Babuska) from Brezzi et al [23]

Let W be a Banach space and let V be a reflexive Banach space. Let $a \in \mathcal{L}(W \times V; \mathbb{R})$ and $\mathbf{f} \in V'$. Then, the problem

$$\text{Seek } \mathbf{u} \in W \text{ s.t. } a(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}) \qquad \forall \mathbf{v} \in V$$

is well-posed if and only if:

$$\exists \alpha > 0, \inf_{\mathbf{w} \in W} \sup_{\mathbf{v} \in V} \frac{a(\mathbf{w}, \mathbf{v})}{||\mathbf{w}||_W ||\mathbf{v}||_V} \geq \alpha \tag{5.13}$$

as well as

$$\forall \mathbf{v} \in V, (\forall \mathbf{w} \in W, a(\mathbf{w}, \mathbf{v}) = \mathbf{0}) \implies (\mathbf{v} = \mathbf{0}). \tag{5.14}$$

Moreover, the a priori estimate holds:

$$\forall \mathbf{f} \in V', ||\mathbf{u}||_W \leq \frac{1}{\alpha} ||\mathbf{f}||_{V'}. \tag{5.15}$$

Note that for us $V = W$ and we have already mentioned that W is a Hilbert space so we have that it is a reflexive Banach space. Now we must show that $a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right)$ is a continuous bilinear form on $W \times W$.

Continuity

Looking at $|a((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c))|$ and applying Cauchy-Schwarz yields

$$\begin{aligned} |a((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c))| &\leq ||\eta\nabla\mathbf{u}||||\nabla\mathbf{v}|| && (5.16)\\ &\quad +||p_f||||\nabla\cdot\mathbf{v}|| && (5.17)\\ &\quad +||p_c||||\nabla\cdot\mathbf{v}|| && (5.18)\\ &\quad +||\nabla\cdot\mathbf{u}||||q_f|| && (5.19)\\ &\quad +||k_D\nabla p_f||||\nabla q_f|| && (5.20)\\ &\quad +||\nabla\cdot\mathbf{u}||||q_c|| && (5.21)\\ &\quad +||\frac{1}{\epsilon}p_c||||q_c|| && (5.22) \end{aligned}$$

Taking this line by line, we can reduce the above. Before we do this let's look at our norm (5.12) again:

$$\sqrt{|\eta\mathbf{u}|_1^2 + ||p_f||^2 + |k_D p_f|_1^2 + ||\frac{1}{\epsilon}p_c||^2} \geq ||\eta\nabla\mathbf{u}||. \tag{5.23}$$

Similarly, we find

$$\sqrt{|\eta\mathbf{u}|_1^2 + ||p_f||^2 + |k_D p_f|_1^2 + ||\frac{1}{\epsilon}p_c||^2} \geq ||k_D\nabla p_f||, \tag{5.24}$$

as well as

$$\sqrt{|\eta\mathbf{u}|_1^2 + ||p_f||^2 + |k_D p_f|_1^2 + ||\frac{1}{\epsilon}p_c||^2} \geq ||\frac{1}{\epsilon}p_c||. \tag{5.25}$$

Now that we have these inequalities, we look at term (5.16):

$$\begin{aligned} ||\eta\nabla\mathbf{u}||||\nabla\mathbf{v}|| &\leq ||\eta\nabla\mathbf{u}||\frac{1}{\eta_{\min}}||\eta\nabla\mathbf{v}|| \\ &\leq \frac{1}{\eta_{\min}}||(\mathbf{u},p_f,p_c)||_W||(\mathbf{v},q_f,q_c)||_W. \end{aligned}$$

We now look at term (5.17):

$$\begin{aligned} ||p_f||||\nabla\cdot\mathbf{v}|| &\leq C_{pc}||\nabla p_f||||\nabla\mathbf{v}|| \\ &\leq C_{pc}\frac{1}{k_{D_{\min}}}||k_D\nabla p_f||\frac{1}{\eta_{\min}}||\eta\nabla\mathbf{v}|| \\ &\leq C_{pc}\frac{1}{k_{D_{\min}}}\frac{1}{\eta_{\min}}||(\mathbf{u},p_f,p_c)||_W||(\mathbf{v},q_f,q_c)||_W. \end{aligned}$$

Similarly, we reduce term (5.18):

$$\begin{aligned} ||p_c||||\nabla\cdot\mathbf{v}|| &\leq ||p_c||||\nabla\mathbf{v}|| \\ &\leq \epsilon_{\max}||\frac{1}{\epsilon}p_c||\frac{1}{\eta_{\min}}||\eta\nabla\mathbf{v}|| \\ &\leq \frac{\epsilon_{\max}}{\eta_{\min}}||(\mathbf{u},p_f,p_c)||_W||(\mathbf{v},q_f,q_c)||_W. \end{aligned}$$

Similarly, we reduce term (5.19):

$$\begin{aligned} ||\nabla\cdot\mathbf{u}||||q_f|| &\leq ||\nabla\mathbf{u}||C_{pc}||\nabla q_f|| \\ &\leq \frac{1}{\eta_{\min}}||\eta\nabla\mathbf{u}||C_{pc}\frac{1}{k_{D_{\min}}}||k_D\nabla q_f|| \\ &\leq \frac{1}{\eta_{\min}}C_{pc}\frac{1}{k_{D_{\min}}}||(\mathbf{u},p_f,p_c)||_W||(\mathbf{v},q_f,q_c)||_W. \end{aligned}$$

where C_{pc} is the Poincaré constant [88]. Similarly, we reduce term (5.20).

$$\begin{aligned} ||k_D \nabla p_f|| ||\nabla q_f|| &\leq ||k_D \nabla p_f|| \frac{1}{k_{D_{\min}}} ||k_D \nabla q_f|| \\ &\leq \frac{1}{k_{D_{\min}}} ||(\mathbf{u}, p_f, p_c)||_W ||(\mathbf{v}, q_f, q_c)||_W \end{aligned}$$

Similarly, we reduce term (5.21):

$$\begin{aligned} ||\nabla \cdot \mathbf{u}|| ||q_c|| &\leq ||\nabla \mathbf{u}|| ||q_c|| \\ &\leq \frac{1}{\eta_{\min}} ||\eta \nabla \mathbf{u}|| \epsilon_{\max} || \frac{1}{\epsilon} q_c|| \\ &\leq \frac{\epsilon_{\max}}{\eta_{\min}} ||(\mathbf{u}, p_f, p_c)||_W ||(\mathbf{v}, q_f, q_c)||_W. \end{aligned}$$

Similarly, we reduce term (5.22):

$$\begin{aligned} || \frac{1}{\epsilon} p_c|| ||q_c|| &\leq || \frac{1}{\epsilon} p_c|| \epsilon_{\max} || \frac{1}{\epsilon} q_c|| \\ &\leq \epsilon_{\max} ||(\mathbf{u}, p_f, p_c)||_W ||(\mathbf{v}, q_f, q_c)||_W. \end{aligned}$$

Therefore, we have that

$$|a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right)| \leq c_1 ||(\mathbf{u}, p_f, p_c)||_W ||(\mathbf{v}, q_f, q_c)||_W. \tag{5.26}$$

where $c_1 = \epsilon_{\max} + \frac{1}{k_{D_{\min}}} + \frac{1}{\eta_{\min}} \left(1 + 2\left(\epsilon_{\max} + C_{pc} \frac{1}{k_{D_{\min}}}\right)\right)$.

Boundedness of Right Hand Side

Adding the right hand sides of equations (5.10) and (5.11) yields

$$\begin{aligned} F(\mathbf{v}, q_f, q_c) &= G(\mathbf{v}) + H(q_f) \\ &\leq ||G||_{H^{-1}}||\nabla \mathbf{v}|| + ||H||_{Y'}||\nabla q_f|| \\ &= ||G||_{H^{-1}}\frac{1}{\eta_{\min}}||\eta \nabla \mathbf{v}|| + ||H||_{Y'}\frac{1}{k_{D_{\min}}}||k_D \nabla q_f|| \\ &\leq \left(\frac{1}{\eta_{\min}}||G||_{H^{-1}} + \frac{1}{k_{D_{\min}}}||H||_{Y'}\right)||(\mathbf{v}, q_f, q_c)||_W. \end{aligned}$$

Therefore, we have

$$F(\mathbf{v}, q_f, q_c) \leq c_3||(\mathbf{v}, q_f, q_c)||_W, \tag{5.27}$$

where $c_3 = \frac{1}{\eta_{\min}}||G||_{H^{-1}} + \frac{1}{k_{D_{\min}}}||H||_{Y'}$.

Fulfilling equation (5.13)

Instead of equation (5.13), we use Remark 2.9 from Brezzi et al [23] to instead satisfy

$$\forall \mathbf{u}, p \quad \exists \mathbf{v}, q \text{ s.t. } \frac{a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right)}{||(\mathbf{v}, q_f, q_c)||} \geq \alpha||(\mathbf{u}, p_f, p_c)|| \tag{5.28}$$

Let $\mathbf{u}, p$ be given and use Lemma $A.42$ in Brezzi et al [23] to write that $\forall \mathbf{v}_f \in H_0'$ such that:

$$p_f = \nabla \cdot \mathbf{v}_f \tag{5.29}$$

$$||\nabla \mathbf{v}_f|| \leq c_0||p_f||. \tag{5.30}$$

Note that if we pick $\mathbf{v} = -\mathbf{v}_f, q_f = q_c = 0$, then we have

$$\begin{aligned} a\left((\mathbf{u}, p_f, p_c), (-\mathbf{v}_f, 0, 0)\right) &= -(\eta\nabla\mathbf{u}, \nabla\mathbf{v}_f) + ||p_f||^2 - (p_c, \nabla\cdot\mathbf{v}_f) \\ &= -(\eta\nabla\mathbf{u}, \nabla\mathbf{v}_f) + ||p_f||^2 - (p_c, p_f). \end{aligned}$$

And therefore, using Holder's Inequality twice with $\xi_1 = \frac{1}{2}$ and $\epsilon_2 < \frac{c_0^2}{2\xi_1}$ yields the following inequality

$$\begin{aligned} a\left((\mathbf{u}, p_f, p_c), (-\mathbf{v}_f, 0, 0)\right) \geq\ & ||p_f||^2 - \frac{1}{2\xi_2}||\eta\nabla\mathbf{u}||^2 - \frac{\xi_2}{2}||\eta\nabla\mathbf{v}_f|| \\ & -\frac{1}{2\xi_1}||p_c||^2 - \frac{\xi_1}{2}||p_f||^2. \end{aligned}$$

And making sure $\frac{1}{2\xi_2}, \frac{\xi_2}{2}, \frac{1}{2\xi_1}$, and $\frac{\xi_1}{2}$ are all strictly less than 1 (which was achieved with our choice of ξ_1 and ξ_2), by letting $\mathbf{v} = \mathbf{u} - \alpha\mathbf{v}_f$ (where $\alpha = \min\left\{\frac{c_0^2\eta_{\min}}{\eta_{\max}^2}, \frac{1}{\epsilon_2}\right\}$), $q_f = p_f$, and $q_f = q_c$), we obtain the following

$$\begin{aligned} a\left((\mathbf{u}, p_f, p_c), (\mathbf{v}, q_f, q_c)\right) &\geq |\eta\mathbf{u}|_1^2 + |k_D p_f|_1^2 + ||\frac{1}{\epsilon}p_c||^2 + \alpha||p_f||^2 \\ &\quad -\frac{\alpha}{2\xi_2}||\eta\nabla\mathbf{u}||^2 - \frac{\alpha\xi_2}{2}||\eta\nabla\mathbf{v}_f|| - \frac{\alpha}{2\xi_1}||p_c||^2 - \frac{\alpha\xi_1}{2}||p_f||^2 \\ &\geq \frac{1}{2}||\mathbf{u}, p||^2 \\ &\geq \frac{1}{2}||\mathbf{u}, p||\frac{1}{1+\alpha\eta_{\max}}||\mathbf{v}, q||_W \end{aligned}$$

where the last inequality comes from the fact that

$$\begin{aligned} ||\mathbf{v}, q|| &\leq ||\mathbf{u}, p|| + ||\alpha \mathbf{v}_f, 0|| \\ &\leq ||\mathbf{u}, p|| + \alpha \eta_{\max} ||\nabla \mathbf{v}_f|| \\ &= ||\mathbf{u}, p|| + \alpha \eta_{\max} ||p_f|| \\ &\leq (1 + \alpha \eta_{\max}) \cdot ||\mathbf{u}, p|| \end{aligned}$$

Fufilling equation (5.14)

For us, equation (5.14) boils down to showing

$$\forall (\mathbf{u}, p), [\forall (\mathbf{v}, q),\ a((\mathbf{u}, p), (\mathbf{v}, q)) = 0] \implies p_f = p_c = 0 \text{ and } \mathbf{u} = \mathbf{0} \tag{5.31}$$

Let $\mathbf{u}, p = (p_f, p_c)$ be given and

$$a((\mathbf{u}, p), (\mathbf{u}, p)) = ||\eta \nabla \mathbf{u}||^2 + ||k_D \nabla p_f||^2 + ||p_f||^2 + ||\frac{1}{\epsilon} p_c|| = 0 \tag{5.32}$$

Now we will use three different steps to show equation (5.31):

Step 1:

Let $q = (0, 0), \mathbf{v} = \mathbf{u}$ and thus

$$\begin{aligned} 0 &= a((\mathbf{u}, p), (\mathbf{u}, 0)) \\ &= ||\eta \nabla \mathbf{u}||^2 \\ &\geq c||\mathbf{u}||^2 \implies ||\mathbf{u}||^2 = 0 \implies \mathbf{u} = \mathbf{0}. \end{aligned}$$

Step 2:

Let $q = (0, p_c), \mathbf{v} = \mathbf{0}$ and thus

$$\begin{aligned} 0 &= a((\mathbf{u}, p), (\mathbf{0}, (0, p_c))) \\ &= ||\frac{1}{\epsilon} p_c||^2 \implies p_c = 0 \text{ if } \epsilon < \infty. \end{aligned}$$

Step 3:

Let $q = (0, 0), \mathbf{v} = \mathbf{u} - \alpha \mathbf{v}_f$ and thus

$$\begin{aligned} 0 &\geq ||p_f||^2 + ||k_D \nabla p_f|| \\ &\implies ||p_f||^2 = 0 \implies p_f = 0. \end{aligned}$$

5.8.2 Convergence Rates

Letting $W = H_0^1 \times H_*^1 \times L_2$, $W_h = Q_k \times Q_l \times Q_m$, and $p = (p_f, p_c)$ we get

$$\begin{aligned} ||\mathbf{u} - \mathbf{u}_h, p - p_h||_W^2 \leq\ & ch^{2k} \eta_{max} |\mathbf{u}|_{k+1}^2 + h^{2(l+1)} |p_f|_{l+1}^2 \\ & + ch^{2l} k_{D_{max}} |p_f|_{l+1}^2 + ch^{2(m+1)} \frac{1}{\epsilon_{min}} |p_c|_{m+1}^2 \end{aligned}$$

$\implies ||\mathbf{u} - \mathbf{u}_h, p - p_h||_W \leq ch^{\min(k, l+1, l, m+1)}$. This agrees with the results found in Dannberg and Heister [32]. When using $Q_2 \times Q_1 \times Q_1$ elements (called $Q_2 Q_1 Q_1$ henceforth in this chapter), both Dannberg and Heister [32] and this analysis shows that you get suboptimal rates.

5.9 Numerical Results

We now present our numerical results for the cases where $k_D = 0$ and $k_D = 1 > 0$.

5.9.1 Test problem

The following test problem is used in ASPECT [72] for all of the proceeding calculations:

$$\begin{aligned} \eta &= 1 \\ \xi &= 0.1 + 0.1e^{1-20(x^2+y^2)} \\ u &= (\cos(y), \sin(x)) \\ \text{div } u &= 0 \\ p_s &= \sin(xy) \\ p_c &= \sin(x+y) \\ p_f &= -2\sin(x+y) + \sin(xy) \end{aligned}$$

5.9.2 Convergence Rates

The convergence rates for the case where $k_D = 0$ is seen in Table 5.1 and the case where $k_D = 1 > 0$ is seen in Table 5.2.

5.9.3 Expected vs. Calculated / Case 3: $k_D \geq 0$

Note that Table 5.3 gives expected and calculated orders of convergence in the L_2 norm, where red implies suboptimal rates and X implies that there was no conver-

Table 5.1: L^2 convergence rates for $k_D = 0$

	$Q_2Q_2Q_1$						$Q_2Q_1Q_1$					
h	**u**	ratio	p_f	ratio	p_c	ratio	**u**	ratio	p_f	ratio	p_c	ratio
3.5E-1	1.8E-4	-	8.4E-3	-	4.1E-3	-	1.8E-4	-	3.3E+4	-	4.1E-3	-
1.7E-1	2.2E-5	8.0	2.1E-3	4.0	1.0E-3	4.0	2.2E-5	8.0	1.3E+4	2.6	1.0E-3	4.0
8.8E-2	2.8E-6	8.0	5.2E-4	4.0	2.5E-4	4.0	2.8E-6	8.0	1.1E+4	1.2	2.5E-4	4.0
4.4E-2	3.5E-7	8.0	1.3E-4	4.0	6.4E-5	4.0	3.5E-7	8.0	4.1E+2	27.4	6.4E-5	4.0

Table 5.2: L^2 convergence rates for $k_D = 1 > 0$

	$Q_2Q_2Q_1$						$Q_2Q_1Q_1$					
h	**u**	ratio	p_f	ratio	p_c	ratio	**u**	ratio	p_f	ratio	p_c	ratio
3.5E-1	1.8E-4	-	4.0E-4	-	4.1E-3	-	3.2E-3	-	1.4E-2	-	4.2E-3	-
1.7E-1	2.2E-5	8.1	5.0E-5	8.0	1.0E-3	4.0	8.1E-4	4.0	3.5E-3	4.0	1.0E-3	4.0
8.8E-2	2.8E-6	8.0	6.3E-6	8.0	2.5E-4	4.0	2.0E-4	4.0	8.8E-4	4.0	2.6E-4	4.0
4.4E-2	3.5E-7	8.0	7.8E-7	8.0	6.4E-5	4.0	5.1E-5	4.0	2.2E-4	4.0	6.5E-5	4.0

gence:

Table 5.3: Optimal, expected and calculated convergence rates (L_2-norm)

	$Q_2\ Q_1\ Q_1$	$Q_2\ Q_2\ Q_1$
optimal rates	3 2 2	3 3 2
$k_D = 0$ (expected)	3 2 2	X X X
$k_D = 0$ (calculated)	3 2 2	3 X 2
$k_D = 0$ (expected)	2 2 1	3 3 2
$k_D >= 1$ 0 (calculated)	2 2 2	3 3 2

In conclusion, $Q_2Q_1Q_1$ elements always converge but have suboptimal rates if $k_D > 0$. $Q_2Q_2Q_1$ elements have optimal rates but do not converge if $k_D = 0$. Therefore, in realistic problems where $k_D > 0$ somewhere but $k_D = 0$ most places, it is important to use $Q_2Q_1Q_1$ elements so that you get convergence. If you bound k_D away from zero, then it is best to use $Q_2Q_2Q_1$ as you see optimal convergence rates with this finite element choice.

5.10 Melt Linear Solver

We now consider the linear solver used in Heister & Dannberg [32] where they obtained the linear system

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{B}^T \\ \mathbf{B} & \mathbf{N} & \mathbf{0} \\ \mathbf{B} & \mathbf{0} & \mathbf{K} \end{pmatrix} \begin{pmatrix} \mathbf{U}_s \\ \mathbf{P}_f \\ \mathbf{P}_c \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \\ \mathbf{0} \end{pmatrix}, \tag{5.33}$$

where $\mathbf{N}$ is the discretization of $-(K_D \nabla p_f, \nabla q_f)$ in the incompressible case and $\mathbf{K}$ is given by $-\left(\frac{1}{\xi} p_c, q_c\right)$.

Based on the work of Rhebergen et al. [92], Heister & Dannberg [32] solved the block system (5.33) using flexible GMRES with the upper block triangular preconditioner (preconditioned from the right)

$$\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{B}^T \\ \mathbf{0} & \mathbf{X} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{Y} \end{pmatrix}^{-1}.$$

Heister & Dannberg [32] chose

$$\hat{\mathbf{X}} = -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D \mathbf{L}_{p_f} \approx -\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{N} = \mathbf{X}$$

and

$$\hat{\mathbf{Y}} = -(\frac{1}{\eta} + \frac{1}{\xi})\mathbf{M}_{p_c} \approx -\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{K} = \mathbf{Y},$$

where $\mathbf{M}_*$ and $\mathbf{L}_*$ are mass and stiffness matrices, respectively.

As stated in Heister & Dannberg [32], "The approximation for $\mathbf{A}^{-1}$ is done using

an inner CG solver with a relative tolerance of 10^{-2} preconditioned by Trilinos ML applied to the diagonal blocks of $\mathbf{A}$. The Schur complement solves for $\hat{\mathbf{X}}^{-1}$ and $\hat{\mathbf{Y}}^{-1}$ are also done using CG preconditioned by a block ILU(0)."

5.10.1 Another Approach

Unfortunately, that approach used in Heister & Dannberg [32] can be made better as a better preconditioner should have the Schur complement $S = BA^{-1}B^T$ in two spots due to Gaussian elimination, and thus should actually be

$$\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{B}^T \\ \mathbf{0} & \mathbf{N} - S & -S \\ \mathbf{0} & -S & \mathbf{K} - S \end{pmatrix}^{-1}.$$

Continuing the use of Gaussian elimination one can see that this is equivalent to Rhebergen et al.'s [92] ideal preconditioner of

$$\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{B}^T \\ \mathbf{0} & \mathbf{N} - S & -S \\ \mathbf{0} & \mathbf{0} & \mathbf{K} - S - S(N-S)^{-1}S \end{pmatrix}^{-1}.$$

Rhebergen et al. [92] then proceed to take the inverse of $\mathbf{K} - S$ and $\mathbf{K} - S - S(N - S)^{-1}S$ whereas we instead take the inverse of the bottom right 2×2 matrix of our preconditioner, which we call Q^{-1}. Recall that $\mathbf{N} - S = -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f}$ and

$\mathbf{K} - S = -(\frac{1}{\eta} + \frac{1}{\xi})\mathbf{M}_{p_c}$, and therefore our preconditioner can be written as:

$$\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \mathbf{B}^T \\ \mathbf{0} & -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \\ \mathbf{0} & -\frac{1}{\eta}\mathbf{M}_{p_c} & -(\frac{1}{\eta} + \frac{1}{\xi})\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

As previously mentioned, we then need to take the bottom left 2×2 matrix and find the inverse of it. One can see that if $\alpha = \frac{\xi}{\eta} \to \infty$ and η is set to be a constant, then

$$\mathbf{Q}^{-1} = \begin{pmatrix} -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \\ -\frac{1}{\eta}\mathbf{M}_{p_f} & -(\frac{1}{\eta} + \frac{1}{\xi})\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

becomes

$$\mathbf{Q}^{-1} = \begin{pmatrix} -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \\ -\frac{1}{\eta}\mathbf{M}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

which becomes

$$\mathbf{Q}^{-1} = \begin{pmatrix} -\frac{1}{\eta}\mathbf{M}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \\ -\frac{1}{\eta}\mathbf{M}_{p_f} & -\frac{1}{\eta}\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

as $K_D \to 0$, which is singular and we can no longer expect a solution from our iterative solvers.

5.10.2 Arbogast-inspired Idea

We need a way to retain solvability when $K_D \to 0$. If we replace p_c with $\bar{p_c}$, where $p_c = \sqrt{K_D} \cdot \bar{p_c}$, and multiply the bottom equation by $\sqrt{K_D}$ then, for the linear system, we get

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \sqrt{K_D}\mathbf{B}^T \\ \mathbf{B} & \mathbf{N} & \mathbf{0} \\ \sqrt{K_D}\mathbf{B} & \mathbf{0} & K_D\mathbf{K} \end{pmatrix} \begin{pmatrix} \mathbf{U}_s \\ \mathbf{P}_f \\ \mathbf{P}_c \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \\ \mathbf{0} \end{pmatrix}, \tag{5.34}$$

which results in the new preconditioner

$$\mathbf{P}^{-1} = \begin{pmatrix} \mathbf{A} & \mathbf{B}^T & \sqrt{K_D}\mathbf{B}^T \\ \mathbf{0} & -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f} & -\frac{\sqrt{K_D}}{\eta}\mathbf{M}_{p_c} \\ \mathbf{0} & -\frac{\sqrt{K_D}}{\eta}\mathbf{M}_{p_c} & -K_D(\frac{1}{\eta} + \frac{1}{\xi})\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

This idea was motivated by Arbogast et al [4]. As done above, looking at the bottom left 2×2 matrix and find the inverse of it for our new method looks like:

$$\mathbf{Q}^{-1} = \begin{pmatrix} -\frac{1}{\eta}\mathbf{M}_{p_f} - K_D\mathbf{L}_{p_f} & -\sqrt{K_D}\frac{1}{\eta}\mathbf{M}_{p_c} \\ -\sqrt{K_D}\frac{1}{\eta}\mathbf{M}_{p_c} & -K_D\frac{1}{\eta}\mathbf{M}_{p_c} \end{pmatrix}^{-1}.$$

which becomes

$$\mathbf{Q}^{-1} = \begin{pmatrix} -\frac{1}{\eta}\mathbf{M}_{p_f} & 0 \\ 0 & 0 \end{pmatrix}^{-1}.$$

as $K_D \to 0$. In practice, we constrain entries of $p_c = 0$ if $\sqrt{kd}$ vanishes, which allows

us to compute Q^{-1} as it is no longer singular.

5.11 Numerical Results

Here we want to use $Q_2Q_1Q_1$ elements for $k_D = 1$ using a modified ASPECT that uses AMG instead of ILU in the preconditioner for S (in order to ensure a fair comparison between the two methods) against our Arbogast-inspired method to check convergence rates as well as look at the number of iterations of the solver as we increase α and decrease h. We use all of the files under `Files/melt_solver_kd1` from the Github repository named `dissertation` of user `rrgrove6`. Note that numerical results for $k_D = 0$ are omitted from this thesis as both approaches gave the same convergence rates and iterations counts even when increasing α or decreasing h.

5.11.1 Convergence Rates of Arbogast-inspired Idea

In Table 5.4, it can be seen that our Arbogast-inspired approach has the same convergence rates as our old approach for $k_D = 1$ and $\alpha = 1$.

Table 5.4: L^2 convergence rates of both approaches with $k_D = 1$ and $\alpha = 1$

	$Q_2Q_1Q_1$					
h	**u**	ratio	p_f	ratio	p_c	ratio
3.5e-1	2.2e-3	-	1.5e-2	-	5.4e-3	-
1.8e-1	5.4e-4	4.0	3.7e-3	4.0	1.4e-3	4.0
8.8e-2	1.4e-4	4.0	9.1e-4	4.0	3.4e-4	4.0
4.4e-2	3.4e-5	4.0	2.3e-4	4.0	8.4e-5	4.0
2.2e-2	8.5e-6	4.0	5.7e-5	4.0	2.1-5	4.0

5.11.2 Iteration Counts

Table 5.5 and Table 5.6 show the iteration counts for approach used in ASPECT currently and the Arbogast-inspired approach, respectively, with $k_D = 1$ and $Q_2Q_1Q_1$ elements. We only look at the number of iterations of the solver here. It can be seen that our Arbogast-inspired approach handles α becoming larger better than the approach used in ASPECT currently. Also, as h becomes smaller, the number of outer iterations for the new method is becoming constant, which is not true for the approach used in ASPECT currently. This means that as h continues to grow smaller, our new method will continue to best the approach used in ASPECT currently.

Table 5.5: Iteration counts for approach used in ASPECT currently with AMG for S and $k_D = 1$ and $Q_2Q_1Q_1$ elements

	$\alpha = 1$			$\alpha = 10$			$\alpha = 100$			$\alpha = 1000$		
DoFs	outer	S_A	S_S	outer	S_A	S_S	outer	S_A	S_S	outer	S_A	S_S
1,977	15	16	16	22	23	23	24	25	25	24	25	25
7,401	16	74	17	24	104	25	27	116	28	27	117	28
28,617	16	81	62	25	116	99	27	126	108	27	126	108
112,521	16	82	65	25	126	104	28	139	116	28	139	116
446,217	15	86	69	27	143	123	32	167	143	34	179	152

Table 5.6: Iteration counts for our Arbogast-inspired approach with $k_D = 1$ and $Q_2Q_1Q_1$ elements

	$\alpha = 1$			$\alpha = 10$			$\alpha = 100$			$\alpha = 1000$		
DoFs	outer	S_A	S_S	outer	S_A	S_S	outer	S_A	S_S	outer	S_A	S_S
1,977	12	13	13	16	17	17	18	19	19	18	19	19
7,401	14	70	15	21	96	22	23	105	24	24	107	25
28,617	14	74	69	22	112	96	24	120	103	24	120	103
112,521	14	74	75	22	113	111	24	121	114	25	126	118
446,217	14	77	75	22	115	118	25	129	131	25	129	131

5.12 Conclusions

If you have a problem where $k_D > 0$, our current suggestion would be to pick Q_2, Q_1, Q_1 elements as they are stable even though you will get suboptimal rates (done in ASPECT now), but one should pick Q_2, Q_2, Q_1 elements if you have $k_D = 0$ everywhere. As for which solver to use, the Arbogast-inspired approach handles increasing α and decreasing h better than the approach used in ASPECT currently for $k_D = 1$. The Arbogast-inspired approach also has the advantage that a smaller amount of expensive A iterations are done at the cost of a larger amount of cheaper S iterations.

Chapter 6

Conclusions

For the GMG for Stokes work, we have shown that applying GMG to the velocity block while solving Stokes is competitive in serial computations in terms of performance and memory usage to UMFPACK, ILU, and AMG. This implies that it will outperform the other methods (especially UMFPACK and ILU) as our systems grow larger and in parallel computations. Additionally, GMG can be parallelized like AMG so it is much more competitive than UMFPACK or ILU for bigger problems. This work is in a good state to serve as a template or starting point for the research of others, as everything has been well documented and the code has been made available.

For the Schwarz smoothers for conforming inf-sup stable discretizations of the Stokes equations, our goal was to extend Kanschat's work to include include $Q_{k+1} \times DGP_k$ elements but we also look at numerical results for Taylor Hood ($Q_{k+1} \times Q_k$), $Q_{\text{Bubble}}(k+1) \times Q_k$, and $Q_{k+1} \times Q_k + DG_0$ elements, that is, we wanted to show that Schwarz methods can be used as multigrid smoother for the Stokes equations using conforming and inf-sup stable discretization spaces, and that the iteration counts are sufficiently

small. We have strong numerical evidence to support that we can do this for the $Q_k \times DGP_{k-1}$ elements, but the analysis is not complete, as we need justification for the step

$$\sum_v a_l(\boldsymbol{u}_v^\perp, \boldsymbol{u}_v^\perp) \leq C a_l(\boldsymbol{u}_l^\perp, \boldsymbol{u}_l^\perp)$$

which is the main difficulty. By applying the GMG preconditioner to the entire system matrix for Stokes, we hoped to get much better numbers for our iteration counts than we did when we just applied the GMG preconditioner to the velocity block of Stokes in Chapter 3. This seems to be the case, and if someone, in the future, finds an efficient way to handle patch-based smoothers (as right now there it is just too expensive build all of the local inverses), then this work could be a stepping stone towards revolutionizing fluid flow solvers.

For the Three-field Stokes work, if you have a problem where $k_D > 0$, our current suggestion would be to pick Taylor Hood ($Q_2 \times Q_1 \times Q_1$) elements as they are stable even though you will get suboptimal rates (done in ASPECT now), but one should pick $Q_2 \times Q_2 \times Q_1$ elements if you have $k_D = 0$ everywhere. We also improved the way ASPECT handles the melt linear solver and preconditioner so that a smaller amount of expensive A iterations are done at the cost of a larger amount of cheaper S iterations.

References

[1] M. Adams. A distributed memory unstructured Gauss-Seidel algorithm for multigrid smoothers. In *Supercomputing, ACM/IEEE 2001 Conference*, pages 14–14. IEEE, 2001.

[2] R. Altmann and J. Heiland. Finite element decomposition and minimal extension for flow equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(5):1489–1509, 2015.

[3] G. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. In *Proceedings of the April 18-20, 1967, spring joint computer conference*, pages 483–485. ACM, 1967.

[4] T. Arbogast, M. Hesse, and A. Taicher. Mixed methods for two-phase Darcy–Stokes mixtures of partially melted materials with regions of zero porosity. *SIAM Journal on Scientific Computing*, 39(2):B375–B402, 2017.

[5] D. Arndt. *Augmented Taylor-Hood Elements for Incompressible Flow*. PhD thesis, 2013.

[6] D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Helta, M. Kronbichler, M. Maier, J. Pelteret, B. Turcksin, and D. Wells. The deal. II library, version 8.5. *Journal of Numerical Mathematics*, 2017.

[7] O. Axelsson and V. Barker. *Finite Element Solution of Boundary Value Problems: Theory and Computation*, volume 35. SIAM, 1984.

[8] Z. Bai. Block preconditioners for elliptic PDE-constrained optimization problems. *Computing*, 91(4):379–395, 2011.

[9] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang. PETSc. *See http://www. mcs. anl. gov/petsc*, 2001.

[10] W. Bangerth. Finite element methods in scientific computing. `http://www.math.colostate.edu/~bangerth/videos/676/slides.33.25.pdf`.

[11] W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler. Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Transactions on Mathematical Software (TOMS)*, 38(2):14, 2011.

[12] W. Bangerth, J. Dannberg, R. Gassmoeller, T. Heister, et al. ASPECT v1.5.0, mar 2017. doi:10.5281/zenodo.344623.

[13] W. Bangerth and R. Hartmann. deal.II – step-7 tutorial program. `https://www.dealii.org/8.4.0/doxygen/deal.II/step_7.html`.

[14] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a General Purpose Object Oriented Finite Element Library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[15] W. Bangerth, T. Heister, et al. ASPECT*: Advanced Solver for Problems in Earth's ConvecTion v1.4.0*. Computational Infrastructure for Geodynamics, 2016.

[16] M. Benzi, G. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, 2005.

[17] M. Berry and R. Plemmons. Algorithms and experiments for structural mechanics on high-performance architectures. *Computer Methods in Applied Mechanics and Engineering*, 64(1-3):487–507, 1987.

[18] D Braess and W. Hackbusch. A new convergence proof for the multigrid method including the v-cycle. *SIAM Journal on Numerical Analysis*, 20(5):967–975, 1983.

[19] D. Braess and R. Verfürth. Multigrid methods for nonconforming finite element methods. *SIAM Journal on Numerical Analysis*, 27(4):979–986, 1990.

[20] J. Bramble. *Multigrid Methods*, volume 294. CRC Press, 1993.

[21] J. Bramble and J. Pasciak. The analysis of smoothers for multigrid algorithms. *Mathematics of Computation*, 58(198):467–488, 1992.

[22] J. Bramble, J. Pasciak, and A. Vassilev. Analysis of the inexact Uzawa algorithm for saddle point problems. *SIAM Journal on Numerical Analysis*, 34(3):1072–1092, 1997.

[23] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*, volume 15. Springer Science & Business Media, 2012.

[24] O. Bröker and M. Grote. Sparse approximate inverse smoothers for geometric and algebraic multigrid. *Applied Numerical Mathematics*, 41(1):61–80, 2002.

[25] C. Canuto, A. Quarteroni, M. Hussaini, and T. Zang. *Fundamentals of Fluid Dynamics*. Springer, 2007.

[26] Z. Cao. Augmentation block preconditioners for saddle point-type matrices with singular (1, 1) blocks. *Numerical Linear Algebra with Applications*, 15(6):515–533, 2008.

[27] R. Chan and X. Jin. A family of block preconditioners for block systems. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1218–1235, 1992.

[28] L. Chen. Multigrid methods for saddle point systems using constrained smoothers. *Computers & Mathematics with Applications*, 70(12):2854–2866, 2015.

[29] T. Clevenger. Partitioning of Parallel Adaptive Geometric Multigrid. Master's thesis, Clemson University, USA, 2016.

[30] D. Arnold and R. Falk and R. Winther. Preconditioning in H (div) and applications. *Mathematics of Computation of the American Mathematical Society*, 66(219):957–984, 1997.

[31] H. Dallmann, D. Arndt, and G. Lube. Local projection stabilization for the Oseen problem. *IMA Journal of Numerical Analysis*, 36(2):796–823, 2015.

[32] J. Dannberg and T. Heister. Compressible magma/mantle dynamics: 3-D, adaptive simulations in ASPECT. *Geophysical Journal International*, 207(3):1343–1366, 2016.

[33] H. Darcy. Les fontaines publique de la ville de Dijon. *Dalmont, Paris*, 647, 1856.

[34] T. Davis. Algorithm 832: UMFPACK V4. 3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software (TOMS)*, 30(2):196–199, 2004.

[35] A. de Niet and F. Wubs. Two preconditioners for saddle point problems in fluid flows. *International Journal for Numerical Methods in Fluids*, 54(4):355, 2007.

[36] H. Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[37] C. Doering and J. Gibbon. *Applied Analysis of the Navier-Stokes Equations*, volume 12. Cambridge University Press, 1995.

[38] E. Duchin and D. Szyld. Application of sparse matrix techniques to inter-regional input-output analysis. *Economics of Planning*, 15(2-3):142–167, 1979.

[39] H. Elman. Preconditioning for the steady-state Navier–Stokes equations with low viscosity. *SIAM Journal on Scientific Computing*, 20(4):1299–1316, 1999.

[40] H. Elman, V. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro. Block preconditioners based on approximate commutators. *SIAM Journal on Scientific Computing*, 27(5):1651–1668, 2006.

[41] H. Elman, D. Silvester, and A. Wathen. Performance and analysis of saddle point preconditioners for the discrete steady-state Navier-Stokes equations. *Numerische Mathematik*, 90(4):665–688, 2002.

[42] H. Elman, D. Silvester, and A. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Oxford University Press (UK), 2014.

[43] A. Ern and J. Guermond. *Theory and Practice of Finite Elements*, volume 159. Springer Science & Business Media, 2013.

[44] X. Feng and O. Karakashian. Two-level additive schwarz methods for a discontinuous galerkin approximation of second order elliptic problems. *SIAM Journal on Numerical Analysis*, 39(4):1343–1365, 2001.

[45] X. Feng and C. Lorton. On Schwarz Methods for Nonsymmetric and Indefinite Problems. *arXiv preprint arXiv:1308.3211*, 2013.

[46] K. Fidkowski, T. Oliver, J. Lu, and D. Darmofal. p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier–Stokes equations. *Journal of Computational Physics*, 207(1):92–113, 2005.

[47] P. Fischer and J. Lottes. Hybrid Schwarz-multigrid methods for the spectral element method: Extensions to Navier-Stokes. In *Domain Decomposition Methods in Science and Engineering*, pages 35–49. Springer, 2005.

[48] M. Furuichi, D. May, and P. Tackley. Development of a Stokes flow solver robust to large viscosity jumps using a Schur complement approach with mixed precision arithmetic. *Journal of Computational Physics*, 230(24):8835–8851, 2011.

[49] J. Gerbeau and C. Farhat. The Finite Element Method for Fluid Mechanics. `http://web.stanford.edu/class/cme358/notes/cme358_lecture_notes_3.pdf`.

[50] V. Girault and P. Raviart. Finite Element Methods for Navier-Stokes Equations, Theory and Algorithms, 1986.

[51] R. Grove and T. Heister. The deal.II tutorial step-56: Geometric Multigrid for the Stokes Equations, March 2017.

[52] J. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44):6011–6045, 2006.

[53] E. Haber and J. Modersitzki. Numerical methods for volume preserving image registration. *Inverse Problems*, 20(5):1621, 2004.

[54] E. Hall. *Computer Image Processing and Recognition*. Elsevier, 1979.

[55] E. Haynsworth. On the Schur complement. Technical report, Basel Univ (Switzerland) Mathemtics Inst, 1968.

[56] T. Heister. *A massively parallel finite element framework with application to incompressible flows*. PhD thesis, Niedersächsische Staats-und Universitätsbibliothek Göttingen, 2011.

[57] T. Heister and G. Rapin. Efficient augmented Lagrangian-type preconditioning for the Oseen problem using Grad-Div stabilization. *International Journal for Numerical Methods in Fluids*, 71(1):118–134, 2013.

[58] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, et al. An overview of the Trilinos project. *ACM Transactions on Mathematical Software (TOMS)*, 31(3):397–423, 2005.

[59] M. Hestenes and E. Stiefel. *Methods of Conjugate Gradients for Solving Linear Systems*, volume 49. NBS, 1952.

[60] J. Heys, T. Manteuffel, S. McCormick, and L. Olson. Algebraic multigrid for higher-order finite elements. *Journal of Computational Physics*, 204(2):520–532, 2005.

[61] F. Hülsemann, M. Kowarschik, M. Mohr, and U. Rüde. Parallel geometric multigrid. In *Numerical Solution of Partial Differential Equations on Parallel Computers*, pages 165–208. Springer, 2006.

[62] B. Janssen and G. Kanschat. Adaptive Multilevel Methods with Local Smoothing for H^1-and H^curl-Conforming High Order Finite Element Methods. *SIAM Journal on Scientific Computing*, 33(4):2095–2114, 2011.

[63] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Courier Corporation, 2012.

[64] G. Kanschat, B. Janssen, and W. Bangerth. deal.II – step-16 tutorial program. `https://www.dealii.org/8.4.0/doxygen/deal.II/step_16.html`.

[65] G. Kanschat and Y. Mao. Multigrid methods for Hdiv-conforming discontinuous Galerkin methods for the Stokes equations. *Journal of Numerical Mathematics*, 23(1):51–66, 2015.

[66] O. Karakashian. On a Galerkin-Lagrange multiplier method for the stationary Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 19(5):909–923, 1982.

[67] D. Kay and D. Loghin. A Green's function preconditioner for the steady-state Navier-Stokes equations. 1999.

[68] T. Keller, D. May, and B. Kaus. Numerical modelling of magma dynamics coupled to tectonic deformation of lithosphere and crust. *Geophysical Journal International*, 195(3):1406–1442, 2013.

[69] A. Klawonn. Block-triangular preconditioners for saddle point problems with a penalty term. *SIAM Journal on Scientific Computing*, 19(1):172–184, 1998.

[70] A. Klawonn and L. Pavarino. A comparison of overlapping Schwarz methods and block preconditioners for saddle point problems. *Numerical Linear Algebra with Applications*, 7(1):1–25, 2000.

[71] M. Kronbichler and W. Bangerth. deal.II – step-22 tutorial program. `https://www.dealii.org/developer/doxygen/deal.II/step_22.html`.

[72] M. Kronbichler, T. Heister, and W. Bangerth. High Accuracy Mantle Convection Simulation through Modern Numerical Methods. *Geophysics Journal International*, 191:12–29, 2012.

[73] A. Krylov. On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined. *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. mat. i estest. nauk*, 7(4):491–539, 1931.

[74] W. Layton. *Introduction to the Numerical Analysis of Incompressible Viscous Flows*, volume 6. SIAM, 2008.

[75] W. Layton, C. Manica, M. Neda, M. Olshanskii, and L. Rebholz. On the accuracy of the rotation form in simulations of the Navier–Stokes equations. *Journal of Computational Physics*, 228(9):3433–3447, 2009.

[76] P. Lions. On the Schwarz alternating method. I. In *First international symposium on domain decomposition methods for partial differential equations*, pages 1–42. Paris, France, 1988.

[77] H. Markowitz and A. Perold. Sparsity and piecewise linearity in large portfolio optimization problems, 1981.

[78] V. Maz'ya. *Sobolev Spaces*. Springer, 2013.

[79] D. McKenzie. The generation and compaction of partially molten rock. *Journal of Petrology*, 25(3):713–765, 1984.

[80] M. Murphy, G. Golub, and A. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21(6):1969–1972, 2000.

[81] C. Nastase and D. Mavriplis. High-order discontinuous Galerkin methods using an hp-multigrid approach. *Journal of Computational Physics*, 213(1):330–357, 2006.

[82] Y. Notay. A new analysis of block preconditioners for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 35(1):143–173, 2014.

[83] M. Olshanskii, G. Lube, T. Heister, and J. Löwe. Grad–div stabilization and subgrid pressure models for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 198(49):3975–3988, 2009.

[84] M. Olshanskii and A. Reusken. Grad-div stablilization for Stokes equations. *Mathematics of Computation*, 73(248):1699–1718, 2004.

[85] P. Solin, K. Segeth and I. Dolezel. *Higher-order Finite Element Methods*. CRC Press, 2003.

[86] I. Perugia and V. Simoncini. Block-diagonal and indefinite symmetric preconditioners for mixed finite element formulations. *Numerical Linear Algebra with Applications*, 7(7-8):585–616, 2000.

[87] X. Ping, R. Chen, K. Tsang, and E. Yung. The SSOR-preconditioned inner outer flexible GMRES method for the FEM analysis of EM problems. *Microwave and Optical Technology Letters*, 48(9):1708–1712, 2006.

[88] H. Poincaré. *Les méthodes nouvelles de la mécanique céleste: Méthodes de MM. Newcomb, Glydén, Lindstedt et Bohlin. 1893*, volume 2. Gauthier-Villars it fils, 1893.

[89] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations*, volume 23. Springer Science & Business Media, 2008.

[90] P. Raviart and J. Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of Finite Element Methods*, pages 292–315. Springer, 1977.

[91] S. Rhebergen, G. Wells, R. Katz, and A. Wathen. Analysis of block preconditioners for models of coupled magma/mantle dynamics. *SIAM Journal on Scientific Computing*, 36(4):A1960–A1977, 2014.

[92] S. Rhebergen, G. Wells, A. Wathen, and R. Katz. Three-field block preconditioners for models of coupled magma/mantle dynamics. *SIAM Journal on Scientific Computing*, 37(5):A2270–A2294, 2015.

[93] J. Ruge and K. Stüben. Algebraic multigrid. *Multigrid Methods*, 3(13):73–130, 1987.

[94] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[95] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

[96] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[97] J. Schöberl. Multigrid methods for a parameter dependent problem in primal variables. *Numerische Mathematik*, 84(1):97–119, 1999.

[98] J. Schöberl. *Robust multigrid methods for parameter dependent problems*. PhD thesis, Johannes Kepler Universität Linz, 1999.

[99] J. Schöberl and W. Zulehner. On Schwarz-type smoothers for saddle point problems. *Numerische Mathematik*, 95(2):377–399, 2003.

[100] H. Schwarz. Über einige Abbildungsaufgaben, Vierteljahresschrift Naturforsch. *Ges. Zurich*, 15:272–286, 1870.

[101] J. Shen. On error estimates of the penalty method for unsteady Navier-Stokes equations. *SIAM Journal on Numerical Analysis*, 32(2):386–403, 1995.

[102] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier–Stokes equations for incompressible flow. *Journal of Computational and Applied Mathematics*, 128(1):261–279, 2001.

[103] D. Silvester and A. Wathen. Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31(5):1352–1367, 1994.

[104] G. Strang and G. Fix. *An Analysis of the Finite Element Method*, volume 212. Prentice-hall Englewood Cliffs, NJ, 1973.

[105] S. Takacs. A robust multigrid method for the time-dependent Stokes problem. *SIAM Journal on Numerical Analysis*, 53(6):2634–2654, 2015.

[106] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.

[107] R. Temam. *Navier-Stokes Equations: Theory and Numerical Analysis*, volume 343. American Mathematical Soc., 2001.

[108] U. Trottenberg, C. Oosterlee, and A. Schuller. *Multigrid*. Academic press, 2000.

[109] S. Turek. *Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach*, volume 6. Springer Science & Business Media, 1999.

[110] J. Volker. Higher order finite element methods and multigrid solvers in a benchmark problem for the 3D Navier–Stokes equations. *International Journal for Numerical Methods in Fluids*, 40(6):775–798, 2002.

[111] P. Wesseling. *Principles of Computational Fluid Dynamics*, volume 29. Springer Science & Business Media, 2009.

[112] P. Wesseling and C. Oosterlee. Geometric multigrid with applications to computational fluid dynamics. *Journal of Computational and Applied Mathematics*, 128(1):311–334, 2001.

[113] O. Widlund and A. Toselli. Domain decomposition methods-algorithms and theory. In *Computational Mathematics*. Springer, 2004.

[114] N. Wilson. *Physics-based algorithms and divergence free finite elements for coupled flow problems*. PhD thesis, Citeseer, 2012.