

Distribution Agreement

In presenting this thesis or dissertation as a partial fulfillment of the requirements for an advanced degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis or dissertation in whole or in part in all forms of media, now or hereafter known, including display on the world wide web. I understand that I may select some access restrictions as part of the online submission of this thesis or dissertation. I retain all ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Signature:

Alexis Aposporidis

Date

Numerical Analysis of Mixed Formulations for Bingham Fluids

By

Alexis Aposporidis
Doctor of Philosophy

Mathematics and Computer Science

Alessandro Veneziani, Ph.D.
Advisor

Michele Benzi, Ph.D.
Committee Member

James Nagy, Ph.D.
Committee Member

Accepted:

Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

Date

Numerical Analysis of Mixed Formulations for Bingham Fluids

by

Alexis Aposporidis
Master of Science, Emory University, 2010

Advisor: Alessandro Veneziani, Ph.D.

An abstract of
A Dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements of the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2012

Abstract

Numerical Analysis of Mixed Formulations for Bingham Fluids By Alexis Aposporidis

Visco-plastic materials have been attracting a great amount of attention among researchers in the study of fluid flow due to their widespread presence in various fields of science. While numerical techniques for simulating their flow have seen significant improvements in the last few decades, the efficient solution of the nonlinear partial differential equations modelling them still poses many challenges. From a mathematical point of view, the major difficulty associated with the numerical solution of these equations is the presence of singularities in (a priori unknown) parts of the domain. This “irregularity” of the equations generally reflects in slow convergence of numerical solvers. In this thesis we introduce an augmented formulation of the Bingham visco-plastic flow which is aimed at circumventing the singularity of the equations. We develop a nonlinear solver based on this new formulation and compare its performance to other common techniques for solving the Bingham flow, indicating superior convergence properties of the solver based on the new formulation. Upon linearization and discretization of the augmented formulation, a sequence of linear systems is obtained which are in general very large and sparse. We introduce a nonlinear geometric multilevel technique for the efficient solution of these linear systems. The convergence of this multilevel technique is accelerated by a flexible Krylov subspace method. We test the resulting numerical scheme on both academic test cases and problems arising from real-life applications with a particular emphasis on problems in hemodynamics.

Numerical Analysis of Mixed Formulations for Bingham Fluids

by

Alexis Aposporidis
Master of Science, Emory University, 2010

Advisor: Alessandro Veneziani, Ph.D.

A Dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements of the degree of
Doctor of Philosophy
in Mathematics and Computer Science
2012

Acknowledgments

I would like to express my deepest gratitude to Dr. Alessandro Veneziani for his support throughout my time as a graduate student and for sharing his passion for research with me. Alessandro's dedication to his work, his group and his students make him a truly special advisor and I have enjoyed every day working with him.

I would also like to thank my committee, Dr. Michele Benzi and Dr. James Nagy, for their suggestions and comments as well as for their support as faculty advisors of the SIAM Student Chapter.

I would like to thank Dr. Eldad Haber for his idea that started this project and that ultimately led to this thesis. I also want to thank him for his financial support and for being "responsible" for my decision to change to the field of applied mathematics at a relatively late stage of my studies; I have never regretted making this choice.

Further, I would like to thank Dr. Panayot Vassilevski for giving me the opportunity to work at Lawrence Livermore National Laboratory in the summers of 2010 and 2011. The experience I had during these internships and what I learned have been of great value to me.

I extend my gratitude to Dr. Maxim Olshanskii for many great contributions to this work that led to our first paper.

I also want to thank my parents for their support and for having me in their thoughts during these years.

Finally, let me thank Verena for making these years so much more enjoyable. This work would not have been possible without her.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Non-Newtonian Fluids | 1 |
| 1.2 | History of Rheology | 4 |
| 1.3 | Bingham Fluids | 6 |
| 1.3.1 | Motivation and Examples | 6 |
| 1.3.2 | The Equations | 7 |
| 1.4 | Notation | 9 |
| 1.5 | Outline of the Thesis | 9 |
| 2 | Mathematical Formulation | 11 |
| 2.1 | Regularized Formulations | 11 |
| 2.2 | The Formulation by Lions and Glowinski | 14 |
| 2.3 | The Mixed Formulation | 15 |
| 2.4 | Weak Formulations | 16 |
| 3 | Well-Posedness of the Continuous Problem | 19 |
| 3.1 | Preliminaries | 19 |
| 3.2 | Monotonicity Methods | 20 |
| 3.3 | Well-Posedness of the Primitive Formulation | 27 |
| 3.4 | Well-Posedness of the Mixed Formulation | 29 |

| | | |
|----------|--|-----------|
| 4 | Linearization and Discretization | 33 |
| 4.1 | The Picard Linearization Method | 33 |
| 4.1.1 | Well-Posedness of the Picard Iterative Method | 34 |
| 4.1.2 | Error Estimates for the Picard Iterative Scheme | 37 |
| 4.2 | The Newton Method | 39 |
| 4.2.1 | Newton Continuation | 40 |
| 4.3 | Discretization | 41 |
| 4.3.1 | Well-Posedness of the Discrete Mixed Problem | 42 |
| 4.3.2 | Algebraic Properties | 45 |
| 4.4 | A Solver Based on the Formulation by Lions and Glowinski | 48 |
| 5 | Performance of the Augmented Nonlinear Solver | 51 |
| 5.1 | Description of the Test Cases | 51 |
| 5.2 | An Analytical Test Case | 52 |
| 5.3 | The Lid-Driven Cavity | 53 |
| 5.4 | A Computation on Non-Rectangular Domains | 57 |
| 6 | Preconditioning | 61 |
| 6.1 | Multigrid Methods | 62 |
| 6.2 | Domain Decomposition Methods | 66 |
| 6.3 | The General Concept of Preconditioning | 71 |
| 6.4 | Approximating the Non-Regularized Problem | 73 |
| 6.5 | Inexact Factorization | 76 |
| 6.5.1 | Yosida and Chorin-Temam Schemes | 77 |
| 6.5.2 | Application to Bingham Fluids | 78 |
| 6.6 | The Multilevel Preconditioner | 79 |
| 6.6.1 | Interpolation and Restriction | 81 |
| 6.6.2 | Smoothing | 81 |

| | | |
|----------|---|------------|
| 7 | Numerical Results | 83 |
| 7.1 | Experimental Setup | 83 |
| 7.1.1 | Choosing the Regularization Parameter | 85 |
| 7.2 | Benchmark Problems | 87 |
| 7.2.1 | The Flow Between Two Parallel Plates | 87 |
| 7.2.2 | The Lid-Driven Cavity | 90 |
| 7.2.3 | The Steady Navier-Stokes type problem | 90 |
| 7.3 | Applications in Hemodynamics | 93 |
| 7.3.1 | Motivation | 93 |
| 7.3.2 | A Simulation on an Idealized Geometry | 94 |
| 7.3.3 | Simulations on a Real Aneurysms | 96 |
| 8 | Conclusion and Future Work | 101 |
| 8.1 | Conclusion | 101 |
| 8.2 | Future Work | 102 |
| A | Background | 105 |
| A.1 | Solving Linear Systems of Equations | 105 |
| A.1.1 | Direct Methods | 105 |
| A.1.2 | Stationary Iterative Methods | 106 |
| A.1.3 | Krylov Subspace Methods | 107 |
| A.2 | Preconditioning | 108 |
| A.3 | Nonlinear Preconditioning | 110 |
| A.4 | Sobolev and Lebesgue Spaces | 111 |
| | Bibliography | 113 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | The shear stress as a function of the shear rate for Newtonian fluids (yellow), shear thickening fluids (red), shear thinning fluids (green) and visco-plastic (Bingham) fluids (blue). | 2 |
| 1.2 | Relationship between shear stress and shear rate for Bingham (blue), Casson (green) and Herschel-Bulkley (red) fluids. | 4 |
| 4.1 | Degrees of freedom for the P1isoP2-P1-P0isoP2. \bullet are the degrees of freedom for the velocity components, \square for the pressure and \circ for the tensor W . Images by Maxim A. Olshanskii, Department of Mechanics and Mathematics, Moscow M. V. Lomonosov State University. | 45 |
| 5.1 | The numerical error $\ \mathbf{u} - \mathbf{u}_{ex}\ _A = \sqrt{(\mathbf{u} - \mathbf{u}_{ex})^T A (\mathbf{u} - \mathbf{u}_{ex})}$ of the mixed formulation for different sizes of the mesh and different ε . The matrix A is the finite element velocity stiffness matrix. The thin solid line is a reference line for $O(h^2)$, the thin dotted line for $O(h)$ | 54 |
| 5.2 | Top, left: the pressure field of the analytical test case computed with $\tau_s = 0.3$, $h = \frac{1}{128}$ and $\varepsilon = 10^{-5}$. Top, right: velocity error for $\tau_s = 0.1$. Bottom: velocity error for $\tau_s = 0.3$ and $\tau_s = 0.4$ | 55 |

| | | |
|-----|--|----|
| 5.3 | Reduction of the difference $\ \boldsymbol{\tau}^{(k)} - \boldsymbol{\tau}^{(k-1)}\ $ for the first 200 iterations when solving the lid-driven cavity problem with $\tau_s = 2$ and $h = \frac{1}{32}$. Left: ALG with different choices of λ . No convergence occurs in the case of $\lambda = 0.001$. Right: Comparison between the Picard iterations (4.2) with $\varepsilon = 0$ and ALG with $\lambda = 0.01$ | 58 |
| 5.4 | Streamlines of the lid-driven cavity flow ($\tau_s = 2$, $h = \frac{1}{128}$ and $\varepsilon = 10^{-5}$). Solution of the Stokes type problem. Center and Right: solution of the Navier-Stokes type problem computed with $\mathbf{u} _{y=0} = (10, 0)^T$, $\mu = 0.1$ (center) and $\mu = 0.01$ (right). | 58 |
| 5.5 | Velocity streamlines (blue) and predicted yield surfaces (brown) of the lid-driven cavity flow computed for different values of the yield stress τ_s with $h = \frac{1}{64}$. Images by Maxim A. Olshanskii, Department of Mechanics and Mathematics, Moscow M. V. Lomonosov State University. | 59 |
| 5.6 | Plot of $ D\mathbf{u} $ for non-trivial domains. Top: Circular cavity attached to a rectangular pipe. Bottom: An occluded pipe. Results obtained using the mixed non-regularized formulation. | 60 |
| 6.1 | Levels visited by Algorithm 6.2 on four levels of multigrid with $\sigma = 1$ (left) and $\sigma = 2$ (right). | 66 |
| 6.2 | Original figure by Schwarz [90]. | 67 |
| 6.3 | Partitioning of the matrix \mathcal{A} | 69 |
| 6.4 | Absolute values of the eigenvalues of the discrete linearized Bingham matrix \mathcal{A} (blue) and eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ (red) in the analytical test case, where \mathcal{A}_ε is the regularized Bingham matrix with $\varepsilon = 10^{-2}$ | 74 |
| 6.5 | Residual of GMRES for the first 30 iterations for when solving the non-regularized problem preconditioned by the regularized one with different values of ε | 76 |

| | | |
|-----|---|-----|
| 7.1 | Condition number of the block N_ε for different values of ε and h | 87 |
| 7.2 | Streamlines and pressure for the two and three dimensional flow between two parallel plates. The pressure field is indicated by the background color, the streamlines are colored by the magnitude of the velocity field. | 88 |
| 7.3 | Streamlines and pressure field of the lid-driven cavity in two (left) and three (right) dimensions. | 90 |
| 7.4 | Velocity and pressure field of the lid-driven cavity on the Navier-Stokes type problem. | 93 |
| 7.5 | Blood vessel with aneurysm. Left: the original domain. Right: domain obtained when tracing the injected contrast material. Images by Frank Tong, Emory School of Medicine, Emory University. | 95 |
| 7.6 | Idealized blood vessel with aneurism on three geometric multigrid levels. Left: The coarsest level (280 elements), center: one level of refinement (2,240 elements), right: two levels of refinement (17,920 elements). | 96 |
| 7.7 | Streamlines of the unsteady Navier-Stokes type problem in a cylindrical domain with an attached sphere. | 97 |
| 7.8 | Streamlines of a Bingham fluid flow through a blood vessel with an aneurysm. | 99 |
| 7.9 | Streamlines of a Bingham fluid flow through the geometry from Figure 7.5, left. | 100 |

List of Tables

| | | |
|-----|--|----|
| 5.1 | Number of Picard iterations required for reducing the residual by a factor of 10^{-6} in the analytical test case for $\tau_s = 0.3$ for different choices of ε and h in both the formulation in primitive and mixed variables. The last column shows the results for the non-regularized case which is not applicable in the primitive formulation. | 53 |
| 5.2 | The numerical error $\ \mathbf{u} - \mathbf{u}_{ex}\ _A$ of the primitive (left) and mixed (right) formulation for different sizes of the mesh and different values of ε (A is a finite element velocity stiffness matrix). | 54 |
| 5.3 | Number of Picard iterations required for reducing the residual by 10^{-6} for the formulation in primitive variables (left) and the formulation in mixed variables (right) for the lid-driven cavity. For $h = \frac{1}{128}$ and in many cases of $h = \frac{1}{64}$ the iterations in primitive variables do not converge (denoted by -). The non-regularized case $\varepsilon = 0$ is included in the last column for the mixed formulation. | 57 |
| 7.1 | Number of levels of multigrid, number of subdomains, size of each subdomain, size of overlap and the size of the linear system to be solved in two and three dimensions. | 85 |

| | | |
|-----|---|-----|
| 7.2 | Tolerances and number of subdomains used in our experiments: relative linear tolerance, nonlinear tolerance, absolute nonlinear tolerance (all measured in the L^2 -norm), number of subdomains used (d is the spacial dimension and ℓ the number of multigrid levels. | 86 |
| 7.3 | The flow between two parallel plates, an analytical test case in two and three dimensions: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations. Three dimensional results in the third table are (marked by (*)) are obtained on a Sun Microsystems SunFire X4600. | 89 |
| 7.4 | The lid-driven cavity flow in two and three dimensions: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations. The third table (marked by (*)) displays timings obtained on a Sun Microsystems SunFire X4600. | 91 |
| 7.5 | The lid-driven cavity flow for the Navier-Stokes type problem: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations. | 92 |
| 7.6 | Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment. | 96 |
| 7.7 | Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment on a real blood vessel with aneurysm. | 98 |
| 7.8 | Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment on the geometry from Figure 7.5, left. | 100 |

List of Algorithms

| | | |
|-----|--|-----|
| 4.1 | Newton Continuation | 41 |
| 4.2 | ALG | 48 |
| 6.1 | Two-Level Multigrid | 64 |
| 6.2 | Multigrid W-Cycle | 65 |
| 6.3 | Alternating Overlapping Schwarz Method | 68 |
| 6.4 | Multiplicative Schwarz Method | 70 |
| 6.5 | The Multilevel Algorithm | 80 |
| A.1 | Projection Method | 107 |
| A.2 | FGMRES | 110 |

Chapter 1

Introduction

1.1 Non-Newtonian Fluids

In fluid mechanics, *rheology* is the study of the flow of materials. A measurement of a fluid's resistance to deformation under shear stress is given by its *viscosity*. Viscosity can be perceived as the fluid's "thickness" – the higher the viscosity the "thicker" the fluid. Some fluids exhibit a constant viscosity under a fixed temperature, we call them *Newtonian fluids*. While many common fluids, such as water, milk and ethanol [79], fall into the category of Newtonian fluids, a much larger group of materials show a viscosity which changes with the shear stress and are referred to as *Non-Newtonian fluids*. Three general categories of Non-Newtonian fluids exist, characterized by the manner in which their behavior deviates from that of a Newtonian [22]:

- shear thickening (dilatant) fluids,
- shear thinning (pseudoplastic) fluids and
- visco-plastic fluids.

Figure 1.1 shows the relationship between the shear rate and the shear stress for these three categories.

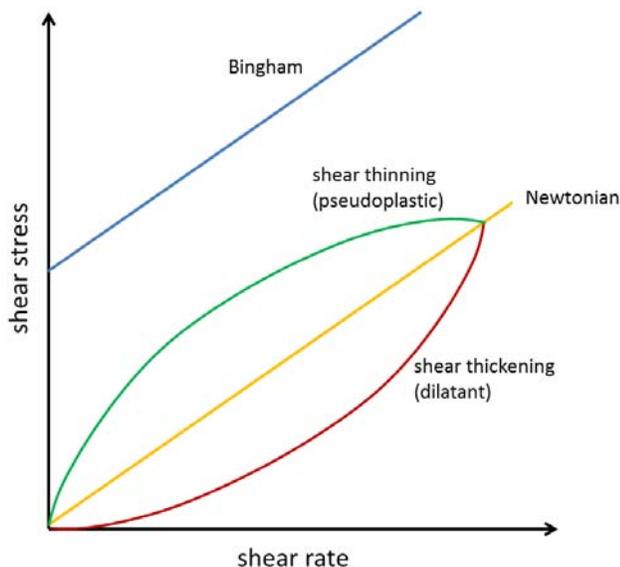


Figure 1.1: The shear stress as a function of the shear rate for Newtonian fluids (yellow), shear thickening fluids (red), shear thinning fluids (green) and visco-plastic (Bingham) fluids (blue).

For shear thickening fluids the viscosity increases as the shear stress increases in magnitude, the material becomes “thicker” under stress. An example of a shear thickening fluid is a 40% corn starch solution [92]. This is a popular example of a Non-Newtonian fluid that is often used in television, literature and on the internet. The effect of the Non-Newtonian behavior is demonstrated by having a person walk over a tub filled with the corn starch solution. As long as the person applies a sufficient amount of force to the surface when stepping on it, the person is able to walk over the fluid without sinking. Once this person stands on the fluid or decreases the force when stepping on the surface, the viscosity of the fluid is no longer high enough to carry the person’s weight, causing the person to sink (see for example [103, 104]).

Shear thinning fluids, on the other hand, exhibit a decreasing viscosity as the shear rate increases. In other terms, they become “thinner” as the applied stress increases in magnitude. Examples of shear-thinning fluids are high fruit concentrates such as orange juice concentrate or apple sauce [92] and blood.

Some materials only show a rate of deformation if the applied stress exceeds a certain threshold. If the applied stress is below this critical value, the material has a vanishing rate of deformation and it behaves like a solid rather than an incompressible fluid. These types of materials are called *visco-plastic* fluids and the critical value that has to be exceeded for it to behave as a fluid is referred to as the *yield stress* [64]. Literature provides three major subcategories of visco-plastic matter, characterized by their relationship between shear stress $\boldsymbol{\tau}$ and shear rate $D\mathbf{u}$ [23, 64]:

- *Casson fluids:*

$$\boldsymbol{\tau} = \left(\sqrt{\mu} + \sqrt{\frac{\tau_s}{|D\mathbf{u}|}} \right)^2 D\mathbf{u} \quad \text{for } |\boldsymbol{\tau}| > \tau_s.$$

Examples of Casson fluids include tomato soup, honey and concentrated fruit juices [54].

- *Herschel-Bulkley fluids:*

$$\boldsymbol{\tau} = \left(K|D\mathbf{u}|^{n-1} + \frac{\tau_s}{|D\mathbf{u}|} \right) D\mathbf{u} \quad \text{for } |\boldsymbol{\tau}| > \tau_s.$$

Minced fish paste or raisin paste are examples of Herschel-Bulkley fluids [92].

- *Bingham fluids:*

$$\boldsymbol{\tau} = \left(2\mu + \frac{\tau_s}{|D\mathbf{u}|} \right) D\mathbf{u} \quad \text{for } |\boldsymbol{\tau}| > \tau_s.$$

The flow of many materials can be described by the Bingham model, examples fluids are ketchup and toothpaste, the flow of certain oil-bearing materials, mud and magma [66], just to name a few.

In all three cases,

$$D\mathbf{u} = 0 \quad \text{for } |\boldsymbol{\tau}| \leq \tau_s,$$

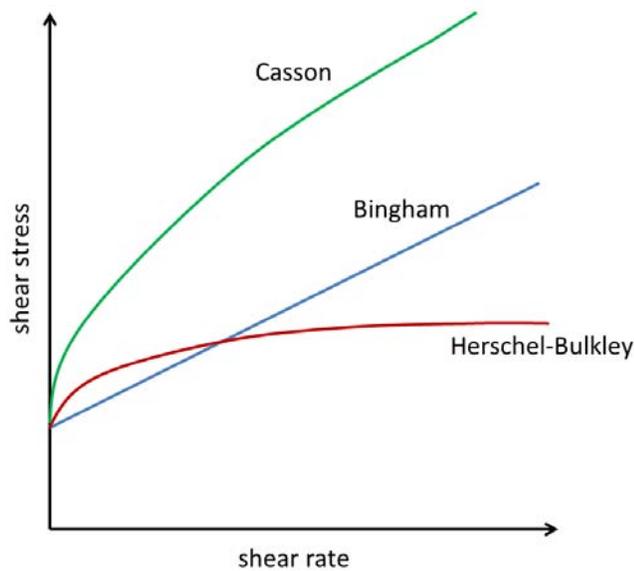


Figure 1.2: Relationship between shear stress and shear rate for Bingham (blue), Casson (green) and Herschel-Bulkley (red) fluids.

$K > 0$ and $n > 0$ are given constants and μ , τ_s are the constant static viscosity and the yield stress, respectively. All these constants are material specific. See Figure 1.2 for a visualization of the relationship between shear rate and shear stress for Bingham, Casson and Herschel-Bulkley fluids.

This thesis will focus on the analysis and the numerical simulation of the flow of Bingham materials. The extension of the ideas presented here to the other types of visco-plastic materials is part of the future work.

1.2 History of Rheology

Scientists have studied fluids for a long time, reaching as far back as antiquity. Probably the earliest application of viscosity effects was performed around 1600 BCE by the Egyptian scientist Amenemhet, who made a correction to the drainage angle of a water clock to account for the viscosity change of water as the temperature changes [19]. In 1687, Isaac Newton stated a famous definition of the resistance of ideal fluids (now

called viscosity): *“The resistance which arises from the lack of slipperiness originating in a fluid – other things being equal – is proportional to the velocity by which the parts of the fluid are separated from each other”* [67]. Newton characterized the class of fluids we now call Newtonian. The study of fluids, however, has not been a separate discipline until the early twentieth century. In 1924, a Plasticity Symposium (for the study of viscosity) was held at Lafayette College, Pennsylvania. Due to a high level of interest in the subject, the symposium was held several times in the following years. During the third symposium, in 1929, Eugene C. Bingham and Markus Reiner (both Lafayette College) suggested the creation of a dedicated discipline for the “study of the flow and the deformation of all forms of matter”, and suggested calling it “rheology”. A committee, which included Winslow H. Herschel, met on April 29th 1929 and decided to follow their suggestion. It was that same day that the field of rheology was coined [37]. The class of visco-plastic materials had still not received much attention during that time, and this did not change for the next few decades, until the early eighties. In 1983, a review by Bird et al. [23] appeared providing a list of several materials exhibiting a yield, making the scientific community more aware of the widespread presence of visco-plastic materials – and their potential applications. Since the appearance of this article, the interest in visco-plastic materials slowly increased and researchers started to study them more thoroughly. Attempts were made to model and simulate their flow. However, measuring the yield stress and finding the unyielded regions were still unresolved problems. At the same time, some researchers questioned the existence of visco-plasticity. Barnes and Walters [9] stated that if a material flows at high stresses it would also flow for low stresses, even though very slowly. This was however quickly disputed by many [7, 53, 88]. The appearance of the modification of the Bingham model by Papanastasiou [70] in 1987 made the implementation of numerical solvers relatively easy. The assumption in his modification was that visco-plastic materials can be approximated as fluids that exhibit an infinitely

high viscosity at the limit of low shear rates, followed by a continuous transition to a viscous fluid. The approximation could be controlled by a material parameter that controls the exponential growth of the stress. The Papanastasiou modification and its variants are still used today and are implemented in all major packages for *Computational Fluid Dynamics* (CFD) for simulating the flow of visco-plastic materials. In the time until today, experimental techniques have been (and still are) experiencing significant improvements. With its applications in different and diverse fields, the field of viscoplasticity enjoys a growing interest among researchers. The subject remains challenging today, with many open problems to be tackled.

See [37] for a full review of the history of rheology and [64] for the particular branch of visco-plastic matter.

1.3 Bingham Fluids

1.3.1 Motivation and Examples

Bingham fluids now enjoy a great amount of attention in the study of visco-plastic materials. The flow of Bingham materials occurs in various fields of modern research, ranging from geology, geophysics to medicine and many other industrial applications. The Bingham model is used, among others, for describing the flow of oil bearing materials, slurries, mud and magma as well as certain powder mixtures. A material often associated with Bingham fluids is ketchup. The visco-plasticity can be observed when turning an opened ketchup bottle upside down. The content will likely remain inside the bottle since the shear stress induced by the earth's gravity is less than the yield stress of the material. To force a flow, the bottle needs to be shaken sufficiently strong so that the stress on the material exceeds the yield stress. This results in the (former rigid) medium to become an incompressible fluid and the content pours out of the bottle [71]. Another material associated with Bingham fluids is toothpaste; a

certain minimal amount of force needs to be applied to the tube before the toothpaste pours out of the tube [73]. Bingham fluids are addressed in various fields of scientific literature. Numerical simulations of Bingham fluid flows are used for analyzing the flow of mud-rich debris and its deposits [101] and for routing the flow of channelized debris. *Magnetorheological fluids* are field-controllable materials whose rheological properties can be altered by applying magnetic fields [49]. These materials are used for example for shock-absorbers and artificial joints and the Bingham model can be used for describing the flow of these fluids [60].

Mud flow is an example of Bingham flow, making the model interesting for applications involving geology. For example in [56], Bingham fluids are used for modeling and predicting landslides. Bingham fluids are of interest to the oil industry; the model is used for analyzing start-up flows of waxy crude oils in pipelines [98, 99]. Bingham fluids may, under certain circumstances, be used for simulating the flow of blood, especially under a high concentration of white blood cells or blood solutes. We will discuss the Bingham model for blood flow problems in more detail in Chapter 7. See also [47] for applications of Bingham fluids involving problems in hemodynamics.

It should be mentioned that the examples listed here are just a small subset of applications and Bingham fluids in many more situations of interest. See [23] for a comprehensive review of these applications.

1.3.2 The Equations

In this section we introduce the partial differential equations that describe the fluid flow of a Bingham plastic. Let \mathbf{u} denote the (two or three dimensional) velocity field and let p be the pressure. We define the rate of deformation tensor $D\mathbf{u} = (\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ with the Frobenius norm $|D\mathbf{u}| = \sqrt{\text{tr}(D\mathbf{u} \cdot D\mathbf{u}^T)}$. Now we define the strain rate tensor $\boldsymbol{\tau}$ by

$$\boldsymbol{\tau} = 2\mu D\mathbf{u} + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|}. \quad (1.1)$$

The constant τ_s is the yield stress, the critical value the shear stress needs to exceed for the Bingham material to become an incompressible fluid. In a given domain Ω , the equations that describe the flow of a Bingham material are given by

$$\begin{cases} \rho[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}] - \nabla \cdot \boldsymbol{\tau} + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad \text{in } \Omega \quad (1.2)$$

when $|\boldsymbol{\tau}| > \tau_s$. For $|\boldsymbol{\tau}| \leq \tau_s$ we set

$$D\mathbf{u} = 0.$$

We call the subset of Ω in which it holds $D\mathbf{u} = 0$ the *plug* or *rigid* region, since the Bingham material behaves like a solid in this region. Outside the rigid region, the material behaves like a fluid and (1.2) holds. This subset is referred to as the *fluid* region. Consequently, the constitutive relation reads

$$D\mathbf{u} = \begin{cases} \mathbf{0} & \text{if } |\boldsymbol{\tau}| \leq \tau_s \text{ (plug region)} \\ \left(1 - \frac{\tau_s}{|\boldsymbol{\tau}|}\right) \frac{\boldsymbol{\tau}}{2\mu} & \text{if } |\boldsymbol{\tau}| > \tau_s \text{ (fluid region)}. \end{cases} \quad (1.3)$$

Equations (1.2), (1.3) can be viewed as a generalization of the Navier-Stokes equations with shear-dependent viscosity $\hat{\mu} = 2\mu + \frac{\tau_s}{|D\mathbf{u}|}$, reducing to the classical Navier-Stokes equations with constant viscosity for $\tau_s = 0$. In the case of a steady flow with a low Reynolds number, we may also neglect the convective term and (1.2) simplifies to

$$\begin{cases} -\nabla \cdot \boldsymbol{\tau} + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad \text{in } \Omega. \quad (1.4)$$

We refer to (1.4) as the steady *Stokes type* equations as opposed to the *Navier-Stokes type* equations (1.2).

1.4 Notation

We introduce some notation that will be used throughout this thesis. The set $L_0^2(\Omega)$ denotes the subspace of $L^2(\Omega)$ functions with zero mean over Ω and $H_0^1(\Omega)$ the subspace of $H^1(\Omega)$ functions with vanishing trace on $\partial\Omega$. The corresponding spaces for two or three dimensional vectors are denoted in bold, i.e. $\mathbf{L}^2(\Omega)$, $\mathbf{H}^k(\Omega)$ or $\mathbf{H}_0^1(\Omega)$. The subspace of $\mathbf{H}_0^1(\Omega)$ of vector-functions that are divergence free is denoted by \mathbf{V} . For tensors whose components are $H^k(\Omega)$ functions we use the notation $\mathcal{H}^k(\Omega)$. For the particular case of symmetric tensors, we use the notation $\mathcal{L}_s^p(\Omega)$, $\mathcal{H}_s^1(\Omega)$. When there is no possibility of confusion, the indication of the domain Ω will be omitted. The norm in H^k is denoted by $\|\cdot\|_k$, the scalar product and the norm in L^2 are denoted by (\cdot, \cdot) and $\|\cdot\|$, respectively. The same norm and product notation is used for the vector and tensor counterparts of H^k and L^2 .

1.5 Outline of the Thesis

The focus of this thesis will be the numerical solution and analysis of the Bingham fluid flow equations. We introduce a new formulation of the problem which we will refer to as the *mixed formulation* of the Bingham fluid flow. In **Chapter 2** we will explore different formulations of the Bingham fluid flow equations, including the formulation in primitive variables, the regularized formulation and the augmented formulation by Lion and Glowinski. We will then introduce the mixed formulation of the problem. In **Chapter 3** we establish the well-posedness of the continuous problem in the primitive and mixed formulation. We start the chapter by presenting the concept of monotonicity methods for nonlinear equations. This concept is then used for proving the well-posedness. **Chapter 4** presents the discretization and linearization of the problem in primitive and mixed variables. We start with the Picard linearization method and address its well-posedness. Error estimates for the Picard

scheme will be provided. We also present the Newton linearization method. We then address the finite element discretization of the problem in primitive and mixed variables and prove the well-posedness. In **Chapter 5** we show a comparison of the performance of the nonlinear solver based on the mixed formulation and the primitive formulation. A comparison to the solver based on the augmented formulation by Lions and Glowinski will be presented as well. These experiments are performed on simple two dimensional test cases. **Chapter 6** presents iterative techniques and preconditioners for solving the large and sparse linear systems arising from the discretization and linearization of the mixed formulation, with a particular emphasis on geometric multilevel preconditioning. Numerical results will be shown in **Chapter 7**. We test our multilevel preconditioner on two and three dimensional problems, including experiments on complex geometries arising from problems in hemodynamics. The thesis concludes with some summarizing remarks and an outlook to future work in **Chapter 8**.

Chapter 2

Mathematical Formulation

In this chapter we present various mathematical formulations of the Bingham fluid flow. We start with different techniques for regularizing the problem in Section 2.1. These techniques force the entire computational domain to be formally a fluid region. In Section 2.2 we present the augmented formulation introduced by Lions and Glowinski, the resulting numerical solver will be addressed in a later chapter. We then proceed to the mixed formulation in Section 2.3. This formulation will be the main focus of this thesis. In Section 2.4 we present the weak formulation of the problem in primitive and mixed variables.

2.1 Regularized Formulations

In Section 1.3 we introduced the partial differential equations (1.2), (1.3) describing the flow of a Bingham material. We will refer to this formulation as the formulation in *primitive variables*. Notice that in the constitutive relation (1.3) the deformation tensor $D\mathbf{u}$ is piecewise defined in fluid and plug region. A major difficulty associated with solving the Bingham equations is that the fluid and plug regions are unknown *a priori* and finding them is part of the problem. Further notice that $\hat{\mu} = 2\mu + \frac{\tau_s}{|D\mathbf{u}|}$ is singular in the plug region where the strain rate tensor vanishes. These difficulties can

be addressed by *regularizing* $\hat{\mu}$. A common regularization technique is the Bercovier-Engelman regularization [16], in which $|D\mathbf{u}|$ is replaced by $|D\mathbf{u}|_\varepsilon = \sqrt{|D\mathbf{u}|^2 + \varepsilon^2}$ for some $\varepsilon > 0$. In practice, regularization techniques replace the plug region by a flow region in which the fluid has a high viscosity. The regularized formulation of the Bingham flow, using the Bercovier-Engelman approach, reads

$$\begin{cases} \rho[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}] - \nabla \cdot (2\mu D\mathbf{u} + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.1)$$

and is imposed in the entire domain Ω . In the case of the steady Stokes type equations, (2.1) reduces to

$$\begin{cases} -\nabla \cdot (2\mu D\mathbf{u} + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (2.2)$$

While the Bercovier-Engelman regularization perturbs $|D\mathbf{u}|$ by a constant parameter ε independent of $|D\mathbf{u}|$, other techniques exist taking into account the size of $|D\mathbf{u}|$. The model of Papanastasiou [70] is such a technique, providing an exponential regularization of (1.1) by introducing a parameter m which controls the exponential growth of the stress tensor. The Papanastasiou variant of (1.1) reads

$$\boldsymbol{\tau} = 2\mu + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|} (1 - e^{-m|D\mathbf{u}|}),$$

which is valid in both fluid and plug region. Here the impact of the perturbation depends on $D\mathbf{u}$: If $D\mathbf{u}$ is large in magnitude, the term $1 - e^{-m|D\mathbf{u}|}$ is negligible whereas it has more impact as $|D\mathbf{u}|$ approaches zero.

Another option for regularizing is to perturb the problem only in those parts of the domain where $|D\mathbf{u}|$ is sufficiently small. An example of such a regularization technique

is

$$|D\mathbf{u}|_{reg(\varepsilon)} = \begin{cases} |D\mathbf{u}| & \text{for } |D\mathbf{u}| \geq 2\varepsilon \\ \frac{|D\mathbf{u}|^2 + 4\varepsilon^2}{4\varepsilon} & \text{for } |D\mathbf{u}| < \varepsilon. \end{cases}$$

The Bercovier-Engelman regularization and the Papanastasiou model are most commonly used for solving the regularized Bingham model, we will use the Bercovier-Engelman approach in this thesis due to its simplicity. However, other approaches exist and we point out in particular two which have appeared in literature:

1. *Tanner and Milthorpe* [94] used a model called the *bi-viscosity model*, having two finite viscosity slopes (that is, combining two Newtonian models): $\boldsymbol{\tau} = 2\mu_0 D\mathbf{u}$ if $|D\mathbf{u}| < \gamma_c$ and $\boldsymbol{\tau} = 2\mu_0 + \tau_s \frac{D\mathbf{u}}{|D\mathbf{u}|}$ if $|D\mathbf{u}| \geq \gamma_c$, where γ_c and μ_0 are chosen parameters.
2. The approach by *Beris et al.* [17] essentially solved the Bercovier-Engelman regularized problem, but took into account the equations for a plastic solid when $|D\mathbf{u}| < \varepsilon$.

Various well-established techniques for the numerical solution of the regularized Bingham model exist, including iterative schemes of Newton or Picard type for linearizing the equations coupled with preconditioned Krylov subspace methods for solving the linear systems. The numerical implementation of the regularized Bingham model is relatively simple in an existing CFD code, however the regularization prevents finding the ‘exact’ visco-plastic solution. In particular, finding plug-regions in the domain is a non-trivial task if $\varepsilon > 0$, see for example [76], or [98,99] for the case of compressible fluids. Thus accurate computations require small values of ε [33, 45, 85]. However, using small values of ε gives rise to an array of computational issues. In particular, the Newton method applied to (4.1) is not robust with respect to ε , the domain of convergence for the Newton method shrinks as $\varepsilon \rightarrow 0$ (see [33] and numerical results in [51, 57]). Both Newton and Picard iterative methods exhibit a severe loss of

efficiency as $\varepsilon \rightarrow 0$ reflecting in slow convergence of both methods.

2.2 The Formulation by Lions and Glowinski

The mathematical and numerical problems associated with solving the Bingham fluid flow equations motivated researchers to consider different formulations of (1.2), (1.3). One approach is based on the variational inequality formulation introduced by Duvaut and Lions [39]. A numerical solver based on this formulation was proposed by Glowinski et. al. (see [34] and the references therein) and is based on Uzawa-type iterative schemes. It has become attractive when solving the non-regularized Bingham equations, especially when it is necessary to find the ‘true’ visco-plastic solution and to determine the plug regions [76, 80]. The idea of this approach is as follows: the solution \mathbf{u} of the Bingham problem is equivalent to the minimization

$$\min_{\mathbf{v} \in H_0^1(\Omega)} J(\mathbf{v}), \text{ with } J(\mathbf{v}) = \mu \int_{\Omega} |D\mathbf{v}|^2 d\mathbf{x} + \tau_s \int_{\Omega} |D\mathbf{v}| d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x}.$$

The goal is now to decouple the nonlinearity and the derivatives by introducing an auxiliary tensor $\boldsymbol{\gamma} = D\mathbf{u}$. Then the extra relation $D\mathbf{u} - \boldsymbol{\gamma} = \mathbf{0}$ is imposed by penalization and use of a Lagrange multiplier. In this way, the solution \mathbf{u} , $\boldsymbol{\tau}$ together with $\boldsymbol{\gamma}$ of the Bingham problem (1.2) is the saddle-point of the following Lagrangian functional:

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{\tau}) = \mu \int_{\Omega} |\boldsymbol{\gamma}|^2 d\mathbf{x} + \tau_s \int_{\Omega} |\boldsymbol{\gamma}| d\mathbf{x} + \int_{\Omega} (D\mathbf{u} - \boldsymbol{\gamma}) : \boldsymbol{\tau} d\mathbf{x} + \lambda \int_{\Omega} |D\mathbf{u} - \boldsymbol{\gamma}|^2 d\mathbf{x} - \int_{\Omega} \mathbf{f} \cdot \mathbf{u} d\mathbf{x}.$$

The constant $\lambda > 0$ is an auxiliary arbitrary parameter. Then

$$\mathcal{L}(\mathbf{u}, \boldsymbol{\gamma}, \boldsymbol{\tau}) = \min_{\boldsymbol{\sigma} \in \mathcal{L}_s^2, \mathbf{v} \in \mathbf{V}} \max_{\boldsymbol{\xi} \in \mathcal{L}_s^2} \mathcal{L}(\mathbf{v}, \boldsymbol{\sigma}, \boldsymbol{\xi}). \quad (2.3)$$

The resulting numerical scheme, which we will refer to as ALG, does not require a regularization $\varepsilon > 0$ and is proven to be convergent [34]. However, the convergence rate depends on the ‘right’ choice of the parameter λ and has not been rigorously established. Numerical experiments suggest that the convergence of this scheme can be very slow. We will provide details on the algorithm in Chapter 4 and a numerical result in Chapter 5.

2.3 The Mixed Formulation

Motivated by the difficulties associated with solving the Bingham equations in the primitive variables (1.2), (1.3) as well as the drawbacks of ALG, we introduce a different formulation of the Bingham fluid flow equations. We consider first the regularized formulation, using the Bercovier-Engelman approach. Define $W = \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}$. Equations (1.2) and (1.3) can then be reformulated into

$$\begin{cases} \rho[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u}] - \nabla \cdot (2\mu D\mathbf{u} + \tau_s W) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \\ D\mathbf{u} - |D\mathbf{u}|_\varepsilon W = \mathbf{0}. \end{cases} \quad (2.4)$$

The idea of adding an auxiliary variable W to the equations was motivated by [26], where a similar approach was successfully applied in the context of an image restoration problem. In this paper, the scalar variable u is the grey-level of an image and the total variation $\int_\Omega |\nabla u|$ is minimized. Using a Tikhonov formulation, this minimization can be written as

$$\min_u \alpha \|\nabla u\|_{L^1} + \frac{1}{2} \|Ku - z\|_{L^2}^2.$$

The corresponding Euler-Lagrange equation is then

$$-\alpha \nabla \cdot \left(\frac{\nabla u}{|\nabla u|} \right) + K^*(Ku - z),$$

and the singularity of the term $\frac{\nabla u}{|\nabla u|}$ is treated by the same idea as presented here. See [45, 46] for analogies between image restoration problems and visco-plastic fluids. Equation (2.4) denotes the mixed formulation of the Bingham fluid flow and will be analyzed and discussed in further detail. Note that the third constraint of (2.4) has been obtained by multiplying both sides of the equation by $|D\mathbf{u}|_\varepsilon$, resulting in a formulation which contains no division. This is particularly relevant when letting $\varepsilon \rightarrow 0$. The corresponding mixed formulation for the Stokes type equations reads

$$\left\{ \begin{array}{l} -\nabla \cdot (2\mu D\mathbf{u} + \tau_s W) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \\ D\mathbf{u} - |D\mathbf{u}|_\varepsilon W = \mathbf{0}. \end{array} \right. \quad (2.5)$$

We introduce this formulation with the intention of enhancing the numerical properties of the regularized formulation of the Bingham fluid flow. Indeed, the numerical experiments that will be presented in Chapter 5 will show that this formulation is efficient for solving the regularized problem and that for a given ε the number of nonlinear iterations is significantly reduced compared to solving the original problem (1.2), (1.3). The fact that the formulation contains no division even makes the mixed formulation applicable to the nonregularized case ($\varepsilon = 0$). In this case this approach and the resulting iterative method compare favorably with ALG.

2.4 Weak Formulations

We now consider the weak formulation of the primitive and mixed formulation. Let us introduce the following bilinear forms:

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} 2\mu D\mathbf{u} : D\mathbf{v} \text{ on } \mathbf{H}_0^1 \times \mathbf{H}_0^1,$$

$$b(p, \mathbf{v}) = - \int_{\Omega} p \nabla \cdot \mathbf{v} \text{ on } L_0^2 \times \mathbf{H}_0^1.$$

To handle the primitive formulation of the Bingham flow, we further define the form

$$a_{\varepsilon}(\mathbf{u}, \mathbf{v}) = a(\mathbf{u}, \mathbf{v}) + \int_{\Omega} \frac{\tau_s}{|D\mathbf{u}|_{\varepsilon}} D\mathbf{u} : D\mathbf{v} \text{ on } \mathbf{H}_0^1 \times \mathbf{H}_0^1.$$

The continuity and coercivity of the form a_{ε} will be addressed in Section 3.3. Similarly, for the mixed formulation we define the bilinear and non-linear forms

$$c(\mathbf{u}, Z) = \int_{\Omega} \tau_s D\mathbf{u} : Z \text{ on } \mathbf{H}_0^1 \times \mathcal{L}_s^2,$$

$$g(|D\mathbf{u}|_{\varepsilon}, W, Z) = \int_{\Omega} \tau_s |D\mathbf{u}|_{\varepsilon} W : Z \text{ on } \mathbf{H}_0^1 \times \mathcal{L}_s^2 \times \mathcal{L}_s^{\infty}.$$

Note that since $\mathbf{u} \in \mathbf{H}_0^1$, we have $D\mathbf{u} \in \mathcal{L}_s^2$ and therefore $W \in \mathcal{L}_s^2$, $|D\mathbf{u}|_{\varepsilon} \in \mathcal{L}_s^2$. Thus we need to take test functions from \mathcal{L}_s^{∞} to ensure that the form g is well-defined, i.e. $Z \in \mathcal{L}_s^{\infty}$. Later in this thesis (see Section 3.4) we will see that $W \in \mathcal{L}_s^{\infty}$ so that the set of test functions can be extended to \mathcal{L}_s^2 . The weak formulation of the regularized primitive Stokes type formulation then reads: Find $\mathbf{u} \in \mathbf{H}_0^1$ and $p \in L_0^2$ such that for any $\mathbf{v} \in \mathbf{H}_0^1$ and $q \in L_0^2$

$$a_{\varepsilon}(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) + b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}). \quad (2.6)$$

For the mixed formulation, the weak formulation of the Stokes type equations reads as follows: Find $\mathbf{u} \in \mathbf{H}_0^1$, $p \in L_0^2$ and $W \in \mathcal{L}_s^2$ such that for any $\mathbf{v} \in \mathbf{H}_0^1$, $q \in L_0^2$ and $Z \in \mathcal{L}_s^{\infty}$

$$a(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) + c(\mathbf{v}, W) + b(q, \mathbf{u}) + c(\mathbf{u}, Z) - g(|D\mathbf{u}|_{\varepsilon}, W, Z) = (\mathbf{f}, \mathbf{v}). \quad (2.7)$$

For the (unsteady) Navier-Stokes type equations, we need to define in addition the form

$$n(\mathbf{u}, \mathbf{v}) = \rho \int_{\Omega} \left[\frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + (\mathbf{u} \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \right] \text{ on } \mathbf{H}_0^1 \times \mathbf{H}_0^1.$$

The weak formulation of the Navier-Stokes type equations in primitive variables then reads: find $\mathbf{u} \in \mathbf{H}_0^1$ and $p \in L_0^2$ such that for all $\mathbf{v} \in \mathbf{H}_0^1$ and $q \in L_0^2$

$$n(\mathbf{u}, \mathbf{v}) + a_{\varepsilon}(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) + b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}). \quad (2.8)$$

Similarly, for the Navier-Stokes type equations in the mixed formulation, we have: find $\mathbf{u} \in \mathbf{H}_0^1$, $p \in L_0^2$ and $W \in \mathcal{L}_s^2$ such that for all $\mathbf{v} \in \mathbf{H}_0^1$, $q \in L_0^2$ and $Z \in \mathcal{L}_s^{\infty}$

$$n(\mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) - b(p, \mathbf{v}) + c(\mathbf{v}, W) + b(q, \mathbf{u}) + c(\mathbf{u}, Z) - g(|D\mathbf{u}|_{\varepsilon}, W, Z) = (\mathbf{f}, \mathbf{v}). \quad (2.9)$$

Chapter 3

Well-Posedness of the Continuous

Problem ¹

In this chapter we will address the well-posedness of the continuous regularized Bingham problem in the primitive and mixed formulation. We start with presenting two inequalities that will be used for our analysis in Section 3.1. In Section 3.2, we explain the concept of monotonicity methods for nonlinear equations, in particular the method of Browder and Minty. This method will be a key component in establishing the well-posedness of the problem. We then proceed to proving well-posedness of the primitive formulation in Section 3.3. The well-posedness of the mixed formulation is based on showing the equivalence (on the continuous level) of the primitive and mixed formulation, the proof is carried out in Section 3.4.

3.1 Preliminaries

For the analysis presented in this chapter there will be two inequalities of importance. From the vector identities $2\nabla \cdot D = \Delta + \nabla \nabla \cdot$ and $\nabla \nabla \cdot = \Delta + \nabla \times \nabla \times$ and by using

¹Results in this chapter have been published in A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: Analysis and numerical solution*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2434-2446

integration by parts we get the Korn type inequalities

$$\|D\mathbf{u}\| \leq \|\nabla\mathbf{u}\| \leq \sqrt{2}\|D\mathbf{u}\| \text{ for all } \mathbf{u} \in \mathbf{H}_0^1, \quad (3.1)$$

stating the equivalence between the L^2 norms of the gradient and its symmetric part. We shall also refer to the Friedrich's inequality (also known as Poincaré's inequality)

$$\|\mathbf{u}\| \leq C_F \|\nabla\mathbf{u}\| \text{ for all } \mathbf{u} \in \mathbf{H}_0^1, \quad (3.2)$$

where C_F is a positive constant depending only on the domain Ω . Throughout this chapter we will assume that the domain Ω is polygonal or $\partial\Omega \in \mathcal{C}^{1,1}$ and that $\mathbf{f} \in \mathbf{L}^2$. Moreover, for the sake of simplicity we assume homogeneous Dirichlet boundary conditions ($\mathbf{u} = \mathbf{0}$ on $\partial\Omega$). The generalization to mixed Dirichlet and Neumann boundary problems is possible. The analysis will be presented for the regularized steady Stokes type case (2.2) and (2.5).

3.2 Monotonicity Methods

We follow Section 9.1 in [42] for explaining the concept of monotonicity. First, we need an important fixed point theorem (see Section 8.1 in [42]).

Theorem 3.1 (Brouwer's Fixed Point Theorem) *Assume*

$$\mathbf{g} : B(0, 1) \rightarrow B(0, 1)$$

is continuous, where $B(0, 1)$ denotes the closed unit ball in \mathbb{R}^d . Then \mathbf{g} has a fixed point, i.e. there is a point $\mathbf{x} \in B(0, 1)$ such that

$$\mathbf{g}(\mathbf{x}) = \mathbf{x}.$$

We now proceed to the monotonicity method. Let $u : \Omega \rightarrow \mathbb{R}$ and consider for the sake of concreteness the quasilinear partial differential equation

$$\begin{cases} -\nabla \cdot \mathbf{a}(Du) = f & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases} \quad (3.3)$$

where the function $f \in L^2(\Omega)$ and the smooth vector field $\mathbf{a} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ are given and $u \in H_0^1(\Omega)$ is the unknown.

Definition 3.1 *A vector field $\mathbf{a} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is called monotone provided*

$$(\mathbf{a}(\mathbf{x}) - \mathbf{a}(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) \geq 0 \quad (3.4)$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$.

We make two assumptions on the vector field \mathbf{a} :

$$|\mathbf{a}(\mathbf{x})| \leq C(1 + |\mathbf{x}|) \quad (3.5)$$

and

$$\mathbf{a}(\mathbf{x}) \cdot \mathbf{y} \geq \alpha|\mathbf{x}|^2 - \beta \quad (3.6)$$

for all $\mathbf{x} \in \mathbb{R}^d$ and some constants $C, \alpha > 0$ and $\beta \geq 0$. Our goal is to build a solution of (3.3) as the limit of finite dimensional approximations. Consider the sequence $\{w_k = w_k(\mathbf{x})\}_k$ of smooth functions so that

$$\{w_k\}_{k=1}^{\infty} \text{ is an orthonormal basis of } H_0^1(\Omega)$$

with the inner product

$$(u, v) = \int_{\Omega} Du \cdot Dv dx.$$

We want to find a function $u_m \in H_0^1(\Omega)$ of the form

$$u_m = \sum_{k=1}^m d_m^k w_k, \quad (3.7)$$

where the coefficients d_m^k should be selected so that

$$\int_{\Omega} \mathbf{a}(Du_m) \cdot Dw_k dx = \int_{\Omega} f w_k dx \quad (3.8)$$

for all $k \in \{1, 2, \dots, m\}$.

Lemma 3.1 (Zeros of a vector field) *Assume that the continuous function $\mathbf{b} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ satisfies*

$$\mathbf{b}(\mathbf{x}) \cdot \mathbf{x} \geq 0 \quad \text{if } |\mathbf{x}| = r \quad (3.9)$$

for some $r > 0$. Then there exists a point $\mathbf{x} \in B(0, r) = \{\mathbf{z} \in \mathbb{R}^d \mid \|\mathbf{z}\| < r\}$ such that

$$\mathbf{b}(\mathbf{x}) = 0.$$

Proof Suppose the statement were not true, that is, $\mathbf{b}(\mathbf{x}) \neq 0$ for all $\mathbf{x} \in B(0, r)$. Define the map $\mathbf{w} : B(0, r) \rightarrow \partial B(0, r)$ by

$$\mathbf{w}(\mathbf{x}) = -\frac{r}{|\mathbf{b}(\mathbf{x})|} \mathbf{b}(\mathbf{x}).$$

Then \mathbf{w} is continuous. By Brouwer's Fixed Point Theorem (Theorem 3.1), there exists a point $\mathbf{z} \in B(0, r)$ with

$$\mathbf{w}(\mathbf{z}) = \mathbf{z}. \quad (3.10)$$

But then $\mathbf{z} \in \partial B(0, r)$ and so (3.9) and (3.10) imply

$$r^2 = \mathbf{z} \cdot \mathbf{z} = \mathbf{w}(\mathbf{z}) \cdot \mathbf{z} = -\frac{r}{|\mathbf{b}(\mathbf{z})|} \mathbf{b}(\mathbf{z}) \cdot \mathbf{z} \leq 0,$$

a contradiction. \square

Theorem 3.2 (Constrction of approximate solutions) *For each integer $m \in \mathbb{N}$ there exists a function u_m of the form (3.7) satisfying the identities (3.8).*

Proof Define the continuous function $\mathbf{b} : \mathbb{R}^m \rightarrow \mathbb{R}^m$, $\mathbf{b} = (b^1, b^2, \dots, b^m)$ by

$$b^k(\mathbf{d}) = \int_{\Omega} \left[\mathbf{a} \left(\sum_{j=1}^m d_j Dw_j \right) \cdot Dw_k - fw_k \right] dx \quad (k \in \{1, 2, \dots, m\}) \quad (3.11)$$

for each point $\mathbf{d} = (d_1, d_2, \dots, d_m) \in \mathbb{R}^m$. Using (3.6) we get

$$\begin{aligned} \mathbf{b}(\mathbf{d}) \cdot \mathbf{d} &= \int_{\Omega} \left[\mathbf{a} \left(\sum_{j=1}^m d_j Dw_j \right) \cdot \left(\sum_{j=1}^m d_j Dw_j \right) - f \left(\sum_{j=1}^m d_j Dw_j \right) \right] dx \\ &\geq \int_{\Omega} \left[\alpha \left| \sum_{j=1}^m d_j Dw_j \right|^2 - \beta - f \left(\sum_{j=1}^m d_j Dw_j \right) \right] dx \\ &= \alpha |\mathbf{d}|^2 - \beta |\Omega| - \sum_{j=1}^m d_j \int_{\Omega} fw_j dx \\ &\geq \frac{\alpha}{2} |\mathbf{d}|^2 - \beta |\Omega| - C \sum_{j=1}^m (f, w_j)_{L^2(\Omega)}. \end{aligned}$$

Now let $u \in H_0^1(\Omega)$ be the solution of the Laplacian $-\Delta u = f$. Then

$$\int_{\Omega} Du \cdot Dw_j dx = \int_{\Omega} fw_j dx \quad \text{for } j \in \{1, 2, \dots, m\}$$

and hence

$$\sum_{j=1}^m (f, w_j)_{L^2(\Omega)}^2 = \sum_{j=1}^m (u, w_j)^2 \leq \|u\|_{H_0^1(\Omega)}^2 \leq C \|f\|_{L^2(\Omega)}^2.$$

Therefore $\mathbf{b}(\mathbf{d}) \cdot \mathbf{d} \geq \frac{\alpha}{2} \|\mathbf{d}\|^2 - C$ for a constant C and, if r is large enough, $\mathbf{b}(\mathbf{d}) \cdot \mathbf{d} \geq 0$ if $\|\mathbf{d}\| = r$. Applying Lemma 3.1 gives $\mathbf{b}(\mathbf{d}) = 0$ for some \mathbf{d} . Then (3.11) implies that u_m defined by (3.7) satisfies (3.8). \square

Next, we provide some uniform estimates.

Theorem 3.3 *There exists a constant C depending only on Ω and \mathbf{a} such that*

$$\|u_m\|_{H_0^1(\Omega)} \leq C (1 + \|f\|_{L^2(\Omega)})$$

for $m \in \mathbb{N}$.

Proof Multiplying (3.8) by d_m^k and summing over k gives

$$\int_{\Omega} \mathbf{a}(Du_m) \cdot Du_m dx = \int_{\Omega} f u_m dx.$$

With (3.6) we have

$$\alpha \int_{\Omega} |Du_m|^2 dx \leq C + \int_{\Omega} f u_m dx \leq C + \varepsilon \int_{\Omega} u_m^2 dx + \frac{1}{4\varepsilon} \int_{\Omega} f^2 dx.$$

With Poincaré's inequality, and choosing $\varepsilon > 0$ small enough, the statement follows.

□

We are now in the position to prove the main result of this section, the existence of a weak solution of (3.3).

Theorem 3.4 *Assume $\mathbf{a}(\cdot, \cdot)$ is monotone. Assume further that $\mathbf{a}(\cdot, \cdot)$ satisfies (3.5) and (3.6). Then there exists a weak solution of the nonlinear problem (3.3).*

Proof Theorem 3.3 implies that we can extract a subsequence $\{u_{m_j}\}_j \subset \{u_m\}_m$ and a function $u \in H_0^1(\Omega)$ such that

$$u_{m_j} \rightharpoonup u \text{ weakly in } H_0^1(\Omega) \tag{3.12}$$

and

$$u_{m_j} \rightarrow u \text{ in } L^2(\Omega) \tag{3.13}$$

(we use here the symbol ‘ \rightharpoonup ’ for weak convergence). It remains to show that u satisfies

$$\int_{\Omega} \mathbf{a}(Du) \cdot Dv dx = \int_{\Omega} f v dx \text{ for all } v \in H_0^1(\Omega).$$

With the continuity condition (3.6), the sequence $\{\mathbf{a}(Du_m)\}_m$ is bounded in $L^2(\Omega, \mathbb{R}^n)$.

This means we may assume

$$\mathbf{a}(Du_m) \rightharpoonup \boldsymbol{\xi} \text{ weakly in } L^2(\Omega, \mathbb{R}^n) \quad (3.14)$$

for some $\boldsymbol{\xi} \in L^2(\Omega, \mathbb{R}^n)$, if necessary by passing to a further subsequence of $\{u_{m_j}\}_j$.

With (3.8),

$$\int_{\Omega} \boldsymbol{\xi} \cdot Dw_k dx = \int_{\Omega} f w_k dx$$

for all $k \in \mathbb{N}$. Thus

$$\int_{\Omega} \boldsymbol{\xi} \cdot Dv dx = \int_{\Omega} f v dx \text{ for each } v \in H_0^1(\Omega). \quad (3.15)$$

Now since \mathbf{a} is monotone,

$$\int_{\Omega} (\mathbf{a}(Du_m) - \mathbf{a}(Dw)) \cdot (Du_m - Dw) dx \geq 0 \quad (3.16)$$

for all $m \in \mathbb{N}$ and all $w \in H_0^1(\Omega)$. Now (3.8) gives the identity

$$\int_{\Omega} \mathbf{a}(Du_m) \cdot Du_m dx = \int_{\Omega} f u_m.$$

Substituting this into (3.16) yields

$$\int_{\Omega} [f u_m - \mathbf{a}(Du_m) \cdot Dw - \mathbf{a}(Dw) \cdot (Du_m - Dw)] dx \geq 0.$$

Let now $j \rightarrow \infty$. With (3.12), (3.13) and (3.14) we get

$$\int_{\Omega} fu - \boldsymbol{\xi} \cdot Dw - \mathbf{a}(Dw) \cdot (Du - Dw) dx \geq 0.$$

Using (3.15) with $v = u$ gives

$$\int_{\Omega} (\boldsymbol{\xi} - \mathbf{a}(Dw)) \cdot D(u - w) dx \geq 0 \quad \text{for all } w \in H_0^1(\Omega).$$

Now fixing $v \in H_0^1(\Omega)$ and setting $w = u - \lambda v$ for $\lambda > 0$ we obtain

$$\int_{\Omega} (\boldsymbol{\xi} - \mathbf{a}(Du - \lambda Dv)) \cdot Dv dx \geq 0.$$

Letting now $\lambda \rightarrow 0$ gives

$$\int_{\Omega} (\boldsymbol{\xi} - \mathbf{a}(Du)) \cdot Dv dx \geq 0 \quad \text{for all } v \in H_0^1(\Omega). \quad (3.17)$$

Replacing v by $-v$ we deduce that in fact the equality above holds and (3.15) together with (3.17) give

$$\int_{\Omega} \mathbf{a}(Du) \cdot Dv dx = \int_{\Omega} f v dx \quad \text{for all } v \in H_0^1(\Omega)$$

and u is a solution of (3.3). □

The use of monotonicity in this proof is the *Method of Browder and Minty*. An additional assumption on \mathbf{a} will guarantee the uniqueness of this weak solution.

Definition 3.2 A vector field $\mathbf{a} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is called *strictly monotone* if

$$(\mathbf{a}(\mathbf{x}) - \mathbf{a}(\mathbf{y})) \cdot (\mathbf{x} - \mathbf{y}) \geq \theta |\mathbf{x} - \mathbf{y}|^2$$

for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and some constant $\theta > 0$.

Theorem 3.5 *Assume $\mathbf{a}(\cdot, \cdot)$ is strictly monotone. Then the solution of (3.3) is unique.*

Proof Assume u and \tilde{u} are two solutions of (3.3). Then

$$\int_{\Omega} \mathbf{a}(Du) \cdot Dv dx = \int_{\Omega} \mathbf{a}(D\tilde{u}) \cdot Dv dx = \int_{\Omega} f v dx,$$

therefore

$$\int_{\Omega} [\mathbf{a}(Du) - \mathbf{a}(D\tilde{u})] \cdot Dv dx = 0$$

for all $v \in H_0^1(\Omega)$. Setting $v = u - \tilde{u}$, since \mathbf{a} is strictly monotone, we get

$$\int_{\Omega} |Du - D\tilde{u}|^2 dx = 0,$$

and hence $u = \tilde{u}$ almost everywhere in Ω . □

3.3 Well-Posedness of the Primitive Formulation

We now proceed to proving the well-posedness of the continuous problem in the primitive formulation. Note first that by using the Korn and Friedrich's inequalities (3.1), (3.2) we get the coercivity, i.e. there exists a constant $c > 0$ such that

$$a_{\varepsilon}(\mathbf{u}, \mathbf{u}) \geq c \|\mathbf{u}\|_1^2 \text{ for all } \mathbf{u} \in \mathbf{H}_0^1 \quad (3.18)$$

and the continuity, i.e.

$$a_{\varepsilon}(\mathbf{u}, \mathbf{v}) \leq \left(2\mu + \frac{\tau_s}{\varepsilon}\right) \|\mathbf{u}\|_1 \|\mathbf{v}\|_1 \text{ for all } \mathbf{u}, \mathbf{v} \in \mathbf{H}_0^1. \quad (3.19)$$

Moreover, one can show the strict monotonicity

$$a_\varepsilon(\mathbf{u}, \mathbf{u} - \mathbf{v}) - a_\varepsilon(\mathbf{v}, \mathbf{u} - \mathbf{v}) \geq c\|\mathbf{u} - \mathbf{v}\|_1^2 \text{ for all } \mathbf{u}, \mathbf{v} \in \mathbf{H}_0^1. \quad (3.20)$$

Indeed, it holds

$$\begin{aligned} & a_\varepsilon(\mathbf{u}, \mathbf{u} - \mathbf{v}) - a_\varepsilon(\mathbf{v}, \mathbf{u} - \mathbf{v}) \\ &= \int_\Omega 2\mu|D\mathbf{u} - D\mathbf{v}|^2 + \frac{\tau_s}{|D\mathbf{u}|_\varepsilon} D\mathbf{u} : (D\mathbf{u} - D\mathbf{v}) - \frac{\tau_s}{|D\mathbf{v}|_\varepsilon} D\mathbf{v} : (D\mathbf{u} - D\mathbf{v}) \\ &= \int_\Omega 2\mu|D\mathbf{u} - D\mathbf{v}|^2 + \tau_s \left(\frac{D\mathbf{u} - D\mathbf{v}}{|D\mathbf{u}|_\varepsilon} + \left(\frac{1}{|D\mathbf{u}|_\varepsilon} - \frac{1}{|D\mathbf{v}|_\varepsilon} \right) D\mathbf{v} \right) : (D\mathbf{u} - D\mathbf{v}) \\ &= \int_\Omega 2\mu|D\mathbf{u} - D\mathbf{v}|^2 + \frac{\tau_s}{|D\mathbf{u}|_\varepsilon} \left(|D\mathbf{u} - D\mathbf{v}|^2 - \frac{|D\mathbf{u}|_\varepsilon - |D\mathbf{v}|_\varepsilon}{|D\mathbf{v}|_\varepsilon} D\mathbf{v} : (D\mathbf{u} - D\mathbf{v}) \right) \\ &\geq \int_\Omega 2\mu|D\mathbf{u} - D\mathbf{v}|^2 + \frac{\tau_s}{|D\mathbf{u}|_\varepsilon} \left(|D\mathbf{u} - D\mathbf{v}|^2 - \frac{|D\mathbf{u} - D\mathbf{v}|}{|D\mathbf{v}|_\varepsilon} D\mathbf{v} : (D\mathbf{u} - D\mathbf{v}) \right). \end{aligned}$$

Monotonicity (3.20) follows from (3.1) applied to the first term in the last inequality and noting that since $|\frac{D\mathbf{v}}{|D\mathbf{v}|_\varepsilon}| \leq 1$, the second term is non-negative.

Proposition 3.1 *The problem (2.6) has a unique solution $\{\mathbf{u}, p\} \in \mathbf{H}_0^1 \times L_0^2$ satisfying the estimate*

$$\|\nabla \mathbf{u}\| \leq \frac{1}{\mu} \|\mathbf{f}\|_{-1}, \quad \|p\| \leq c(\|\mathbf{f}\|_{-1} + \tau_s \min\{1, \frac{1}{\varepsilon} \|\mathbf{f}\|_{-1}\}). \quad (3.21)$$

Proof First, consider (2.6) restricted to the divergence free subspace \mathbf{V} : find $\mathbf{u} \in \mathbf{V}$ such that

$$a_\varepsilon(\mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \text{ for all } \mathbf{v} \in \mathbf{V}. \quad (3.22)$$

Due to (3.18), (3.19) and (3.20) as well as $\mathbf{V} \subset \mathbf{H}_0^1$ we may apply the Browder-Minty method of strictly monotone operators (Theorems 3.4 and 3.5) to prove the existence and uniqueness of $\mathbf{u} \in \mathbf{V}$ solving (2.6). The equivalence of (2.6) and (3.22) together with the existence and uniqueness of the pressure as a Lagrange multiplier corresponding to the divergence free constraint can be shown by a standard argument,

see [50]. To prove the estimate (3.21) for the velocity we set in (2.6) $\mathbf{u} = \mathbf{v}$ and $q = p$ and apply the inequality $(\mathbf{f}, \mathbf{v}) \leq \|\mathbf{f}\|_{-1} \|\nabla \mathbf{v}\|$ to estimate the right-hand side. The bound for the pressure follows by the Nečas inequality

$$\|p\| \leq c \sup_{\mathbf{v} \in \mathbf{H}_0^1} \frac{(\nabla \cdot \mathbf{v}, p)}{\|\nabla \mathbf{v}\|}. \quad (3.23)$$

Indeed, setting in (2.6) $q = 0$, dividing the equality by $\|\nabla \mathbf{v}\|$ and exploiting $|\frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}| < 1$ we get, together with the Korn and Cauchy inequalities (3.1) and (A.6) that

$$\begin{aligned} \frac{(\nabla \cdot \mathbf{v}, p)}{\|\nabla \mathbf{v}\|} &= \frac{2\mu(D\mathbf{u}, D\mathbf{v}) + \tau_s(|D\mathbf{u}|_\varepsilon^{-1} D\mathbf{u}, D\mathbf{v}) - (\mathbf{f}, \mathbf{v})}{\|\nabla \mathbf{v}\|} \\ &\leq \frac{2\mu\|\nabla \mathbf{u}\|\|\nabla \mathbf{v}\| + \tau_s(1, |D\mathbf{v}|) + \|\mathbf{f}\|_{-1}}{\|\nabla \mathbf{v}\|} \leq 2\mu\|\nabla \mathbf{u}\| + \tau_s\sqrt{|\Omega|} + \|\mathbf{f}\|_{-1}. \end{aligned}$$

Passing to the upper limit with respect to \mathbf{v} and using (3.23) yields

$$\|p\| \leq c(2\mu\|\nabla \mathbf{u}\| + \tau_s\sqrt{|\Omega|} + \|\mathbf{f}\|_{-1}).$$

The estimate

$$\|p\| \leq c\left(2\mu + \frac{\tau_s}{\varepsilon}\right)\|\nabla \mathbf{u}\| + \|\mathbf{f}\|_{-1}$$

is proven by the same arguments using (3.19). Combining both estimates gives the pressure estimate in (3.21). \square

Notice from the second inequality in (3.21) that for $\mathbf{f} \rightarrow \mathbf{0}$ it holds $p \rightarrow 0$. However this is not true for $\varepsilon = 0$ since in the model the kinematic pressure is under-determined in the plug region.

3.4 Well-Posedness of the Mixed Formulation

We consider now the well-posedness of Bingham problem in the mixed formulation.

Theorem 3.6 *The problem (2.7) has a unique solution $\{\mathbf{u}, p, W\}$ from $\mathbf{H}_0^1 \times L_0^2 \times \mathcal{L}_s^2$ such that*

$$\|\mathbf{u}\|_1^2 + \varepsilon \tau_s \|W\|^2 \leq \|\mathbf{f}\|_{-1}, \quad \|p\| \leq c(\|\mathbf{f}\|_{-1} + \tau_s \min\{1, \frac{1}{\varepsilon} \|\mathbf{f}\|_{-1}\}). \quad (3.24)$$

Moreover $W \in \mathcal{L}_s^\infty$ and

$$\|W\|_{L^\infty} \leq 1. \quad (3.25)$$

Proof The proof of the well-posedness is based on showing the equivalence of (2.6) and (2.7) and applying Proposition 3.1. Indeed, assume $\{\mathbf{u}, p\} \in \mathbf{H}_0^1 \times L_0^2$ solves (2.6) and define $W = \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}$. Since $D\mathbf{u} \in \mathcal{L}_s^2$ we conclude that the components of W are measurable functions (as a product of such functions). Moreover, since for any $\mathbf{x} \in \Omega$ it holds $|\frac{D\mathbf{u}(\mathbf{x})}{|D\mathbf{u}(\mathbf{x})|_\varepsilon}| \leq 1$, we have $W \in \mathcal{L}_s^\infty$ and both equalities $W|D\mathbf{u}|_\varepsilon^{-1} = D\mathbf{u}$ and $W = \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}$ hold in \mathcal{L}_s^2 . Therefore, in view of (2.6) and noting that for $W = \frac{D\mathbf{u}}{|D\mathbf{u}|_\varepsilon}$ it holds

$$a(\mathbf{u}, \mathbf{v}) + c(\mathbf{v}, W) = a_\varepsilon(\mathbf{u}, \mathbf{v}),$$

we conclude that $\{\mathbf{u}, p, W\}$ satisfies (2.7) for any $\{\mathbf{v}, q, Z\}$ from $\mathbf{H}_0^1 \times L_0^2 \times \mathcal{L}_s^\infty$. Thus the existence of a solution to (2.7) follows by Proposition 3.1. Now assume that some $\{\mathbf{u}, p, W\}$ solves (2.7). Setting $\mathbf{v} = 0$, $q = 0$ and varying $Z \in \mathcal{L}_s^\infty$ we conclude that $|D\mathbf{u}|_\varepsilon W = D\mathbf{u}$ holds in $(\mathcal{L}_s^\infty)' \equiv \mathcal{L}_s^1$. Hence for any solution of (2.7) it holds

$$W = |D\mathbf{u}|_\varepsilon^{-1} D\mathbf{u} \text{ almost everywhere.} \quad (3.26)$$

Using this in the third term of (2.7) and setting $Z = 0$ we conclude that \mathbf{u} and p satisfy (2.6). The uniqueness of the solution of (2.7) follows from Proposition 3.1 and (3.26). Finally, the first two estimates in (3.24) follow by the same arguments as in (3.21), that is, by taking $\mathbf{v} = \mathbf{u}$, $q = p$, $Z = W$, and $W \in \mathcal{L}_s^\infty$ with (3.25) follows from (3.26). \square

Remark Although the formulations (2.6) and (2.7) are equivalent, their finite element discretizations are in general not equivalent and feature different numerical properties.

The extension of Theorem 3.6 and Proposition 3.1 to the limit case $\varepsilon = 0$ is still open. Some well-posedness results for the (non-regularized) Bingham problem are based on the reformulation of the problem as a variational inequality, see [39]. In [10] a non-regularized nonhomogeneous problem is analyzed as a limit case of a regularized problem with an existence result for two dimensional periodic problems. Somewhat related results can be found in [18], where the unsteady Navier-Stokes equations are considered for fluids featuring a stress tensor of the form

$$\boldsymbol{\tau} = \mu(\delta + |D\mathbf{u}|)^{s-2} D\mathbf{u}, \quad s \in (1, \infty).$$

The extension of this result to the case $s = 1$ (and thus to the Bingham problem) remains open.

Remark Once we proved that the solution W of (2.7) is from \mathcal{L}_s^∞ we can extend the space of admissible test functions Z in (2.7) from \mathcal{L}_s^∞ to \mathcal{L}_s^2 .

Chapter 4

Linearization and Discretization ¹

In this chapter we discuss the linearization and discretization of the Bingham fluid flow problem in primitive and mixed variables. We start with the Picard linearization. This scheme will be important for the next chapters and will be addressed in detail. In particular, we establish the well-posedness of a Picard iteration applied to the steady Stokes type problem and provide error estimates in the primitive and mixed formulation. We then present the Newton linearization method and one of its variants, the Newton Continuation. The discretization is then discussed and well-posedness of the discrete problem is proven. We conclude this chapter with a result on the inf-sup stability of the regularized discrete problem.

4.1 The Picard Linearization Method

We will focus on a Picard type linearization of (2.4) in this thesis. For the formulation in primitive variables, starting with some initial guess \mathbf{u}_0 , (2.1) can be linearized by

¹Results in this chapter have been published in A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: Analysis and numerical solution*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2434-2446

the iterations

$$\begin{cases} \rho \left[\frac{\partial \mathbf{u}^k}{\partial t} + (\mathbf{u}^{k-1} \cdot \nabla) \mathbf{u}^k \right] - \nabla \cdot (2\mu D\mathbf{u}^k + \tau_s \frac{D\mathbf{u}^k}{|D\mathbf{u}^{k-1}|_\varepsilon}) + \nabla p^k = \mathbf{f} \\ \nabla \cdot \mathbf{u}^k = 0 \end{cases} \quad (4.1)$$

for $k = 1, 2, 3, \dots$ until some convergence criterion is satisfied. For the mixed formulation, starting with some initial guess $\mathbf{u}^{(0)}$, we solve

$$\begin{cases} \rho \left[\frac{\partial \mathbf{u}^{(k)}}{\partial t} + (\mathbf{u}^{(k-1)} \cdot \nabla) \mathbf{u}^{(k)} \right] - \nabla \cdot (2\mu D\mathbf{u}^{(k)} + \tau_s W^{(k)}) + \nabla p^{(k)} = \mathbf{f} \\ \nabla \cdot \mathbf{u}^{(k)} = 0 \\ D\mathbf{u}^{(k)} - |D\mathbf{u}^{(k-1)}|_\varepsilon W^{(k)} = \mathbf{0} \end{cases} \quad (4.2)$$

for $\mathbf{u}^{(k)}$, $p^{(k)}$ and $W^{(k)}$. Other linearization techniques exist, such as a Newton type linearization or a combination of Newton and Picard linearization, as described in [6].

4.1.1 Well-Posedness of the Picard Iterative Method

We first introduce two auxiliary linear problems. Let $\boldsymbol{\beta} \in \mathbf{H}_0^1$ be a given vector field. For the first problem we want to find \mathbf{u} and p such that

$$\begin{cases} -\nabla \cdot \left(\left(2\mu + \frac{\tau_s}{|D\boldsymbol{\beta}|_\varepsilon} \right) D\mathbf{u} \right) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad \text{in } \Omega \quad (4.3)$$

with $\mathbf{u} = \mathbf{0}$ on $\partial\Omega$ for simplicity. We define the bilinear form

$$a_\beta(\mathbf{u}, \mathbf{v}) = \int_\Omega \left(2\mu + \frac{\tau_s}{|D\boldsymbol{\beta}|_\varepsilon} \right) D\mathbf{u} : D\mathbf{v}$$

on $\mathbf{V} \times \mathbf{V}$. The weak formulation of (4.3) reads: find $(\mathbf{u}, p) \in \mathbf{H}_0^1 \times L_0^2$ such that

$$a_\beta(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) - b(q, \mathbf{u}) = (\mathbf{f}, \mathbf{v}) \quad \text{for all } (\mathbf{v}, q) \in \mathbf{H}_0^1 \times L_0^2. \quad (4.4)$$

Proposition 4.1 *Given $\mathbf{f} \in \mathbf{L}^2$ and $\boldsymbol{\beta} \in \mathbf{H}_0^1$ there exists a unique solution (\mathbf{u}, p) to (4.4).*

Proof By Korn and Friedrich's inequality, the bilinear form a_β is coercive:

$$a_\beta(\mathbf{u}, \mathbf{u}) \geq \mu \|\nabla \mathbf{u}\|^2 \geq c \|\mathbf{u}\|_1^2.$$

It is also straightforward to check that $a_\beta(\cdot, \cdot)$ is continuous on $\mathbf{V} \times \mathbf{V}$. Since \mathbf{V} can be equivalently defined as $\mathbf{V} = \{\mathbf{v} \in \mathbf{H}_0^1 : b(q, \mathbf{v}) = 0 \text{ for all } q \in L_0^2\}$, the result follows by Corollary 5.1 from [50]. \square

For the second auxiliary problem we want to find \mathbf{u} , p and W such that

$$\begin{cases} -\nabla \cdot (2\mu D\mathbf{u} + \tau_s W) + \nabla p = \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \text{ in } \Omega. \\ |D\boldsymbol{\beta}|_\varepsilon W - D\mathbf{u} = 0 \end{cases} \quad (4.5)$$

Let

$$g_{\varepsilon, \boldsymbol{\beta}}(W, Z) = \int_\Omega \tau_s |D\boldsymbol{\beta}|_\varepsilon W : Z.$$

The weak form of (4.5) reads: find $(\mathbf{u}, p, W) \in \mathbf{H}_0^1 \times L_0^2 \times \mathcal{L}_s^2$ such that

$$a(\mathbf{u}, \mathbf{v}) + b(p, \mathbf{v}) + c(\mathbf{v}, W) - b(q, \mathbf{u}) - c(\mathbf{u}, Z) + g_{\varepsilon, \boldsymbol{\beta}}(W, Z) = (\mathbf{f}, \mathbf{v}) \quad (4.6)$$

for any $(\mathbf{u}, p, W) \in \mathbf{H}_0^1 \times L_0^2 \times \mathcal{L}_s^\infty$.

Proposition 4.2 *Given $\mathbf{f} \in \mathbf{L}^2$, there exists a unique solution (\mathbf{u}, p, W) to (4.6).*

Proof The proof is carried out by showing the equivalence of (4.4) and (4.6) and then applying Proposition 4.1. The arguments are almost the same as in the nonlinear case, see Theorem 3.6. The solution (\mathbf{u}, p) of (4.4) together with $W = \frac{D\mathbf{u}}{|D\boldsymbol{\beta}|_\varepsilon} \in \mathcal{L}_s^\infty$ solves (4.6). Conversely, if (\mathbf{u}, p, W) solves (4.4), then setting $\mathbf{v} = \mathbf{0}$, $q = 0$ and varying

$Z \in \mathcal{L}_s^\infty$ it holds $\frac{W}{|D\beta|_\varepsilon} = \mathbf{u}$ in $(\mathcal{L}_s^\infty)' \equiv \mathcal{L}_s^1$. Thus $W = \frac{D\mathbf{u}}{|D\beta|_\varepsilon}$ holds almost everywhere in Ω . Inserting this in the third term of (4.6) and letting $Z = 0$ we find that (\mathbf{u}, p) solves (4.4). \square

Note that the Picard linearized versions of the Bingham problem in the primitive and mixed formulation (4.1) and (4.2) are of the form (4.3) and (4.5) with $\beta = \mathbf{u}^{k-1}$ and should be considered in the weak sense (4.4) and (4.6). Propositions 4.1 and 4.2 imply the well-posedness of a single Picard iteration.

Remark The results in this section can be easily extended to the unsteady Navier-Stokes type equations, discretized in time by a classical backward Euler scheme. In this case, the auxiliary problem reads

$$\begin{cases} \rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\beta \cdot \nabla) \mathbf{u} \right) - \nabla \cdot \left(\left(2\mu + \frac{\tau_s}{|D\beta|_\varepsilon} \right) D\mathbf{u} \right) + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 \end{cases} \quad (4.7)$$

$\mathbf{u} = \mathbf{0}$ on $\partial\Omega$. We then define two additional forms:

$$q(\mathbf{u}, \mathbf{v}) = \frac{\rho}{\Delta t} \int_{\Omega} \mathbf{u} \cdot \mathbf{v} \text{ on } \mathbf{H}_0^1 \times \mathbf{H}_0^1, \quad c_\beta(\mathbf{u}, \mathbf{v}) = \rho \int_{\Omega} (\beta \cdot \nabla) \mathbf{u} \cdot \mathbf{v} \text{ on } \mathbf{H}_0^1 \times \mathbf{H}_0^1,$$

where Δt is the time discretization parameter. The weak formulation, after time discretization, then reads: find $(\mathbf{u}, p) \in \mathbf{H}_0^1 \times L_0^2$ such that

$$\begin{aligned} q(\mathbf{u}^{k+1}, \mathbf{v}) + c_\beta(\mathbf{u}^{k+1}, \mathbf{v}) + a_\beta(\mathbf{u}, \mathbf{v}) + b(p^{k+1}, \mathbf{v}) - b(q, \mathbf{u}^{k+1}) \\ = (\mathbf{f}, \mathbf{v}) + q(\mathbf{u}^k, \mathbf{v}) \text{ for all } (\mathbf{v}, q) \in \mathbf{H}_0^1 \times L_0^2, \end{aligned}$$

where the superscript k , $k = 0, 1, 2, \dots$ denotes the time discretization step. The bilinear form $q(\cdot, \cdot)$ is clearly well-defined, continuous and coercive. The boundedness and continuity of the form $c_\beta(\cdot, \cdot)$ is a consequence of the Hölder's inequality (A.5) together with the Sobolev embedding theorem $H^1(\Omega) \hookrightarrow L^4(\Omega)$, see [44]. Then the

same stream of arguments as used for (4.3) can be applied to (4.7) for showing the well-posedness of (4.7).

4.1.2 Error Estimates for the Picard Iterative Scheme

Denote by (\mathbf{u}, p, W) the solution of (2.7). Then (\mathbf{u}, p) also solve (2.6). Let $\mathbf{e}^{(k)} \equiv \mathbf{u}^{(k)} - \mathbf{u}$, $e^{(k)} = p^{(k)} - p$ and $E^{(k)} = W^{(k)} - W$. We first consider the iterations for the problem in primitive variables (4.1). Equations (2.6) and (4.1) yield the error equation

$$a(\mathbf{e}^{(k)}, \mathbf{v}) + \int_{\Omega} \tau_s \left(\frac{D\mathbf{u}^{(k)}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} - \frac{D\mathbf{u}}{|D\mathbf{u}|_{\varepsilon}} \right) : D\mathbf{v} + b(\mathbf{v}, e^{(k)}) - b(\mathbf{e}^{(k)}, q) = 0 \quad (4.8)$$

for all $\mathbf{v} \in \mathbf{V}$ and $q \in L_0^2$.

Proposition 4.3 *The velocity error of the iterations (4.1) satisfies*

$$\|\mathbf{e}^{(k)}\|_1 \leq \frac{C}{\varepsilon} \|\mathbf{e}^{(k)}\|_1 + O(\|\mathbf{e}^{(k)}\|_1^2). \quad (4.9)$$

Proof We rewrite the second term in (4.9) as

$$\begin{aligned} & \int_{\Omega} \tau_s \left(\frac{D\mathbf{u}^{(k)}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} - \frac{D\mathbf{u}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} + \frac{D\mathbf{u}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} - \frac{D\mathbf{u}}{|D\mathbf{u}|_{\varepsilon}} \right) : D\mathbf{v} \\ &= \int_{\Omega} \tau_s \frac{D\mathbf{e}^{(k)} : D\mathbf{v}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} + \int_{\Omega} \tau_s \left(\frac{1}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} - \frac{1}{|D\mathbf{u}|_{\varepsilon}} \right) D\mathbf{u} : D\mathbf{v}. \end{aligned}$$

Upon Frechét linearization, which is always possible as long as $\varepsilon > 0$, we have that

$$\frac{1}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} - \frac{1}{|D\mathbf{u}|_{\varepsilon}} = \frac{D\mathbf{u} : D\mathbf{e}^{(k-1)}}{|D\mathbf{u}|_{\varepsilon}^3} + h.o.t.$$

Selecting $\mathbf{v} = \mathbf{e}^{(k)}$ and $q = \mathbf{e}^{(k)}$ in equation (4.8) gives

$$a(\mathbf{e}^{(k)}, \mathbf{e}^{(k)}) + \int_{\Omega} \tau_s \frac{D\mathbf{e}^{(k)} : D\mathbf{e}^{(k)}}{|D\mathbf{u}^{(k-1)}|_{\varepsilon}} = - \int_{\Omega} \tau_s \left(\frac{D\mathbf{u} : D\mathbf{e}^{(k-1)}}{|D\mathbf{u}|_{\varepsilon}^3} + h.o.t. \right) D\mathbf{u} : D\mathbf{e}^{(k)}.$$

With the coercivity of the bilinear form $a(\cdot, \cdot)$, the inequality (3.1), the ε -independent bound (3.23) on \mathbf{u} and noting that $\frac{x^2}{\sqrt{(x^2+\varepsilon^2)^3}} \leq \frac{2}{\sqrt{27\varepsilon}}$ for any $x \in \mathbb{R}$, we obtain the inequality

$$\mu \|\nabla \mathbf{e}^{(k)}\|^2 \leq \frac{C}{\varepsilon} (\|\mathbf{e}^{(k-1)}\|_1 + O(\|\mathbf{e}^{(k-1)}\|_1^2)) \|\mathbf{e}^{(k)}\|_1$$

where C here and after is a constant independent of ε . The latter inequality implies (4.9). \square

This Proposition quantifies the impact of small values of the regularization parameter ε on the Picard iteration in the formulation in primitive variables: in general, small values of ε slow down the convergence, as mentioned earlier and as it is pointed out in [51, 57]. We can prove a similar result for the mixed formulation:

Proposition 4.4 *The error of the iterative scheme (4.2) satisfies*

$$\|\mathbf{e}^{(k)}\|_1 + \sqrt{\varepsilon} \|E^{(k)}\| \leq \frac{C}{\sqrt{\varepsilon}} \|\mathbf{e}^{(k-1)}\|_1. \quad (4.10)$$

Proof After a memberwise subtraction of (4.2) and (2.7) and standard manipulations with $\mathbf{v} = \mathbf{e}^{(k)}$, $q = e^{(k)}$ and $Z = E^{(k)}$ (test functions can be taken from \mathcal{L}_s^2 , see Remark 3.4), we get

$$a(\mathbf{e}^{(k)}, \mathbf{e}^{(k)}) + g_\varepsilon(|D\mathbf{u}^{(k-1)}|_\varepsilon - |D\mathbf{u}|_\varepsilon, W, E^{(k)}) + g_\varepsilon(|D\mathbf{u}^{(k-1)}|_\varepsilon, E^{(k)}, E^{(k)}) = 0. \quad (4.11)$$

For the mapping $f : \mathbf{H} \rightarrow L^2$, $f(\mathbf{v}) = |D\mathbf{v}|_\varepsilon$, we find the Frechet derivative operator

$$d(f)|_{\mathbf{a}} = \frac{D\mathbf{a}}{|D\mathbf{a}|_\varepsilon} : D \implies \|d(f)\|_{\mathbf{H}_1 \rightarrow L^2} \leq 1 \text{ for all } \mathbf{a} \in \mathbf{H}_0^1.$$

Therefore

$$\| |D\mathbf{u}^{(k-1)}|_\varepsilon - |D\mathbf{u}|_\varepsilon \| \leq \|D\mathbf{e}^{(k-1)}\|. \quad (4.12)$$

Recalling that $\|W\|_{\mathcal{L}^\infty} \leq 1$ (cf. Theorem 3.6) and $g_\varepsilon(|D\mathbf{u}^{(k-1)}|_\varepsilon, E^{(k)}, E^{(k)}) \geq \varepsilon \|E^{(k)}\|^2$,

we obtain from (4.11) and (4.12) the inequality

$$\mu \|\nabla \mathbf{e}^{(k)}\|_1^2 + \varepsilon \|E^{(k)}\|^2 \leq \tau_s \|\mathbf{e}^{(k-1)}\|_1 \|E^{(k)}\| \leq \frac{\tau_s}{2\varepsilon} \|\mathbf{e}^{(k-1)}\|_1^2 + \frac{\varepsilon}{2} \|E^{(k)}\|^2.$$

Thus

$$\|\mathbf{e}^{(k)}\|_1^2 + \varepsilon \|E^{(k)}\|^2 \leq C \frac{\tau_s}{\varepsilon} \|\mathbf{e}^{(k-1)}\|_1^2.$$

□

Remark Notice that (4.10) enjoys a milder dependence on ε than (4.9); also the higher order terms disappear. At the same time, the velocity and pressure iterations from (4.1) and (4.2) are the same due to the equivalence of the auxiliary linear systems. Hence the velocity error $\mathbf{e}^{(k-1)}$ from (4.1) should also satisfy the improved bound (4.10). However, the argument to show this improved bound is indirect and resorts to the mixed formulation. Moreover, such an indirect argument may not be valid in the discrete case since the equivalence does not hold any longer in general. Comparing the numerical performance of both formulations shows however that iterative methods for the mixed formulation are less sensitive to small values of ε (see Chapter 5).

4.2 The Newton Method

For the primitive variables formulation (2.2) one step of the Newton method can be written as follows: given $\mathbf{u}^{(k-1)}$, find $\mathbf{u}^{(k)}$ and $p^{(k)}$ satisfying

$$\left\{ \begin{array}{l} -\nabla \cdot \left(2\mu + \frac{\tau_s}{|D\mathbf{u}^{(k-1)}|_\varepsilon} \left[1 - \frac{D\mathbf{u}^{(k-1)} : D\mathbf{u}^{(k-1)}}{|D\mathbf{u}^{(k-1)}|_\varepsilon^2} \right] \right) D\mathbf{u}^{(k)} + \nabla p^{(k)} \\ \qquad \qquad \qquad = \mathbf{f} - \tau_s \nabla \cdot \frac{D\mathbf{u}^{(k-1)} |D\mathbf{u}^{(k-1)}|^2}{|D\mathbf{u}^{(k-1)}|_\varepsilon^3} \\ \qquad \qquad \qquad \nabla \cdot \mathbf{u}^{(k)} = 0. \end{array} \right.$$

For the mixed formulation, one step of the Newton method reads: given $\mathbf{u}^{(k-1)}$ and $W^{(k-1)}$, find $\mathbf{u}^{(k)}$, $p^{(k)}$ and $W^{(k)}$ satisfying

$$\begin{cases} -\nabla \cdot (2\mu D\mathbf{u}^{(k)} + \tau_s W^{(k)}) + \nabla p^{(k)} = \mathbf{f} \\ \nabla \cdot \mathbf{u}^{(k)} = 0 \\ D\mathbf{u}^{(k)} - |D\mathbf{u}^{(k-1)}|_\varepsilon W^{(k)} - \frac{D\mathbf{u}^{(k-1)}:D\mathbf{u}^{(k-1)}}{|D\mathbf{u}^{(k-1)}|_\varepsilon} W^{(k-1)} = -\frac{|D\mathbf{u}^{(k-1)}|^2}{|D\mathbf{u}^{(k-1)}|_\varepsilon} W^{(k-1)}. \end{cases}$$

In Section 2.1 we mentioned the lack of robustness of the Newton method when ε gets smaller. In particular, the domain of convergence shrinks as ε decreases, making a good initial guess necessary to ensure convergence. This problem can be remedied in different ways. One possibility is to use a mix of Picard and Newton iterations: We may perform a few iterations of the more robust Picard method and start the Newton iterations once a good initial guess has been reached. Another possibility is to use a so-called *continuation strategy* within the Newton method.

4.2.1 Newton Continuation

The idea behind the continuation method is to have a non-stationary selection of ε , such that in the k -th iterate we have $\varepsilon = \varepsilon(k)$ and $\lim_{k \rightarrow \infty} \varepsilon(k) = 0$. We start with a (large value of) ε and perform one Newton iteration. If this iteration reduces the residual by more than ε (in the algorithm below denoted by “success == true”), the new solution is accepted and the value of ε is reduced. If not (“success == false”), the new solution is discarded and ε is increased. The procedure continues until a specified target value of ε is reached. The resulting algorithm reads:

Algorithm 4.1 Newton Continuation

```

 $\varepsilon_{new} = \varepsilon_0, \varepsilon_{curr} = \varepsilon_{new}, \rho = \rho_0;$ 

 $\mathbf{u} = \mathbf{u}_0, p = p_0, W = 0;$ 

while  $\varepsilon_{curr} > \varepsilon_{target}$  do
   $[\mathbf{u}_{new}, p_{new}, W_{new}, \text{success}] = \text{Newton}(\mathbf{u}, p, W, \mathbf{f}, \varepsilon_{curr});$ 
  if success == true then
     $\varepsilon_{new} = \varepsilon_{curr}, \varepsilon_{curr} = \rho\varepsilon_{curr}, \rho = 0.9\rho;$ 
     $\mathbf{u} = \mathbf{u}_{new}, p = p_{new}, W = W_{new};$ 
  else
     $\rho = \frac{1+\rho}{2};$ 
     $\varepsilon_{curr} = \rho\varepsilon_{new};$ 
  end if
end while

```

4.3 Discretization

There are different ways to discretize (4.1) in space, examples are the MAC discretization on staggered grids and collocated finite difference methods [1, 66], finite volume [84] or LBB-stable finite elements [34]. This thesis will focus on Galerkin finite element discretization methods. The application of the approach presented here to other discretization methods is however possible. Denote by $\mathbf{H}_h \subset \mathbf{H}_0^1$, $Q_h \subset L_0^2$ and $\mathcal{W}_h \subset \mathcal{L}_s^2$ the finite dimensional subspaces for the velocity, pressure, and the auxiliary variable W , respectively. We assume throughout this thesis that the pair of spaces \mathbf{H}_h and Q_h is LBB stable [50]. The finite element method for (2.7) reads: Find $\mathbf{u}_h \in \mathbf{H}_h$, $p_h \in Q_h$ and $W_h \in \mathcal{W}_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) + b(p_h, \mathbf{v}_h) + c(\mathbf{v}_h, W_h) - b(q_h, \mathbf{u}_h) - c(\mathbf{u}_h, Z_h) + g_\varepsilon(\mathbf{u}_h, W_h, Z_h) = (\mathbf{f}, \mathbf{v}_h) \quad (4.13)$$

for any $\mathbf{v}_h \in \mathbf{H}_h$, $q_h \in Q_h$ and $Z_h \in \mathcal{W}_h$.

4.3.1 Well-Posedness of the Discrete Mixed Problem

We now turn our attention to the well-posedness and stability of the discrete Bingham problem (4.13). Unlike the continuous case (see Theorem 3.6), the discrete problem (4.13) is in general not equivalent to the finite element counterpart of the original problem (2.6). Thus the well-posedness for (4.13) does not follow directly from the theory of monotone operators applied to the weak formulation of (2.6). Instead, the proof of the well-posedness of the discrete problem relies on the Schaefer's extension of the Brouwer theorem (see [42], Section 9.2):

Theorem 4.1 *Suppose*

$$F : X \rightarrow X$$

is a continuous and compact mapping. Assume further that the set

$$\{u \in X | u = \lambda F[u] \text{ for some } \lambda \text{ with } 0 \leq \lambda \leq 1\}$$

is bounded. Then F has a fixed point.

The next theorem is the well-posedness of the discrete problem.

Theorem 4.2 *The problem (4.13) has a solution (\mathbf{u}, p, W) from $\mathbf{H}_h \times W_h \times Q_h$ such that*

$$\|\mathbf{u}_h\|_1^2 + \varepsilon \tau_s \|W_h\|_0^2 \leq \|\mathbf{f}\|_{-1}, \quad \|p_h\| \leq c(1 + \tau_s \varepsilon^{-1}) \|\mathbf{f}\|_{-1} \quad (4.14)$$

for a constant $c > 0$. If \mathbf{f} is sufficiently small or μ and ε are sufficiently large then the solution is unique.

Proof Define the discrete divergence free space $\mathbf{V}_h = \{\mathbf{v}_h \in \mathbf{H}_h : b(q_h, \mathbf{v}_h) = 0 \text{ for all } q_h \in Q_h\}$. For arbitrary $\lambda \in [0, 1]$ consider the problem: find $\mathbf{u}_h^\lambda \in \mathbf{V}_h$,

$W_h^\lambda \in \mathcal{W}_h$ such that

$$\begin{aligned} a(\mathbf{u}_h^\lambda, \mathbf{v}_h) + \lambda c(\mathbf{v}_h, W_h^\lambda) &= \lambda(\mathbf{f}, \mathbf{v}_h), \\ g_\varepsilon(\mathbf{u}_h^\lambda, W_h^\lambda, Z_h) - \lambda c(\mathbf{u}_h^\lambda, Z_h) &= 0 \end{aligned} \quad (4.15)$$

for any $\mathbf{v}_h \in V_h$ and $Z_h \in \mathcal{W}_h$. For $\lambda = 1$ the problem (4.15) is equivalent to (4.13).

To apply Theorem 4.1, it suffices to show:

- (i) the set of solutions to (4.15) is bounded uniformly with respect to λ and
- (ii) the mapping $\{\mathbf{u}_h^{old}, W_h^{old}\} \rightarrow \{\mathbf{u}_h^{new}, W_h^{new}\}$ defined by

$$\begin{aligned} a(\mathbf{u}_h^{new}, \mathbf{v}_h) &= (\mathbf{f}, \mathbf{v}_h) - c(\mathbf{v}_h, W_h^{old}) \text{ for all } \mathbf{v}_h \in \mathbf{V}_h, \\ g_\varepsilon(\mathbf{u}_h^{old}, W_h^{new}, Z_h) &= c(\mathbf{u}_h^{old}, Z_h) \text{ for all } Z_h \in \mathcal{W}_h \end{aligned} \quad (4.16)$$

is continuous and bounded (all spaces are finite dimensional, so the boundedness implies compactness).

To find a bound for $\{\mathbf{u}_h^\lambda, W_h^\lambda\}$ we set in (4.15) $\mathbf{v}_h = \mathbf{u}_h^\lambda$, $Z_h = W_h^\lambda$. Summing up the equalities gives

$$\begin{aligned} &\min\{2\mu, \tau_s, \varepsilon\} (\|\nabla \mathbf{u}_h^\lambda\|^2 + \|W_h^\lambda\|^2) \\ &\leq a(\mathbf{u}_h^\lambda, \mathbf{u}_h^\lambda) + g_\varepsilon(\mathbf{u}_h^\lambda, W_h^\lambda, W_h^\lambda) = \lambda(\mathbf{f}, \mathbf{u}_h^\lambda) \\ &\leq \frac{1}{2} \min\{2\mu, \tau_s, \varepsilon\} \|\nabla \mathbf{u}_h^\lambda\|^2 + \max\{\mu^{-1}, \tau_s^{-1} \varepsilon^{-1}\} \|\mathbf{f}\|_{-1}^2. \end{aligned}$$

Thus

$$\|\nabla \mathbf{u}_h^\lambda\|^2 + \|W_h^\lambda\|^2 \leq \max\{\mu^{-2}, \tau_s^{-2} \varepsilon^{-2}\} \|\mathbf{f}\|_{-1}^2 \text{ for all } \lambda \in [0, 1].$$

Now we check that the mapping defined by (4.16) is bounded and continuous. To see the boundedness we set in (4.16) $\mathbf{v}_h = \mathbf{u}_h^{new}$ and $Z_h = W_h^{new}$ and get through the

Cauchy and Friedrich's inequalities

$$\|\nabla \mathbf{u}_h^{new}\| \leq c(\tau_s \|W_h^{old}\| + \|\mathbf{f}\|_{-1}) \text{ and } \varepsilon \|W_h^{new}\| \leq \|\nabla \mathbf{u}_u^{old}\|.$$

The continuity follows from the observation that the forms in (4.16) are continuous with respect to every argument. Therefore Theorem 4.1 provides the existence of a solution to (4.15) for $\lambda = 1$. Let \mathbf{u}_1 , W_1 and \mathbf{u}_2 , W_2 be two solutions to (4.15) with $\lambda = 1$. Denote $\mathbf{e}_h = \mathbf{u}_1 - \mathbf{u}_2$ and $E_h = W_1 - W_2$. Setting $\mathbf{v}_h = \mathbf{e}_h$ and $Z_h = E$ in (4.15) yields

$$2\mu \|D\mathbf{e}_h\|^2 + g_\varepsilon(\mathbf{u}_1, E_h, E_h) + (g_\varepsilon(\mathbf{u}_1, W_2, E_h) - g_\varepsilon(\mathbf{u}_2, W_2, E_h)) = 0. \quad (4.17)$$

Therefore (4.17) and (4.12) result in

$$2\mu \|D\mathbf{e}_h\|^2 + \varepsilon \|E_h\|^2 - \|W_2\|_{L^\infty} \|D\mathbf{e}_h\| \|E_h\| \leq 0.$$

Now the a priori bound (4.14) and the smallness assumption yield the uniqueness result. The standard argument in [50] shows the existence and uniqueness of the pressure p_h as a Lagrange multiplier. \square

Even though the equivalence between the formulation in primitive and mixed variables does not hold in general on the discrete level, there is a particular selection of finite elements for which the equivalence indeed holds: P1isoP2 for velocity, P1 for pressure and P0isoP2 for W . Here the \mathbf{V}_h consists of continuous piecewise linear functions with respect to the triangulation built by connecting the middle points of the edges of the original triangulation. Similarly, \mathcal{W}_h consists of piecewise constant functions with respect to the same refined triangulation (by analogy denoted by P0isoP2). See Figure 4.1 for the degrees of freedom of \mathbf{u} , p and W in these types of finite elements.

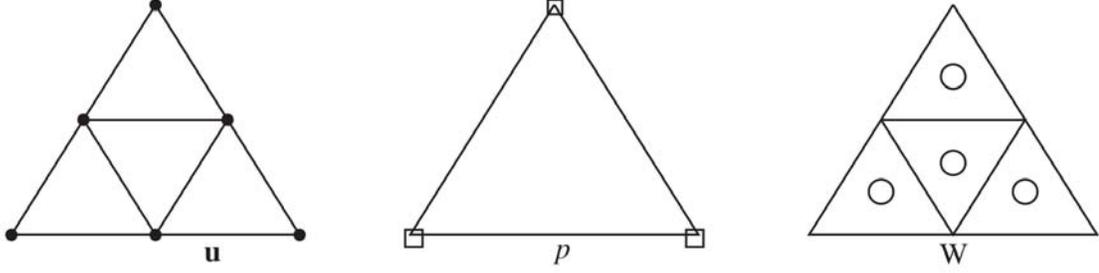


Figure 4.1: Degrees of freedom for the P1isoP2-P1-P0isoP2. \bullet are the degrees of freedom for the velocity components, \square for the pressure and \circ for the tensor W . Images by Maxim A. Olshanskii, Department of Mechanics and Mathematics, Moscow M. V. Lomonosov State University.

4.3.2 Algebraic Properties

We now establish the stability of the linear systems arising from the discretization and linearization of the problem. Denote by $N_{\mathbf{u}}$, N_p and N_W the number of degrees of freedom of each velocity component in \mathbf{H}_h , the pressure in Q_h and each component of the symmetric tensor in \mathcal{W}_h , respectively. Each iteration of the Picard method for the discrete mixed formulation requires solving a linear system of the form

$$\mathcal{A} \begin{bmatrix} \mathbf{u} \\ p \\ W \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \\ \mathbf{0} \end{bmatrix}, \quad (4.18)$$

with

$$\mathcal{A} = \begin{bmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & -N \end{bmatrix}. \quad (4.19)$$

If d is the space dimension, $d \in \{2, 3\}$, we have $A \in \mathbb{R}^{dN_{\mathbf{u}} \times dN_{\mathbf{u}}}$, $B \in \mathbb{R}^{N_p \times dN_{\mathbf{u}}}$. For the size of C and N , we may exploit the symmetry of the tensor W . Note that $W \in \mathbb{R}^d$ and $W = W^T$ so that entries on the strictly lower part of the diagonal can be eliminated. Thus, the number of additional unknowns added to the system is $d^2 - \frac{1}{2}(d^2 - d) =$

$\frac{d(d+1)}{2}$. It follows that $C \in \mathbb{R}^{\frac{d(d+1)}{2}N_W \times dN_u}$, $N \in \mathbb{R}^{\frac{d(d+1)}{2}N_W \times \frac{d(d+1)}{2}N_W}$. The matrix N denotes the discrete counterpart of the term $|D\mathbf{u}|_\varepsilon W$, which is a weighted mass matrix. In computations it can often be replaced by a lumped (diagonal) matrix. When using Q1 or P1 finite elements for discretizing W (as we do in our experiments, see Chapters 5 and 7), this lumping can be obtained by using a trapezoidal rule for the numerical integration when assembling the matrix N .

Proposition 4.5 *For $\varepsilon > 0$, the linear system (4.18), (4.19) is non-singular for any choice of the finite element subspace \mathcal{W}_h .*

Proof For convenience, let us reorder the rows and columns of \mathcal{A} such that

$$\tilde{\mathcal{A}} = \begin{bmatrix} -N & C & 0 \\ C^T & A & B^T \\ 0 & B & 0 \end{bmatrix}.$$

This reordering will not effect the rank of \mathcal{A} . Let

$$\mathcal{D} = \begin{bmatrix} -N & C \\ C^T & A \end{bmatrix} \text{ and } \mathcal{B} = \begin{bmatrix} 0 & B \end{bmatrix}, \text{ then } \tilde{\mathcal{A}} = \begin{bmatrix} \mathcal{D} & \mathcal{B}^T \\ \mathcal{B} & 0 \end{bmatrix}.$$

First we prove that \mathcal{D} is non-singular. Consider the matrix factorization

$$\mathcal{D} = \begin{bmatrix} I & 0 \\ -C^T N^{-1} & I \end{bmatrix} \begin{bmatrix} -N & 0 \\ 0 & \Sigma_W \end{bmatrix} \begin{bmatrix} I & -N^{-1}C \\ 0 & I \end{bmatrix}, \quad (4.20)$$

with the Schur complement matrix $\Sigma_W = A + C^T N^{-1}C$. It is clear from (4.20) that \mathcal{D} is non-singular if and only if N and Σ_W are non-singular. Since $\varepsilon > 0$, N is symmetric and positive definite. The non-singularity of Σ_W follows by the observation

that also A is symmetric and positive definite. By the same argument, since

$$\hat{\mathcal{A}} = \begin{bmatrix} I & 0 \\ \mathcal{B}\mathcal{D}^{-1} & I \end{bmatrix} \begin{bmatrix} \mathcal{D} & 0 \\ 0 & -\mathcal{B}\mathcal{D}^{-1}\mathcal{B}^T \end{bmatrix} \begin{bmatrix} I & \mathcal{D}^{-1}\mathcal{B}^T \\ 0 & I \end{bmatrix},$$

we have that $\tilde{\mathcal{A}}$ (and hence \mathcal{A}) is non-singular if \mathcal{D} and the Schur complement $\mathcal{B}\mathcal{D}^{-1}\mathcal{B}^T$ are non-singular. Noticing from (4.20) that

$$\mathcal{D}^{-1} = \begin{bmatrix} I & N^{-1}C \\ 0 & I \end{bmatrix} \begin{bmatrix} -N^{-1} & 0 \\ 0 & \Sigma_W^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ C^T N^{-1} & I \end{bmatrix},$$

we obtain through a direct computation

$$-\mathcal{B}\mathcal{D}^{-1}\mathcal{B}^T = -B\Sigma_W^{-1}B^T,$$

which is a symmetric and positive definite matrix, so the Proposition is proven. \square

Note that if we impose $\mathbf{u} = 0$ on the entire boundary $\partial\Omega$, then the block B is one-rank deficient having the hydrostatic pressure mode in its kernel. In computations, we may overcome this by modifying \mathcal{A} such that

$$\mathcal{A} = \begin{bmatrix} A & B^T & C^T \\ B & -\alpha Q & 0 \\ C & 0 & -N \end{bmatrix}. \quad (4.21)$$

The matrix Q serves as a pressure stabilization term which allows for a non inf-sup stable space discretization. In addition, it also disambiguates the pressure in purely Dirichlet boundary condition problems. In our experiments we choose Q to be a mass matrix in the discrete pressure space and take the parameter α as small as 10^{-10} .

Remark The result in the previous proposition does not hold for $\varepsilon = 0$ and the

non-singularity of the linear system can not be guaranteed. A necessary condition in this case is that $\ker(C) \cap \ker(N) = \emptyset$. The problem of finding finite dimensional spaces forcing the well-posedness of the discrete problem in this case remains open.

4.4 A Solver Based on the Formulation by Lions and Glowinski

In Section 2.2 we presented an augmented formulation of the Bingham problem introduced by Duvaut and Lions, resulting in the problem (2.3) of saddle point type. Based on this formulation, Glowinski et. al. [34] advocated an iterative scheme of Uzawa-type for solving (2.3):

1. find $\mathbf{u} = \min_{\mathbf{v} \in H_0^1(\Omega)} \mathcal{L}(\mathbf{v}, \boldsymbol{\tau}, \boldsymbol{\gamma})$,
2. find $\boldsymbol{\tau} = \min_{\boldsymbol{\xi} \in \mathcal{L}_s^2} \mathcal{L}(\mathbf{u}, \boldsymbol{\xi}, \boldsymbol{\gamma})$,
3. update $\boldsymbol{\gamma}$ by $\boldsymbol{\gamma}_{new} = \boldsymbol{\gamma} + \lambda(D\mathbf{u} - \boldsymbol{\tau})$.

This results in the following algorithms, which we refer to as ALG:

Algorithm 4.2 ALG

Initialize $\boldsymbol{\gamma}_h = \boldsymbol{\gamma}_h^{(0)}$, $\boldsymbol{\tau}_h = \boldsymbol{\tau}_h^{(0)} \in \mathcal{W}_h$;

while convergence == false **do**

find $[\mathbf{u}_h, p_h]$ s.t.
$$\begin{cases} \int_{\Omega} \lambda D\mathbf{u}_h : D\mathbf{v}_h + b(p_h, \mathbf{v}_h) = \frac{1}{2} \int_{\Omega} (\boldsymbol{\tau}_h - 2\lambda\boldsymbol{\gamma}_h) : D\mathbf{v}_h + \int_{\Omega} \mathbf{f}\mathbf{v}_h, \\ b(q_h, \mathbf{u}_h) = 0. \end{cases}$$

Set $\boldsymbol{\gamma}_h = \begin{cases} 0 & \text{if } |\boldsymbol{\tau}_h + 2\lambda D\mathbf{u}_h| \leq \tau_s, \\ \left(1 - \frac{\tau_s}{|\boldsymbol{\tau}_h + 2\lambda D\mathbf{u}_h|}\right) \frac{\boldsymbol{\tau}_h + 2\lambda D\mathbf{u}_h}{2(\lambda + \mu)} & \text{otherwise.} \end{cases}$

Set $\boldsymbol{\tau}_h = \boldsymbol{\tau}_h + 2\lambda(D\mathbf{u}_h - \boldsymbol{\gamma}_h)$.

Test for convergence;

end while

There are different ways for performing the convergence test. One possibility is to consider the difference between $\boldsymbol{\tau}_h^{(k+1)}$ and $\boldsymbol{\tau}_h^{(k)}$ which corresponds to testing the residual $D\mathbf{u}_h^{(k)} - \boldsymbol{\gamma}_h^{(k)}$. As mentioned before, this scheme does not require a regularization, but the convergence rate depends on the choice of the parameter λ and can be very slow. In Section 5.3 we will present a numerical experiment in which ALG is applied to one of our test cases and where its performance is compared to the solver based on the mixed formulation.

Chapter 5

Performance of the Nonlinear Solver ¹

This chapter comprises several numerical experiments performed on the Bingham problem in the primitive and mixed formulation. We present three test cases in two dimensional domains. In the first test case, we consider a problem where the analytical solution for the Bingham problem is known. We then test our solver on the classical lid-driven cavity benchmark with a Bingham fluid flow. Our third experiment is a test on a non-rectangular geometry.

5.1 Description of the Test Cases

For the implementation we use the IFISS package [40] in MATLAB. Problems (4.1) and (4.2) are discretized using the inf-sup stable finite element pair Q2-Q1 for the velocity and pressure, respectively. For the discretization of the auxiliary variable W , no inf-sup constraint exists if $\varepsilon > 0$ (see Section 4.3), however possible existence of such a constraint is still open for $\varepsilon = 0$. We discretize W with Q1 finite elements since

¹Results in this chapter have been published in A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: Analysis and numerical solution*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2434-2446

this choice delivered convergent and accurate results. Starting with the zero vector as the initial guess, we perform Picard iterations until the initial residual \mathbf{r}_0 drops by six orders of magnitude, i.e. $\frac{\|\mathbf{r}_k\|_\infty}{\|\mathbf{r}_0\|_\infty} \leq 10^{-6}$. Unless stated otherwise, we choose $\mu = 1$, $\mathbf{f} = \mathbf{0}$ and $\Omega = [0, 1] \times [0, 1]$. MATLAB's backslash operator serves as the linear solver at each nonlinear iteration step.

5.2 An Analytical Test Case

The flow between two parallel plates is one of the few available analytical test cases for both the Stokes and Navier-Stokes type Bingham fluid flow. It is an analytical solution of (1.2) and (1.4). In two dimensions this analytical solution is given by

$$u_1 = \begin{cases} \frac{1}{8}[(1 - 2\tau_s)^2 - (1 - 2\tau_s - 2y)^2], & \text{if } 0 \leq y \leq \frac{1}{2} - \tau_s, \\ \frac{1}{8}(1 - 2\tau_s)^2, & \text{if } \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s, \\ \frac{1}{8}[(1 - 2\tau_s)^2 - (2y - 2\tau_s - 1)^2], & \text{if } \frac{1}{2} + \tau_s < y \leq 1, \end{cases} \quad (5.1)$$

$u_2 \equiv 0$ and $p = -x$. The plug region $\{y \in \Omega \mid \frac{1}{2} - \tau_s \leq y \leq \frac{1}{2} + \tau_s\}$ is the kernel moving at a constant velocity. In this experiment we choose $\tau_s = 0.3$. We impose Dirichlet boundary conditions on Ω according to (5.1). The number of Picard iterations for both the primitive and mixed formulation can be seen in Table 5.1. The comparison clearly emphasizes the advantage of using the mixed formulation. Results for the non-regularized case $\varepsilon = 0$ are included in the table as well. In particular, we notice that even for the non-regularized case the mixed formulation is convergent and produces accurate results. Figure 5.1 shows the numerical error in the discrete energy norm for different choices of ε and h (obtained by using the mixed formulation). By a direct computation, one realizes that the second derivative of the analytical solution (5.1) is discontinuous. Therefore we have that for the velocity solution \mathbf{u} from (5.1) it holds $\mathbf{u} \in \mathbf{H}^2$, but $\mathbf{u} \notin \mathbf{H}^3$. This limits the guaranteed order of convergence to

| $h \downarrow \setminus \varepsilon \rightarrow$ | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 0 |
|--|---------------------|-----------|-----------|-----------|-----------|-----------------|-----------|-----------|-----------|-----------|----|
| | Primitive Variables | | | | | Mixed Variables | | | | | |
| $\frac{1}{16}$ | 7 | 20 | 15 | 30 | 38 | 5 | 12 | 17 | 20 | 23 | 19 |
| $\frac{1}{32}$ | 7 | 23 | 60 | 81 | 83 | 4 | 12 | 19 | 24 | 25 | 24 |
| $\frac{1}{64}$ | 7 | 27 | 89 | 95 | 88 | 4 | 9 | 14 | 16 | 16 | 22 |
| $\frac{1}{128}$ | 6 | 23 | 61 | 134 | 192 | 3 | 8 | 11 | 12 | 12 | 13 |

Table 5.1: Number of Picard iterations required for reducing the residual by a factor of 10^{-6} in the analytical test case for $\tau_s = 0.3$ for different choices of ε and h in both the formulation in primitive and mixed variables. The last column shows the results for the non-regularized case which is not applicable in the primitive formulation.

$O(h)$ in the energy norm. Large values of ε with small values of h clearly prevent the optimal convergence rate of the finite element method. For smaller values of ε the observed order of convergence is between one and two. Table 5.2 compares the numerical error of the primitive and mixed formulation in the discrete energy norm. The table indicates that slightly more accurate solutions are computed by the formulation in primitive variables. It should be kept in mind, however, that the primitive formulation needs significantly more iterations to converge to this solution. Figure 5.2, top left, shows the computed pressure field. The pressure points out a numerical error in the area around the plug region. This is expected since in the model the stress tensor is under-determined in the plug zone. The other subplots in the figure illustrate the velocity error for different values of τ_s , pointing out error spikes in the neighborhood of the plug region. Note however that the maximum error is of the order of 10^{-5} .

5.3 The Lid-Driven Cavity

We perform two different numerical simulations, both in the unit square domain $\Omega = [0, 1] \times [0, 1]$. For the first case, we solve (1.2) and (1.3) with $\rho = 0$ imposing Dirichlet boundary conditions by $\mathbf{u}|_{y=1} = (0, 1)^T$ and $\mathbf{u} = \mathbf{0}$ everywhere else. Table 5.3 shows the number of Picard iterations for different sizes of the mesh and different

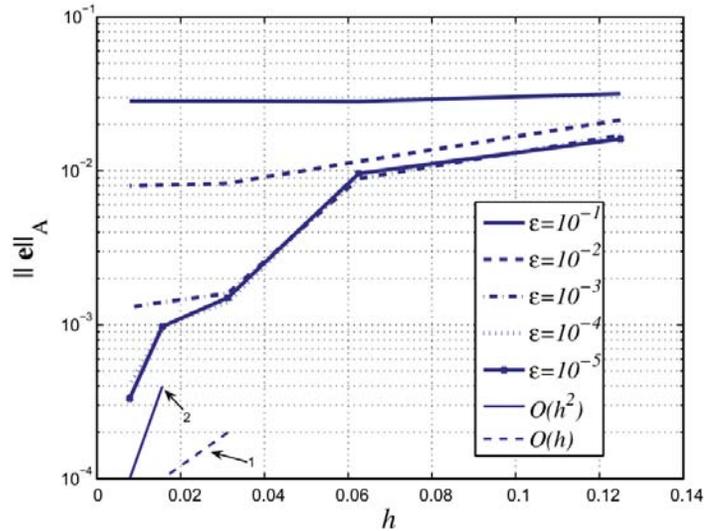


Figure 5.1: The numerical error $\|\mathbf{u} - \mathbf{u}_{ex}\|_A = \sqrt{(\mathbf{u} - \mathbf{u}_{ex})^T A (\mathbf{u} - \mathbf{u}_{ex})}$ of the mixed formulation for different sizes of the mesh and different ε . The matrix A is the finite element velocity stiffness matrix. The thin solid line is a reference line for $O(h^2)$, the thin dotted line for $O(h)$.

| $h \downarrow \setminus \varepsilon \rightarrow$ | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} |
|--|-----------|-----------|-----------|-----------|-----------|
| Primitive Variables | | | | | |
| $h = \frac{1}{16}$ | 0.0280 | 0.0093 | 0.0058 | 0.0060 | 0.0060 |
| $h = \frac{1}{32}$ | 0.0282 | 0.0076 | 0.0016 | 0.0013 | 0.0013 |
| $h = \frac{1}{64}$ | 0.0282 | 0.0075 | 0.0013 | 7.525e-04 | 7.290e-04 |
| $h = \frac{1}{128}$ | 0.0282 | 0.0074 | 0.0012 | 3.252e-04 | 2.349e-04 |
| Mixed Variables | | | | | |
| $h = \frac{1}{16}$ | 0.0283 | 0.0108 | 0.0070 | 0.0082 | 0.0082 |
| $h = \frac{1}{32}$ | 0.0284 | 0.0082 | 0.0018 | 0.0015 | 0.0016 |
| $h = \frac{1}{64}$ | 0.0284 | 0.0081 | 0.0016 | 0.0010 | 9.925e-04 |
| $h = \frac{1}{128}$ | 0.0284 | 0.0081 | 0.0013 | 5.114e-04 | 4.857e-04 |

Table 5.2: The numerical error $\|\mathbf{u} - \mathbf{u}_{ex}\|_A$ of the primitive (left) and mixed (right) formulation for different sizes of the mesh and different values of ε (A is a finite element velocity stiffness matrix).

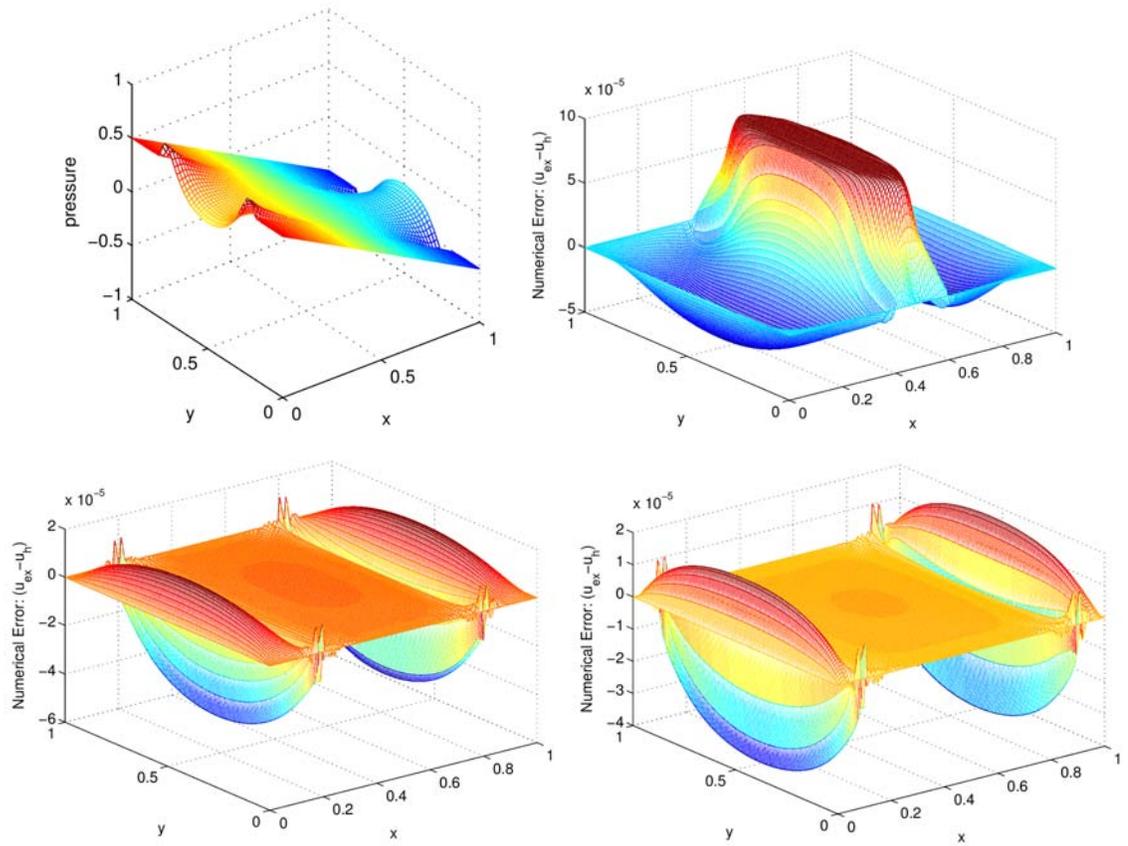


Figure 5.2: Top, left: the pressure field of the analytical test case computed with $\tau_s = 0.3$, $h = \frac{1}{128}$ and $\varepsilon = 10^{-5}$. Top, right: velocity error for $\tau_s = 0.1$. Bottom: velocity error for $\tau_s = 0.3$ and $\tau_s = 0.4$.

values of τ_s and ε . Again the comparison between the mixed and the primitive formulation demonstrates the effectiveness of the former; for the formulation in primitive variables the method does not converge within 500 iterations in some cases (denoted in the table by -). We also compare the numerical results of the mixed formulation with $\varepsilon = 0$ to the augmented formulation by Lions and Glowinski described in Section 2.2. The latter involves a parameter λ . In Figure 5.3, left, we compare the dynamics of the reduction of the difference $\|\boldsymbol{\tau}^{(k)} - \boldsymbol{\tau}^{(k-1)}\|$ along the iterations for several values of λ . The best choice for λ is $\lambda = 0.01$. Figure 5.3 on the right illustrates the dynamics of $\|\boldsymbol{\tau}^{(k)} - \boldsymbol{\tau}^{(k-1)}\|$ along 200 iterations of the non-regularized version of (4.2), i.e. $\varepsilon = 0$ and the Lions-Glowinski method with $\lambda = 0.01$. Note that in previous results we used the Picard residual as a stopping criterion, whereas here we check the difference between two consecutive computations of the stress. This is consistent with the common stopping criterion of ALG. We use $\tau_s = 2$ and $h = \frac{1}{32}$. The convergence rate of the mixed formulation is better than the one of Lions and Glowinski measured in terms of $\|\boldsymbol{\tau}^{(k)} - \boldsymbol{\tau}^{(k-1)}\|$ (notice however that the Picard residual indicates a convergence to the same solution in fewer iterations). In general the identification of the optimal parameter λ is not easy whereas the solver of the mixed formulation is parameter free.

In the second experiment, we solve the unsteady Navier-Stokes type Bingham problem, i.e. (1.2) and (1.3) with $\rho \neq 0$. We take $\rho = 1$ for simplicity. We impose Dirichlet boundary conditions by $\mathbf{u}|_{y=1} = (10, 0)^T$ and homogeneous Dirichlet boundary conditions everywhere else. We choose $\Delta t = 0.1$ as the time step, $h = \frac{1}{128}$, $\varepsilon = 10^{-5}$, $\tau_s = 2$ and $\mu = 0.1$. In Figure 5.4 we show the streamlines of the solution of the Stokes type problem with $\mu = 1$ as well as the solution of the Navier-Stokes type problems with $\mu = 0.1$ and $\mu = 0.01$ at time $t = 1.0$ after the solution has reached a steady state. Furthermore, in Figure 5.5 we show equally distributed velocity streamlines for the Stokes type problem together with the isoline of $(2\mu + \frac{\tau_s}{|D\mathbf{u}|_\varepsilon})|D\mathbf{u}| = \tau_s$, which may be

| $h \downarrow \setminus \varepsilon \rightarrow$ | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 10^{-1} | 10^{-2} | 10^{-3} | 10^{-4} | 10^{-5} | 0 |
|--|---------------------|-----------|-----------|-----------|-----------|-----------------|-----------|-----------|-----------|-----------|----|
| | Primitive Variables | | | | | Mixed Variables | | | | | |
| $\tau_s = 2:$ | | | | | | | | | | | |
| $\frac{1}{16}$ | 22 | 49 | 51 | 51 | 51 | 11 | 21 | 26 | 27 | 27 | 21 |
| $\frac{1}{32}$ | 99 | 173 | 224 | 224 | 224 | 10 | 15 | 17 | 17 | 17 | 23 |
| $\frac{1}{64}$ | 213 | - | - | - | - | 8 | 12 | 12 | 12 | 12 | 15 |
| $\frac{1}{128}$ | - | - | - | - | - | 7 | 9 | 9 | 9 | 9 | 11 |
| $\tau_s = 5:$ | | | | | | | | | | | |
| $\frac{1}{16}$ | 18 | 37 | 48 | 51 | 51 | 17 | 31 | 37 | 37 | 38 | 27 |
| $\frac{1}{32}$ | 66 | 94 | 269 | 267 | 266 | 14 | 22 | 23 | 24 | 24 | 32 |
| $\frac{1}{64}$ | 128 | - | - | - | - | 12 | 17 | 18 | 18 | 18 | 22 |
| $\frac{1}{128}$ | - | - | - | - | - | 10 | 13 | 14 | 14 | 14 | 15 |

Table 5.3: Number of Picard iterations required for reducing the residual by 10^{-6} for the formulation in primitive variables (left) and the formulation in mixed variables (right) for the lid-driven cavity. For $h = \frac{1}{128}$ and in many cases of $h = \frac{1}{64}$ the iterations in primitive variables do not converge (denoted by -). The non-regularized case $\varepsilon = 0$ is included in the last column for the mixed formulation.

interpreted as a reasonable approximation of the yield surfaces [76]. Results presented in Figure 5.5 are consistent with those found in literature (see for example [65]).

5.4 A Computation on Non-Rectangular Domains

We finally present two simple test cases carried out in non-rectangular geometries, showing that the mixed formulation even with no regularization is a viable approach for more realistic problems. Simulations are carried out with the software **FreeFem++** (version 3.9), P1 bubble finite elements for the velocity, P1 for the pressure and each component of the tensor W . A streamline upwind Petrov-Galerkin stabilization [21] has been added for the treatment of the convective term (linearized with a Picard approach). In the first case we simulate a two dimensional circular cavity attached to a two dimensional rectangular channel. In the second case we consider a rectangular channel with an obstacle represented by a sinusoidal bump. In Figure 5.6, top, we show the computed shear rate for the first geometry with $\tau_s = 1$, $\mu = 0.05$ and an incoming velocity profile with peak $u_M = 2$. In Figure 5.6, bottom, we present results

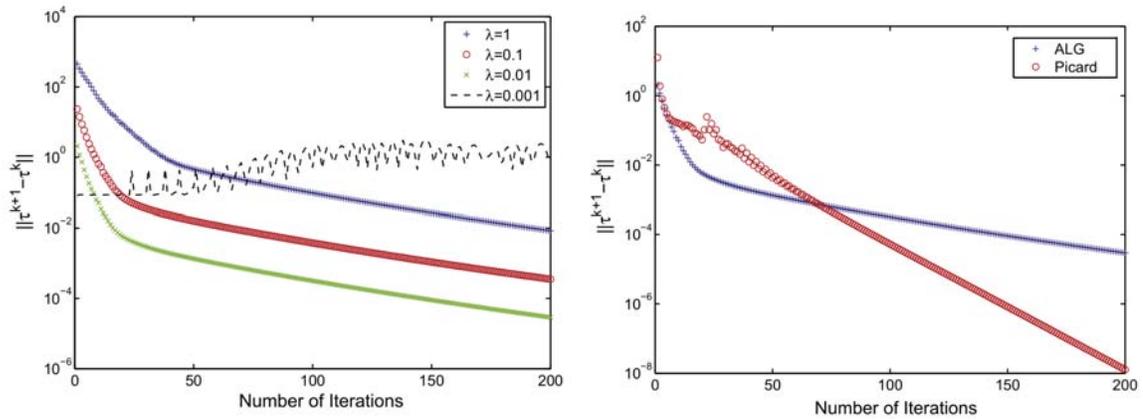


Figure 5.3: Reduction of the difference $\|\tau^{(k)} - \tau^{(k-1)}\|$ for the first 200 iterations when solving the lid-driven cavity problem with $\tau_s = 2$ and $h = \frac{1}{32}$. Left: ALG with different choices of λ . No convergence occurs in the case of $\lambda = 0.001$. Right: Comparison between the Picard iterations (4.2) with $\varepsilon = 0$ and ALG with $\lambda = 0.01$.

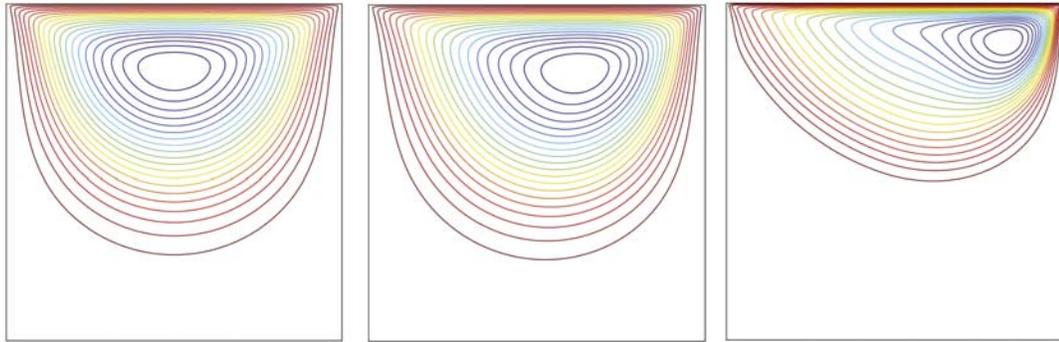


Figure 5.4: Streamlines of the lid-driven cavity flow ($\tau_s = 2$, $h = \frac{1}{128}$ and $\varepsilon = 10^{-5}$). Solution of the Stokes type problem. Center and Right: solution of the Navier-Stokes type problem computed with $\mathbf{u}|_{y=0} = (10, 0)^T$, $\mu = 0.1$ (center) and $\mu = 0.01$ (right).

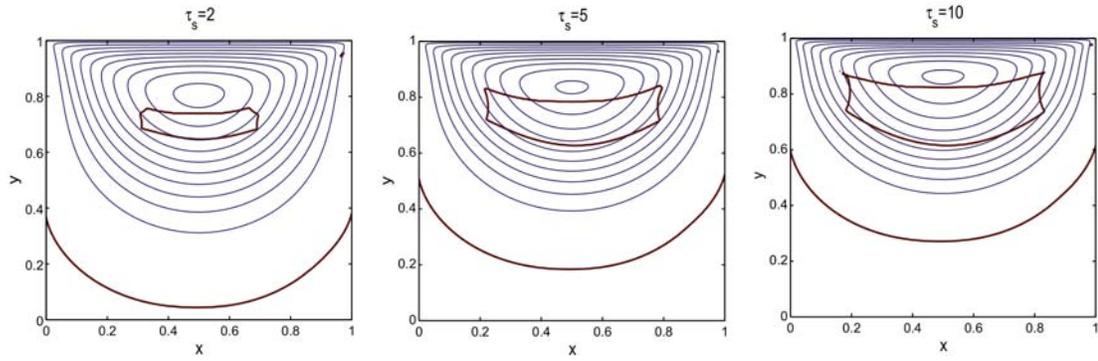


Figure 5.5: Velocity streamlines (blue) and predicted yield surfaces (brown) of the lid-driven cavity flow computed for different values of the yield stress τ_s with $h = \frac{1}{64}$. Images by Maxim A. Olshanskii, Department of Mechanics and Mathematics, Moscow M. V. Lomonosov State University.

for the second geometry with $\tau_s = 1$, $\mu = 0.1$ and $u_M = 10$. The non-regularized mixed formulation produces convergent results in all these test cases.

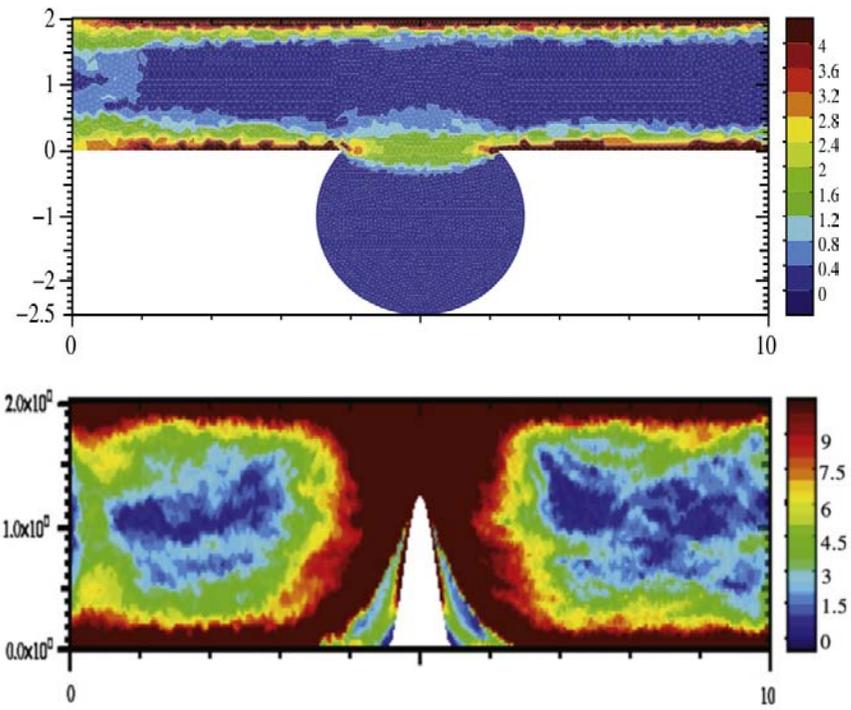


Figure 5.6: Plot of $|Du|$ for non-trivial domains. Top: Circular cavity attached to a rectangular pipe. Bottom: An occluded pipe. Results obtained using the mixed non-regularized formulation.

Chapter 6

Preconditioning ¹

We have seen in Chapter 5 that the nonlinear iterative solver based on the mixed formulation of the Bingham problem converges – for “academic” test cases – within a small number of iterations and is robust with respect to the regularization parameter ε and the size of the mesh. Also, the solver works well for the non-regularized case $\varepsilon = 0$. The price of solving the mixed formulation as opposed to the formulation in primitive variables is that additional unknowns augment the linear system, that is, the linear systems to be solved at each nonlinear iteration are larger than those for the formulation in primitive variables. Note that our numerical experiments from Chapter 5 are performed on two dimensional test problems on a relatively small scale and a direct method is used for solving the linear systems. Direct methods, however, are usually unsuitable for realistic problems which are normally on a larger scale because of their poor scaling properties and an excessive memory requirement for larger systems. This is in particular the case when solving three dimensional problems, when direct methods become unaffordable even for problems of moderate size. The purpose of this chapter is to introduce a more sophisticated linear solver which is suitable for problems on larger scale, in particular for problems in three dimensions.

¹The work presented in this chapter was performed in close collaboration with Panayot Vasilevski, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.

More precisely, we will solve linear systems arising from the non-regularized Bingham problem using a Krylov subspace method. We will introduce a preconditioner which results in a fast and robust convergence of the linear iterative solver within a small number of iterations independent of the mesh size.

We begin this chapter by explaining the concept of two important methods in numerical analysis, multigrid methods and domain decomposition methods. We then provide an overview over the linear systems to be solved and the existing methods for preconditioning them in Section 6.3. In Section 6.4 we suggest using the regularized Bingham problem as a preconditioner for solving the non-regularized problem. For efficient preconditioning of the linear system, the regularized Bingham problem needs to be approximated. This can be done in different ways. We present two of these possibilities in this chapter, the inexact factorization of the system in Section 6.5 and geometric multigrid in Section 6.6.

6.1 Multigrid Methods

We start by introducing multigrid methods, which are often used as a preconditioning technique that will be of high importance in this thesis. We follow [20] for explaining the concept of multigrid.

We start with considering a general linear system of the form

$$\mathcal{A}\mathbf{x} = \mathbf{f}. \tag{6.1}$$

For now we want to assume that the matrix \mathcal{A} is obtained by discretizing a partial differential equation on a given domain, i.e. the matrix \mathcal{A} is associated to a mesh. When it is not feasible to compute the exact solution \mathbf{x} to (6.1), some approximation $\tilde{\mathbf{x}}$ is found. The corresponding (unknown) error is given by $\mathbf{e} = \mathbf{x} - \tilde{\mathbf{x}}$. We may

compute the residual $\mathbf{r} = \mathcal{A}\tilde{\mathbf{x}} - \mathbf{f}$. The *residual equation*

$$\mathcal{A}\mathbf{e} = \mathbf{r} \tag{6.2}$$

can be easily derived from the considerations above. Suppose now that an approximation $\tilde{\mathbf{x}}$ to (6.1) has been computed. The approximation can be improved by solving (6.2) for \mathbf{e} and then updating \mathbf{x} by

$$\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{e}. \tag{6.3}$$

One way to find an approximate solution to (6.1) is by applying a fixed number of iterations of a stationary iterative method (see Appendix A). When testing stationary iterative methods on cases where the analytical solution (hence the error at each iteration is known), it can be observed that the norm of the error drops quickly for the first few iterations, but reduces slowly for the proceeding iterations. This is due to the fact that stationary iterative methods tend to “smooth” out the *high frequency modes* of the error, whereas the *low frequency modes* are damped significantly slower. More precisely, if we assume that the error corresponding to the initial guess has the form

$$\mathbf{e} = \{e_i\}_{i=1}^n = \sum_{k=1}^n \alpha_k \left\{ \sin \left(\frac{ki\pi}{n} \right) \right\}_{i=1}^n, \quad \alpha_k \in \mathbb{R},$$

then those components of the error corresponding to large values of k (i.e. the high frequency modes) are damped rapidly while those components corresponding to the lower values of k (the low frequency modes) decrease much more slowly. Whether a frequency mode is to be considered high or low depends on the size of the mesh $h = 1/n$. The property of eliminating high frequency modes and leaving low frequency modes is called the *smoothing property*.

The idea behind multigrid is to first apply a fixed number of an iterative scheme

satisfying the smoothing property to (6.1). This will eliminate high frequency modes and only low to medium frequency modes will remain. We call this step *pre-smoothing* and the iterative scheme the *smoother*. If the obtained approximation is represented on a coarser mesh – this can be done in different ways, e.g. by restriction – some of the modes which were medium frequency modes on the fine mesh now appear as higher frequency modes on the coarse mesh and applying again the smoother will damp these components. The new approximation is then transferred back to the fine mesh (either by interpolation or by projection). In practice, after the pre-smoothing step, we approximate the corresponding error on the coarse mesh using the residual equation (6.2). This approximation of the error is interpolated to the fine mesh and used to correct the solution obtained from the smoothing step using (6.3). We call this the *coarse-grid correction* step. After the coarse-grid correction step, we again apply the smoother to the updated solution. This is called *post-smoothing*. These steps yield the following general two-level multigrid algorithm, letting \mathcal{A}_c be the coarse-grid representation of \mathcal{A} and \mathbf{x}_0 be some initial guess:

Algorithm 6.1 Two-Level Multigrid

- 1: obtain approximation $\tilde{\mathbf{x}}$ by smoothing ν times on $\mathcal{A}\mathbf{x}_0 = \mathbf{f}$;
 - 2: compute residual $\mathbf{r} = \mathcal{A}\tilde{\mathbf{x}} - \mathbf{f}$;
 - 3: find \mathbf{r}_c , the coarse-grid representation of \mathbf{r} ;
 - 4: solve the residual equation $\mathcal{A}_c\mathbf{e}_c = \mathbf{r}_c$;
 - 5: find \mathbf{e} , the fine-grid representation of \mathbf{e}_c ;
 - 6: update $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{e}$;
 - 7: smooth ν times on $\mathcal{A}\mathbf{x} = \mathbf{f}$;
-

Note that if the size of the coarse-grid matrix \mathcal{A}_c in line 4 of Algorithm 6.1 is small enough, the residual equation may be solved exactly by a direct method. If \mathcal{A}_c is still too large for a direct solve, the residual equation may be approximated by applying again the multigrid method in a recursive fashion. This yields a multigrid V-cycle

which may be defined recursively. More generally, we may approximate the error \mathbf{e}_c by applying the multigrid method σ times to $\mathcal{A}_c \mathbf{x}_c = \mathbf{r}_c$, resulting in a multigrid W-cycle: Let \mathbf{y} be a generic input vector, \mathbf{x} the corresponding output vector and ν , σ given constants. Let further ℓ denote the current level in the multigrid scheme. We assume that the linear system matrix \mathcal{A} can be represented on L multigrid levels by $\{\mathcal{A}_\ell\}_{\ell=1}^L$. The multigrid W-cycle then looks like

Algorithm 6.2 Multigrid W-Cycle

```

WCycle( $\{\mathcal{A}_\ell\}_\ell, \mathbf{x}, \mathbf{y}, \nu, \sigma, \ell$ ):
  iterate  $\nu$  times on  $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$ ; //pre-smoothing
  compute residual  $\mathbf{r} = \mathbf{b} - \mathcal{A}\mathbf{x}$ ;
  project  $\mathbf{r}$  a coarser grid,  $\mathbf{r}_{\ell-1}$ ;
  //coarse-grid correction:
  if  $\ell - 1 == 1$  then
    solve  $\mathcal{A}_{\ell-1} \mathbf{x}_c = \mathbf{r}_{\ell-1}$  directly;
  else
    call  $\sigma$  times WCycle( $\{\mathcal{A}_\ell\}_\ell, \mathbf{x}_c, \mathbf{r}_{\ell-1}, \nu, \sigma, \ell - 1$ );
  end if
  iterate  $\nu$  times on  $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$ ; //post-smoothing

```

Figure 6.1 visualizes the levels visited by the W-cycle on four levels of multigrid for $\sigma = 1$ (V-cycle) and $\sigma = 2$.

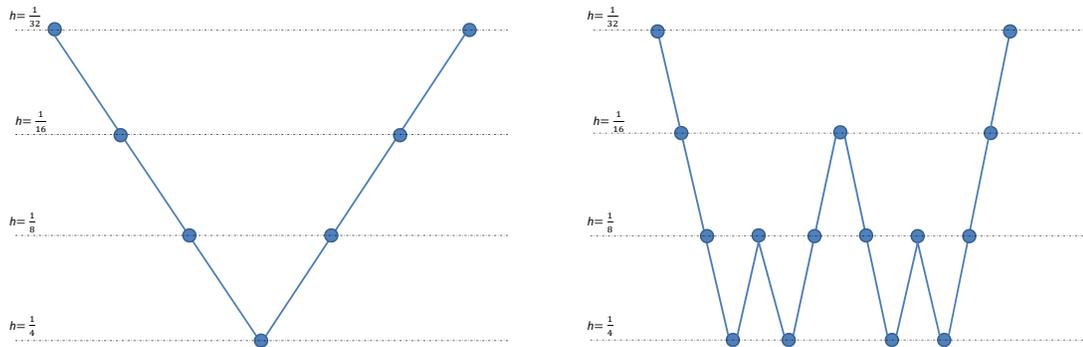


Figure 6.1: Levels visited by Algorithm 6.2 on four levels of multigrid with $\sigma = 1$ (left) and $\sigma = 2$ (right).

We refer to [20] for a more detailed approach to multigrid methods. Convergence theory and some theoretical results can be found in [52].

6.2 Domain Decomposition Methods

Another important class of methods we will use for the linear solver are domain decomposition methods. The concept will be explained following [90].

The first method based on a decomposition of the domain was introduced by Schwarz in 1870. His method, referred to as the *alternating Schwarz* method, can be used to solve elliptic boundary value problems on domains that are the union of two subdomains by alternatingly solving the elliptic boundary value problem restricted to the individual subdomains. Consider the domain $\Omega = \Omega_1 \cup \Omega_2$ as shown in Figure 6.2 on which we wish to solve the linear elliptic PDE

$$\begin{cases} Lu = f & \text{in } \Omega, \\ u = g & \text{on } \partial\Omega. \end{cases} \quad (6.4)$$

We consider here Dirichlet boundary conditions, but other types of boundary

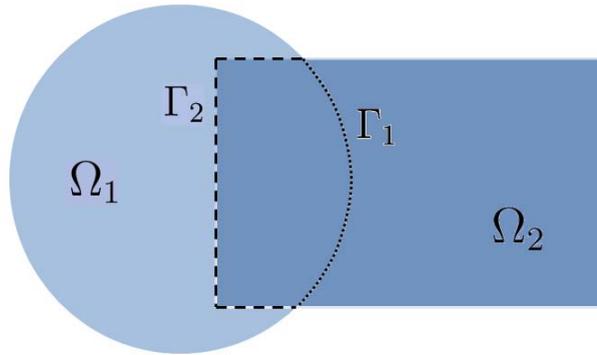


Figure 6.2: Original figure by Schwarz [90].

conditions are possible as well. The artificial boundaries are denoted by Γ_1 and Γ_2 , as shown in Figure 6.2. Let $u_i^{(k)}$ denote the approximate solution of (6.4) on the subdomain $\overline{\Omega}_i$, $i = 1, 2$. We begin the alternating Schwarz iterations by selecting an initial guess $u_2^{(0)}$ for the solution in subdomain Ω_2 . We then solve for $k = 1, 2, \dots$ the boundary value problem

$$\begin{cases} Lu_1^{(k)} = f & \text{in } \Omega_2 \\ u_1^{(k)} = g & \text{on } \partial\Omega_1 \setminus \Gamma_1 \\ u_1^{(k)} = u_2^{(k-1)}|_{\Gamma_1} & \text{on } \Gamma_1 \end{cases}$$

for $u_1^{(k)}$. Once $u_1^{(k)}$ is known, the boundary value problem

$$\begin{cases} Lu_2^{(k)} = f & \text{in } \Omega_2 \\ u_2^{(k)} = g & \text{on } \partial\Omega_2 \setminus \Gamma_2 \\ u_2^{(k)} = u_1^{(k)}|_{\Gamma_2} & \text{on } \Gamma_2 \end{cases}$$

is solved for $u_2^{(k)}$.

For the numerical solution, (6.4) needs to be discretized, resulting in a linear system of the form (6.1). Let \mathcal{A} be the discretization of the operator L . Also, we consider

the operator L restricted to the local subdomain Ω_i and denote the corresponding discrete operator by \mathcal{A}_i . Since in the overlap region the meshes may not be matching, it is necessary to define an interpolation operator $\mathcal{I}_{\Omega_j \rightarrow \Gamma_i}$, $i, j = 1, 2$ for determining the internal boundary on the artificial boundaries Γ_1 and Γ_2 . The operator $\mathcal{I}_{\Omega_j \rightarrow \Gamma_i}$, $i, j = 1, 2$ interpolates values from Ω_j to the boundary Γ_i . We are now ready to state the algorithm for the discretized version of the alternating Schwarz method:

Algorithm 6.3 Alternating Overlapping Schwarz Method

$\mathbf{w}_1^0 \leftarrow \mathbf{0}$

while (convergence==false) **do**

 solve

$$\begin{cases} \mathcal{A}_1 \mathbf{x}_1^{(k)} = \mathbf{f}_1 & \text{on } \Omega_1 \\ \mathbf{x}_{\partial\Omega_1 \setminus \Gamma_1}^{(k)} = \mathbf{g}_1 & \text{on } \partial\Omega_1 \setminus \Gamma_1 \\ \mathbf{x}_{\Gamma_1}^{(k)} = \mathbf{w}_1^{(k-1)} & \text{on } \Gamma_1 \end{cases}$$

 for $\mathbf{x}_1^{(k)}$;

 interpolate $\mathbf{w}_2^{(k)} \leftarrow \mathcal{I}_{\Omega_1 \rightarrow \Gamma_2} \mathbf{x}_{\Omega_1}^{(k)}$;

 solve

$$\begin{cases} \mathcal{A}_2 \mathbf{x}_2^{(k)} = \mathbf{f}_2 & \text{on } \Omega_2 \\ \mathbf{x}_{\partial\Omega_2 \setminus \Gamma_2}^{(k)} = \mathbf{g}_2 & \text{on } \partial\Omega_2 \setminus \Gamma_2 \\ \mathbf{x}_{\Gamma_2}^{(k)} = \mathbf{w}_2^{(k)} & \text{on } \Gamma_2 \end{cases}$$

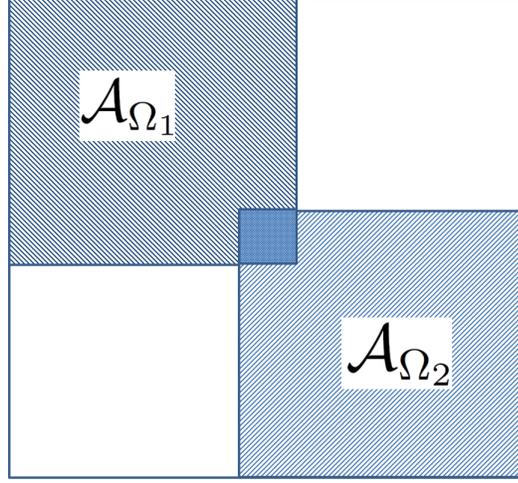
 for $\mathbf{x}_2^{(k)}$;

 interpolate $\mathbf{w}_1^{(k)} \leftarrow \mathcal{I}_{\Omega_2 \rightarrow \Gamma_1} \mathbf{x}_{\Omega_2}^{(k)}$;

 check for convergence;

end while

The convergence rate of Algorithm 6.3 depends on the size of the overlap region; more overlap results in faster convergence of the method. In case the grids in the subdomains are matching, the interpolation is not necessary. Our goal is now to eliminate the explicit dependence on the artificial boundary from the description of

Figure 6.3: Partitioning of the matrix \mathcal{A} .

the algorithm. Let us reorder the matrix \mathcal{A} in the following ways (see Figure 6.2):

$$\mathcal{A}_{reordered,1} = \begin{bmatrix} \mathcal{A}_{\Omega_1} & \mathcal{A}_{\Omega \setminus \Omega_1} \\ * & * \end{bmatrix} = \begin{bmatrix} \mathcal{A}_{\Omega_1} & \mathcal{A}_{\Gamma_1} & \mathcal{A}_{\Omega \setminus \overline{\Omega_1}} \\ * & * & * \end{bmatrix}$$

and

$$\mathcal{A}_{reordered,2} = \begin{bmatrix} * & * \\ \mathcal{A}_{\Omega \setminus \Omega_2} & \mathcal{A}_{\Omega_2} \end{bmatrix} = \begin{bmatrix} * & * & * \\ \mathcal{A}_{\Omega_1 \setminus \overline{\Omega_2}} & \mathcal{A}_{\Gamma_2} & \mathcal{A}_{\Omega_2} \end{bmatrix}.$$

In general, \mathcal{A}_{Ω_1} and \mathcal{A}_{Ω_2} will contain some common entries from degrees of freedom in the interior of the overlap region. These degrees of freedom will generate identical coefficients in the matrices \mathcal{A}_{Ω_1} and \mathcal{A}_{Ω_2} (in Figure 6.2, these coefficients lie in the small square at the intersection of \mathcal{A}_{Ω_1} and \mathcal{A}_{Ω_2}). With this reordering, we may write Algorithm 6.3 in the iterative form

$$\begin{aligned} \mathbf{x}_{\Omega_1}^{(k)} &\leftarrow \mathbf{x}_{\Omega_1}^{(k-1)} + \mathcal{A}_{\Omega_1}^{-1} \left(\mathbf{f}_1 - \mathcal{A}_{\Omega_1} \mathbf{x}_{\Omega_1}^{(k-1)} - \mathcal{A}_{\Gamma_1} \mathbf{x}_{\Gamma_1}^{(k-1)} \right), \\ \mathbf{x}_{\Omega_2}^{(k)} &\leftarrow \mathbf{x}_{\Omega_2}^{(k-1)} + \mathcal{A}_{\Omega_2}^{-1} \left(\mathbf{f}_2 - \mathcal{A}_{\Omega_2} \mathbf{x}_{\Omega_2}^{(k-1)} - \mathcal{A}_{\Gamma_2} \mathbf{x}_{\Gamma_2}^{(k)} \right). \end{aligned} \tag{6.5}$$

If we make the additional assumption that there is no direct coupling between

the degrees of freedom on opposite sides of the artificial boundaries, then $\mathcal{A}_{\Omega_2 \setminus \overline{\Omega_1}} = 0$ and $\mathcal{A}_{\Omega_1 \setminus \overline{\Omega_2}} = 0$ and we observe that $\mathcal{A}_{\Gamma_1} \mathbf{x}_{\Gamma_1}^{(k-1)} \equiv \mathcal{A}_{\Omega \setminus \Omega_1} \mathbf{x}_{\Omega \setminus \Omega_1}^{(k-1)}$ and $\mathcal{A}_{\Gamma_2} \mathbf{x}_{\Gamma_2}^{(k-1)} \equiv \mathcal{A}_{\Omega \setminus \Omega_2} \mathbf{x}_{\Omega \setminus \Omega_2}^{(k-1)}$. This way, we may replace all terms involving explicitly the artificial boundaries Γ_i in (6.5) and obtain the formulation

$$\begin{aligned} \mathbf{x}_{\Omega_1}^{(k)} &\leftarrow \mathbf{x}_{\Omega_1}^{(k-1)} + \mathcal{A}_{\Omega_1}^{-1} \left(\mathbf{f}_1 - \mathcal{A}_{\Omega_1} \mathbf{x}_{\Omega_1}^{(k-1)} - \mathcal{A}_{\Omega \setminus \Omega_1} \mathbf{x}_{\Omega \setminus \Omega_1}^{(k-1)} \right), \\ \mathbf{x}_{\Omega_2}^{(k)} &\leftarrow \mathbf{x}_{\Omega_2}^{(k-1)} + \mathcal{A}_{\Omega_2}^{-1} \left(\mathbf{f}_2 - \mathcal{A}_{\Omega_2} \mathbf{x}_{\Omega_2}^{(k-1)} - \mathcal{A}_{\Omega \setminus \Omega_2} \mathbf{x}_{\Omega \setminus \Omega_2}^{(k-1)} \right). \end{aligned} \quad (6.6)$$

Iteration (6.6) results in the *multiplicative Schwarz* algorithm:

Algorithm 6.4 Multiplicative Schwarz Method

choose initial guess $\mathbf{x}^{(0)}$;

while (convergence == false) **do**

$$\mathbf{x}^{(k+1/2)} = \mathbf{x}^{(k)} + \begin{bmatrix} \mathcal{A}_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} (\mathbf{f} - \mathcal{A}\mathbf{x}^{(k)});$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k+1/2)} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{A}_{\Omega_2}^{-1} \end{bmatrix} (\mathbf{f} - \mathcal{A}\mathbf{x}^{(k+1/2)});$$

check for convergence;

$k \leftarrow k + 1$;

end while

When writing Algorithm 6.4 in matrix form it can be viewed as a block Gauss-Seidel iterative method. A major limitation of Algorithm 6.4 is that the computation on subdomain Ω_2 depends on the result obtained on Ω_1 , making it unsuitable for parallel computations. This issue may be overcome by considering the corresponding block Jacobi method. This yields the *overlapping additive Schwarz* method. It can be written in iterative form as

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \left(\begin{bmatrix} \mathcal{A}_{\Omega_1}^{-1} & 0 \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{A}_{\Omega_2}^{-1} \end{bmatrix} \right) (\mathbf{f} - \mathcal{A}\mathbf{x}).$$

The computation of the local systems \mathcal{A}_{Ω_1} and \mathcal{A}_{Ω_2} is independent and may be computed on separate processors, allowing a parallel implementation of the method. The additive Schwarz method does not converge in general, thus in practice it is often used as a preconditioner inside a Krylov method.

For simplicity, we explained the concept of domain decomposition methods on two domains, but the extension to several domains follows in a natural way. For the additive Schwarz method, the generalizes to

$$\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} + \left(\begin{aligned} & \begin{bmatrix} \mathcal{A}_{\Omega_1}^{-1} & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \mathcal{A}_{\Omega_2}^{-1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} \\ & + \cdots + \begin{bmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{A}_{\Omega_n}^{-1} \end{bmatrix} \end{aligned} \right) (\mathbf{f} - \mathcal{A}\mathbf{x}) \quad (6.7)$$

We will use the additive Schwarz method (6.7) for the preconditioner presented in this thesis. Further modifications of the method exist, such as the *restricted additive Schwarz* method which was introduced with the purpose of reducing the communication in parallel computations [24]. For a more detailed introduction to domain decomposition methods we refer to [90], some theoretical analysis can be found in [95].

6.3 The General Concept of Preconditioning

We now focus on the preconditioning of the linearized and discretized Bingham problem. In our computations, we will use a pressure stabilization term, that is, \mathcal{A} is

given as in (4.21):

$$\mathcal{A}_\varepsilon(\mathbf{u}) = \begin{bmatrix} A & B^T & C^T \\ B & -\alpha Q & 0 \\ C & 0 & -N_\varepsilon(\mathbf{u}) \end{bmatrix}.$$

The efficient solution of this system with complex geometries or a large number of degrees of freedom can be obtained either with an approximate factorization of \mathcal{A}_ε , resorting to a sequential computation of velocity, pressure and the tensor W , or with an efficient preconditioner. In the latter case, we could take advantage of the twofold saddle point-structure of the problem. As a matter of fact, notice that there are two different ways to recognize the saddle-point structure of (4.21). Letting

$$\mathcal{B} = \begin{bmatrix} B \\ C \end{bmatrix} \text{ and } \mathcal{N}_\varepsilon = \begin{bmatrix} -\alpha Q & 0 \\ 0 & -N_\varepsilon(\mathbf{u}^{k-1}) \end{bmatrix}$$

gives a saddle-problem of the form

$$\mathcal{A}_\varepsilon = \begin{bmatrix} A & \mathcal{B}^T \\ \mathcal{B} & \mathcal{N}_\varepsilon \end{bmatrix} \tag{6.8}$$

with a positive definite (1,1)-block, which is also symmetric in the case of the Stokes-type problem. On the other hand, one may define

$$\mathcal{F} = \begin{bmatrix} A & B^T \\ B & -\alpha Q \end{bmatrix} \text{ and } \mathcal{G} = \begin{bmatrix} C & 0 \end{bmatrix}.$$

In this case, the problem becomes

$$\mathcal{A}_\varepsilon = \begin{bmatrix} \mathcal{F} & \mathcal{G}^T \\ \mathcal{G} & -N_\varepsilon(\mathbf{u}^{k-1}) \end{bmatrix} \tag{6.9}$$

and the (1,1)-block of the saddle-point problem is indefinite and represents in turn a saddle-point problem. Many preconditioners have been suggested for saddle-point problems either when the matrix (1,1)-block of the system is s.p.d (symmetric positive definite), or its symmetric part is s.p.d. A broad spectrum of preconditioners relies on inexact factorizations of the system and an approximation of the Schur complement, such as the least square commutator preconditioner or the pressure convection diffusion preconditioner [40, 41]. Other preconditioning techniques for saddle-point problems include augmented Lagrangian preconditioners [14, 15] or preconditioners based on a dimensional splitting [12, 13].

6.4 Approximating the Non-Regularized Problem

We have seen so far that one of the main advantages of the mixed formulation is the treatment of the singularity represented by the plug regions. The numerical solver based on this formulation is more robust with respect to the regularization parameter. Also, the numerical results from Chapter 5 show that the mixed formulation can be used for the non-regularized case. The idea we pursue here is to regard *the regularized mixed formulation of the Bingham problem as a preconditioner for the non-regularized case*. In other terms, we use a preconditioner built up for the regularized problem for solving the case $\varepsilon = 0$. To support this idea, in Figure 6.4 we report the eigenvalues of the non-regularized Bingham matrix \mathcal{A} for the case of the Stokes-type equations computed for one of our test cases, the flow between parallel plates (see Section 5.2) with a number of degrees of freedom small enough to use MATLAB's `eig`, namely $h = 1/16$ in a 2D unit square. The Figure also displays the eigenvalues of \mathcal{A} when preconditioned by the regularized problem \mathcal{A}_ε , i.e. the eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ with $\varepsilon = 10^{-2}$. Clustering of the eigenvalues around $\lambda = 1$ is evident.

To quantify the impact of the regularization parameter on the non-regularized

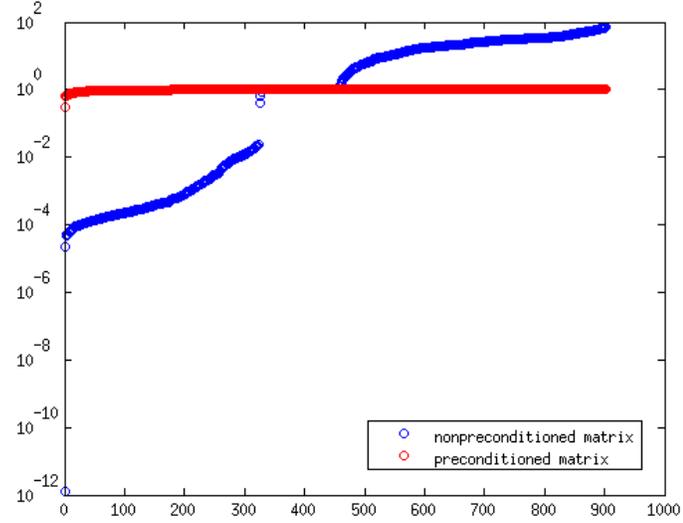


Figure 6.4: Absolute values of the eigenvalues of the discrete linearized Bingham matrix \mathcal{A} (blue) and eigenvalues of $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ (red) in the analytical test case, where \mathcal{A}_ε is the regularized Bingham matrix with $\varepsilon = 10^{-2}$.

problem, we define

$$\mathcal{S} = \begin{bmatrix} A & B^T \\ B & -\alpha Q \end{bmatrix}, \quad \mathcal{F} = \begin{bmatrix} B \\ C \end{bmatrix}$$

and consider the following factorization of the linear system matrix \mathcal{A}_ε with $\varepsilon > 0$:

$$\mathcal{A}_\varepsilon = \begin{bmatrix} \mathcal{S} & \mathcal{F}^T \\ \mathcal{F} & N_\varepsilon \end{bmatrix} = \begin{bmatrix} \mathcal{S} & 0 \\ \mathcal{F} & N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix}.$$

Then

$$\mathcal{A}_\varepsilon^{-1} = \begin{bmatrix} I & -\mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} \mathcal{S}^{-1} & 0 \\ (N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)^{-1}\mathcal{F}\mathcal{S}^{-1} & -(N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)^{-1} \end{bmatrix}$$

and by a direct computation we get

$$\mathcal{A}_\varepsilon^{-1}\mathcal{A} = \begin{bmatrix} I & -\mathcal{S}^{-1}\mathcal{F}^T \\ 0 & I \end{bmatrix} \begin{bmatrix} I & \mathcal{S}^{-1}\mathcal{F}^T \\ 0 & -\mathcal{X} \end{bmatrix} \quad (6.10)$$

with the matrix block

$$\mathcal{X} = (\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T) - N_\varepsilon^{-1}(\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T - N). \quad (6.11)$$

Note that \mathcal{S} represents the (Newtonian) Stokes or Navier-Stokes part of the linear system and the inverse \mathcal{S}^{-1} is well-defined provided either \mathbf{u} and p are discretized in *inf-sup* compatible spaces. We can see from (6.10) and (6.11) that the eigenvalues of the preconditioned matrix $\mathcal{A}_\varepsilon^{-1}\mathcal{A}$ cluster around one if the spectrum of $(N_\varepsilon - \mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T)$ is similar to the spectrum of $(\mathcal{F}\mathcal{S}^{-1}\mathcal{F}^T - N)$. Should the inverses be computed exactly, this trivially holds true for $\varepsilon \rightarrow 0$. Figure 6.5 shows the residual for the first 30 iterations of **GMRES** when solving the preconditioned system $\mathcal{A}_\varepsilon^{-1}\mathcal{A}\mathbf{x} = \mathcal{A}_\varepsilon^{-1}\mathbf{f}$ for different values of ε . To provide this proof of concept, a coarse grid is used (again unit square domain with $h = 1/16$) and the inverse of \mathcal{A}_ε is applied exactly using a direct method. The smaller ε , the faster the **GMRES** iterations reach any given tolerance.

As far as we know, there been a strict distinction between solvers for the regularized model and solvers for the non-regularized model in the literature on the numerical solution of Bingham fluids so far. To our knowledge, this type of combination of regularized and non-regularized model presented here has not been advocated before.

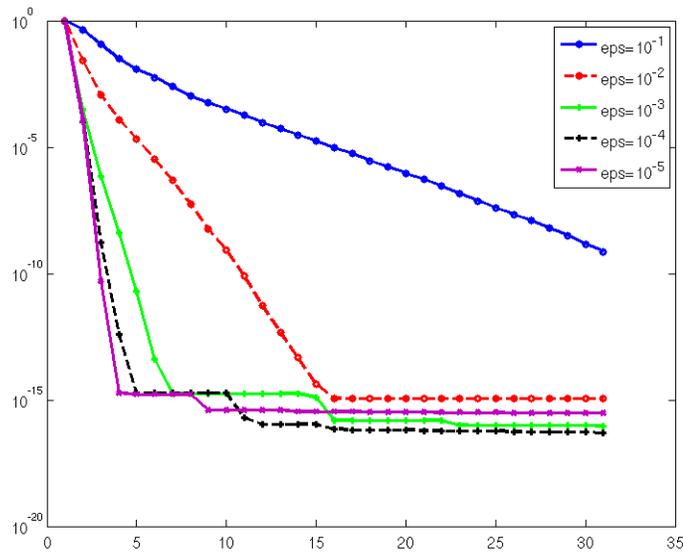


Figure 6.5: Residual of GMRES for the first 30 iterations for when solving the non-regularized problem preconditioned by the regularized one with different values of ε .

6.5 Inexact Factorization

Using the exact inverse of \mathcal{A}_ε , with $\varepsilon > 0$ as a preconditioner as done in Figure 6.4, is clearly not practical if the problem is large. The efficient solution of the linear system requires an approximation of the inverse which can be computed at a relatively low cost in terms of memory and CPU time and which significantly reduces the number of iterations of the linear iterative solver. Many options exist how to obtain such an approximation. We consider first inexact factorization methods for preconditioning the linear system. While we will not follow this approach in this thesis, we believe that these methods are worth mentioning here since they have been shown to be very efficient when applied to the unsteady Navier-Stokes equations for incompressible fluids [74, 77, 78] and their applicability to the Bingham flow (or Non-Newtonian fluid flow in general) should be investigated in the future. We begin by recalling two important types of inexact factorization methods applied to the unsteady Navier-

Stokes equations (for Newtonian fluids) before presenting some ideas how this may be used for the twofold saddle point problem (4.21) for Non-Newtonian fluids.

6.5.1 Yosida and Chorin-Temam Schemes

After linearization and discretization of the Navier-Stokes equations, the saddle point problem

$$\mathcal{A}_{NS} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}$$

is obtained. Note that the matrix blocks A and B are indeed the same as in (4.21).

The problem may be factorized as

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} A & 0 \\ B & -BA^{-1}B^T \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} \approx \begin{bmatrix} A & 0 \\ B & -BH_1B^T \end{bmatrix} \begin{bmatrix} I & H_2B^T \\ 0 & I \end{bmatrix}, \quad (6.12)$$

where H_1 and H_2 are two approximations of the inverse A^{-1} .

Note now that $A = \frac{1}{\Delta t}M + T$, where Δt is the time step, M a mass matrix in the discrete velocity space and T a discrete Laplacian. We have

$$A = \frac{1}{\Delta t}M + T = \frac{1}{\Delta t}M(I + \Delta tM^{-1}T),$$

hence

$$A^{-1} = \Delta t(I + \Delta tM^{-1}T)^{-1}M^{-1}.$$

Let us assume that $I + \Delta tM^{-1}T \approx I$. Then

$$A^{-1} = \Delta t(I + \Delta tM^{-1}T)^{-1} \approx \Delta tM^{-1}.$$

The resulting three-step method is the (algebraic) *Chorin-Temam scheme* [74], letting $H_1 = H_2 = \Delta tM^{-1}$ in (6.12):

1. Solve $A\hat{\mathbf{u}} = \mathbf{f}$ for the “intermediate velocity” $\hat{\mathbf{u}}$,
2. find the pressure p by solving $-\Delta t B M^{-1} B^T p = -B\hat{\mathbf{u}}$,
3. “correct” the velocity by solving $\Delta t M \mathbf{u} = \Delta t M \hat{\mathbf{u}} - \Delta t B^T p$ for \mathbf{u} .

Similarly, the *Yosida scheme* [77] is obtained by letting $H_1 = \Delta t M$ and $H_2 = A^{-1}$ in (6.12) and reads

1. $A\hat{\mathbf{u}} = \mathbf{f}$ (intermediate velocity computation),
2. $-\Delta t B M^{-1} B^T p = -B\hat{\mathbf{u}}$ (pressure computation),
3. $A\mathbf{u} = \mathbf{f} - \Delta t B^T p$ (final velocity computation).

6.5.2 Application to Bingham Fluids

Recall from Section 6.3 that there are two ways to interpret the saddle point structure of (4.21). We therefore start our considerations on a general factorization of a saddle point system of the form

$$\begin{bmatrix} P & L^T \\ L & J \end{bmatrix} = \begin{bmatrix} P & L^T \\ L & LP^{-1}L^T - SR \end{bmatrix},$$

with $S = LP^{-1}L^T$ and $R = S^{-1}(S + J) = (I + S^{-1}J)$. Efficient computations require good approximations \hat{S}^{-1} , \hat{R}^{-1} and \hat{P}^{-1} of S^{-1} , R^{-1} and P^{-1} respectively. This can be done in different ways and depends on the particular choice of P and L . The resulting inexact factorization reads

$$\begin{bmatrix} P & L^T \\ L & J \end{bmatrix} = \begin{bmatrix} P & L^T \\ L & LP^{-1}L^T - SR \end{bmatrix} \approx \begin{bmatrix} P & 0 \\ L & -\hat{S} \end{bmatrix} \begin{bmatrix} I & \hat{P}^{-1}L^T \\ 0 & \hat{R} \end{bmatrix}. \quad (6.13)$$

Possible approximations of $R^{-1} = (I - S^{-1}J)$ are of Neumann type [61], that is, under the assumption that the spectral radius of $S^{-1}J$ is less than one, $\rho(S^{-1}J) < 1$, we consider the Neumann series

$$(I - S^{-1}J)^{-1} = \sum_{i=1}^{\infty} (S^{-1}J)^i$$

and consequently obtain an approximation of the form

$$(I - S^{-1}J)^{-1} \approx \sum_{i=1}^q (S^{-1}J)^i$$

for some $q \in \mathbb{N}$. We may also use the Chaouet-Chabard type approximation [48]

$$(I + S^{-1}L)^{-1} \approx I + (S^{-1}J)^{-1},$$

provided J is non-singular. The idea is to use the inexact factorization (6.12) to sequentially compute \mathbf{u} , p and W by interpreting the saddle point problem by either (6.8) or (6.9). In the latter case, P represents the Navier-Stokes equations (for a Newtonian fluid) and \hat{P}^{-1} may be obtained by using Chorin-Temam or Yosida type schemes.

Remark Recall that we may use the regularized problem as a preconditioner for the non-regularized one. Hence, when considering inexact factorization methods, we may assume that the matrix block N in (4.21) symmetric positive definite.

6.6 The Multilevel Preconditioner

While inexact factorization methods are a promising technique for approximating the inverse of the regularized Bingham system $\mathcal{A}_\varepsilon^{-1}$, in this thesis we focus on a geometric multigrid method. Multigrid methods have experienced an increasing popularity for a

large range of problems, including the solution of indefinite problems (see, e.g. [87,102] in the context of constrained optimization problems and fluid-structure interaction, respectively). Let $\{\mathcal{A}_k\}_{k=1}^L$ be the linear system matrices representing the discretization of (4.21) on a sequence of L meshes, \mathcal{A}_{k-1} being on a coarser mesh than \mathcal{A}_k . Let further $\{P_k\}_{k=1}^{L-1}$ be the natural interpolation matrices relating variationally the system matrix \mathcal{A}_k to its coarser counterpart \mathcal{A}_{k-1} , and let \mathbf{y} be the generic input vector, \mathbf{x} the corresponding output vector and σ , ν , ℓ and tol be given constants, where ℓ represents the current level and tol the given tolerance. We propose the following recursive preconditioner:

Algorithm 6.5 The Multilevel Algorithm

MLPrecond(\mathbf{x} , \mathbf{y} , $\{\mathcal{A}_k\}_k$, $\{P_k\}_k$, σ , ν , ℓ , tol):

smooth ν times on $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$;

restrict residual: $\mathbf{r} = P_{\ell-1}^T(\mathbf{y} - \mathcal{A}_\ell \mathbf{x})$;

if $\ell - 1 = 1$ **then**

$\mathbf{x}_c = (\mathcal{A}_{\ell-1}) \setminus \mathbf{r}$; //coarsest level: Matlab notation for a direct method

else

$\mathbf{x}_c = \mathbf{0}$;

$Precond = @MLPrecond(\mathbf{x}_c, \mathbf{r}, \{\mathcal{A}_k\}_{k=1}^{\ell-1}, \{P_k\}_{k=1}^{\ell-1}, \sigma, \nu, \ell - 1, tol)$;

$FGMRES(\mathcal{A}_{\ell-1}, \mathbf{x}_c, \mathbf{r}, tol, \sigma, Precond)$;

end if

update $\mathbf{x} = \mathbf{x} + P_{\ell-1} \mathbf{x}_c$;

smooth ν times on $\mathcal{A}_\ell \mathbf{x} = \mathbf{y}$;

Algorithm 6.5 is a multigrid W-cycle; the method “MLPrecond” recursively calls itself σ times as a preconditioner inside an FGMRES scheme [81]. Note that we use in the smoother a Krylov subspace method, which will also contribute towards the nonlinear nature of the proposed preconditioner.

6.6.1 Interpolation and Restriction

Starting with a mesh that is sufficiently coarse to allow a fast solution of the discrete system (e.g. with a direct solver that we have denoted with “\”), we refine the mesh uniformly L times. With each mesh, we associate a corresponding triple of finite element spaces, $\mathbf{V}_k, Q_k, \mathcal{Z}_k, k = 1, \dots, L$. By construction, the coarse level spaces are subspaces of the next fine level spaces. This defines natural embeddings $\{P_k^{\mathbf{u}}\}_{k=1}^L$, $\{P_k^p\}_{k=1}^L$ and $\{P_k^W\}_{k=1}^L$ which transfer (interpolate) the degrees of freedom of \mathbf{u}, p and W , respectively from the coarse mesh to the fine mesh. The (monolithic) interpolation operator is then given by

$$P_k = \begin{bmatrix} P_k^{\mathbf{u}} & 0 & 0 \\ 0 & P_k^p & 0 \\ 0 & 0 & P_k^W \end{bmatrix}.$$

The matrix \mathcal{A}_L is assembled on the finest mesh, and the coarse ones are variationally related via the Galerkin condition $\mathcal{A}_{k-1} = P_k^T \mathcal{A}_k P_k$ for $k = 1, \dots, L$.

6.6.2 Smoothing

Several types of smoothers may be considered. A classical approach is to perform a few iterations of a preconditioned iterative method. This has been done in [?, 97] for an s.p.d. problem, with a fixed number of conjugate gradient iterations used for smoothing. The iterations can involve the preconditioner defined recursively on previous coarse levels. This nonlinear preconditioner is used to accelerate the convergence of a flexible Krylov subspace method.

We will follow a similar idea on the indefinite system (4.21). An efficient smoother is given by the overlapping additive Schwarz method. For the sake of simplicity, we will omit the index k indicating the level of discretization for the remainder of this section. Given the discretized domain Ω on any given level, we may subdivide the domain into m overlapping subsets $\{\Omega_i\}_{i=1}^m$. Then we set up linear mappings $\{I_i^{\mathbf{u}}\}_{i=1}^m$,

$\{I_i^p\}_{i=1}^m$ and $\{I_i^W\}_{i=1}^m$ restricting the degrees of freedom of \mathbf{u} , p and W , respectively, to the local domain Ω_i . The discrete local matrix is then given by

$$\mathcal{A}_i = I_i \mathcal{A} I_i^T \text{ with } I_i = \begin{bmatrix} I_i^{\mathbf{u}} & 0 & 0 \\ 0 & I_i^p & 0 \\ 0 & 0 & I_i^W \end{bmatrix}$$

and the inverse of the global matrix M is approximated by the formula

$$\mathcal{A}^{-1} \approx \sum_{i=1}^m I_i^T \mathcal{A}_i^{-1} I_i.$$

The size of the subdomains should be chosen sufficiently small so that the inverse of the local matrices \mathcal{A}_i can be computed quickly.

In the next chapter we will discuss in detail the numerical performance of the preconditioner presented here.

Chapter 7

Numerical Results ¹

This chapter comprises several numerical experiments that demonstrate the effectiveness of the multilevel preconditioner from Chapter 6 for solving linear system arising from the discrete linearized Bingham equations. We consider both two dimensional and three dimensional test cases, the latter being not affordable with the direct solver used in Chapter 5. We start with explaining the experimental setup, with details on how the parameters for the preconditioners are chosen. In Section 7.2 we present numerical results on benchmark problems that have been introduced in Chapter 5. We then proceed to problems from hemodynamics in Section 7.3.

7.1 Experimental Setup

Unless stated otherwise, the computational domain is a unit square or a unit cube. We discretize in space with P2-P1 finite elements for velocity and pressure. The auxiliary variable W is discretized with P1 finite elements. Again, we choose an inf-sup stable pair for velocity and pressure and the choice for W was made as an appropriate trade-off between accuracy and efficiency. We refer to the C++ finite

¹The work presented in this chapter was performed in close collaboration with Panayot Vassilevski, Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA.

element library MFEM [63]. For the coarsest grid we choose $h = 1/4$ in the two dimensional case and $h = 1/2$ in three dimensions. As a solver on the coarsest grid we use a direct solver within the C library SUITESPARSE. More precisely, the coarse level matrix is factorized into an LDL^T factorization (LDL package [29]) for the Stokes type equations (1.4) and into an LU factorization (UMFPACK package [27,28,31,32]) for the Navier-Stokes type problem (1.2). Before computing all factorizations we apply a fill-in reducing reordering provided by AMD [2,3,30].

To set up the smoother on each level (except for the coarsest), we first generate an adjacency matrix $S = [s_{ij}]$ (with $s_{ij} = 1$ if element i and j share a common face in three dimensions or a common edge in two dimensions and $s_{ij} = 0$ otherwise). We then apply a graph partitioner in METIS [58] on S . This procedure results in a partitioning of the mesh in which the overlap consists of one layer of elements at the interface. Extra layers of overlap may be included as well. The solves on each subdomain is again done by the direct solvers provided in SUITESPARSE. Table 7.1 shows the different meshes we use for our experiments and the number of multigrid levels used for each mesh. Also, the number of subdomains is shown. The number of overlapping nodes is specified as well. The number of subdomains on each level has a strong influence on the performance of our preconditioner. The trade-off is between the size of the local system (not too large) and the overall efficacy of the smoother. This is achieved by increasing the number of subdomains by a factor of 4 in 2D and a factor of 6 in 3D for each additional multigrid level, as shown in the table. The size of the discrete system is shown as well. To produce the results in the following subsections, we start the nonlinear Picard iterations with the initial guess $\mathbf{u} = \mathbf{u}^0$, $p \equiv 0$, $W \equiv 0$ where \mathbf{u}^0 is the solution of $-\mu\Delta\mathbf{u}^0 = \mathbf{f}$ solved with preconditioned CG iterations. We continue the nonlinear iterations until

$$\frac{\|\mathbf{r}\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-2},$$

| Experiments on Unit Square | | | | | | |
|----------------------------|----------|---------|-----------------|-------------|-----------|---------|
| mesh | # levels | # subd. | #overlap. nodes | size overl. | size lin. | syst. |
| $h = 1/8$ | 2 | 3 | 34-40 | 27 | | 902 |
| $h = 1/16$ | 3 | 9 | 42-50 | 126 | | 3,334 |
| $h = 1/32$ | 4 | 27 | 56-72 | 498 | | 12,806 |
| $h = 1/64$ | 5 | 81 | 68-85 | 1,839 | | 50,182 |
| $h = 1/128$ | 6 | 243 | 89-110 | 6,751 | | 198,662 |
| $h = 1/256$ | 7 | 729 | 114-143 | 24,343 | | 790,534 |

| Experiments on Unit Cube | | | | | |
|--------------------------|----------|---------|------------|-------------|-----------------|
| mesh | # levels | # subd. | size subd. | size overl. | size lin. syst. |
| $h = 1/4$ | 2 | 12 | 24-31 | 97 | 3,062 |
| $h = 1/8$ | 3 | 72 | 27-49 | 669 | 19,842 |
| $h = 1/16$ | 4 | 432 | 34-55 | 4,697 | 142,202 |
| $h = 1/32$ | 5 | 2,392 | 41-70 | 34,925 | 1,075,434 |

Table 7.1: Number of levels of multigrid, number of subdomains, size of each subdomain, size of overlap and the size of the linear system to be solved in two and three dimensions.

where \mathbf{r} (\mathbf{r}_0) is the current (initial) residual. We set the absolute tolerance to $5 \cdot 10^{-6}$. By choosing this nonlinear stopping criterion we make sure that the linear solver is accurate enough to achieve nonlinear convergence. The linear system is solved by FGMRES with geometric multigrid preconditioner and is considered converged if the quotient of current and initial residual drops below 10^{-6} in the L^2 -norm,

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq 10^{-6}.$$

All tables display the number of linear iterations needed for convergence of the first nonlinear iteration. Table 7.2 shows the linear and nonlinear tolerances we use for our experiments.

7.1.1 Choosing the Regularization Parameter

In Section 6.4 we stated that the performance of the preconditioner \mathcal{A}_ε improves as ε decreases, provided that the inverse $\mathcal{A}_\varepsilon^{-1}$ is computed exactly. However, the

| linear tol. | nonlinear tol. | nonlinear tol. (absolute) | # subdomains |
|-------------|----------------|---------------------------|--------------|
| 10^{-6} | 10^{-2} | $5 \cdot 10^{-6}$ | d^ℓ |

Table 7.2: Tolerances and number of subdomains used in our experiments: relative linear tolerance, nonlinear tolerance, absolute nonlinear tolerance (all measured in the L^2 -norm), number of subdomains used (d is the spacial dimension and ℓ the number of multigrid levels).

reduction of the regularization parameter in general deteriorates the conditioning properties of the matrix and this may impair the quality of the approximation. In this respect, finding the optimal value of ε involves finding the right trade-off between numerical stability and approximating the physical problem to be solved. In our experiments we empirically found the optimal choice to be $\varepsilon = 10^{-2}$. However, a rigorous analysis justifying this choice is still missing. It is worth noticing that the domain decomposition used in our experiments is based entirely on the mesh and not on the solution. If a subdomain is entirely contained in a plug region, we may experience some performance degradation. As a matter of fact, the local representation of the linear system is extremely ill-conditioned for small values of ε and the local (direct) solves may be very inaccurate, resulting in failure of the smoother. Larger values of ε yield an improved conditioning of the local system and local solves are more accurate.

We also noticed that the condition number of the regularized block N_ε in (4.21) grows mildly as $\varepsilon \rightarrow 0$ except when between 10^{-2} and 10^{-3} where the increase is more evident (see Figure 7.1).

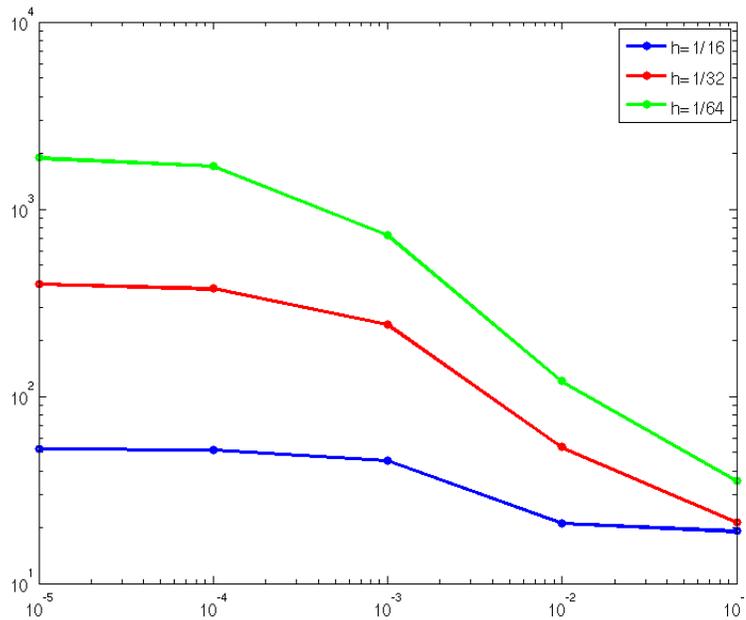


Figure 7.1: Condition number of the block N_ε for different values of ε and h .

7.2 Benchmark Problems

7.2.1 The Flow Between Two Parallel Plates

We test the linear solver on the analytical test case from Chapter 5. To precondition the FGMRES iterations we use Algorithm 6.5 with two smoothing steps ($\nu = 2$) in two dimensions and four smoothing steps ($\nu = 4$) in three dimensions as well as two iterations of FGMRES ($\sigma = 2$) on each multigrid level. Table 7.3 displays the number of FGMRES iterations needed for convergence for the first Picard step, the total number of nonlinear iterations needed as well as the CPU time needed for solving the linear system. In the three dimensional case the number of iterations increases slightly with growing mesh size. However, the parameters for the preconditioner were chosen to minimize the CPU time as opposed to the iteration count. Mesh independent convergence may be achieved by either increasing the number of smoothings or by decreasing the number of subdomains. Note that when solving the Stokes type

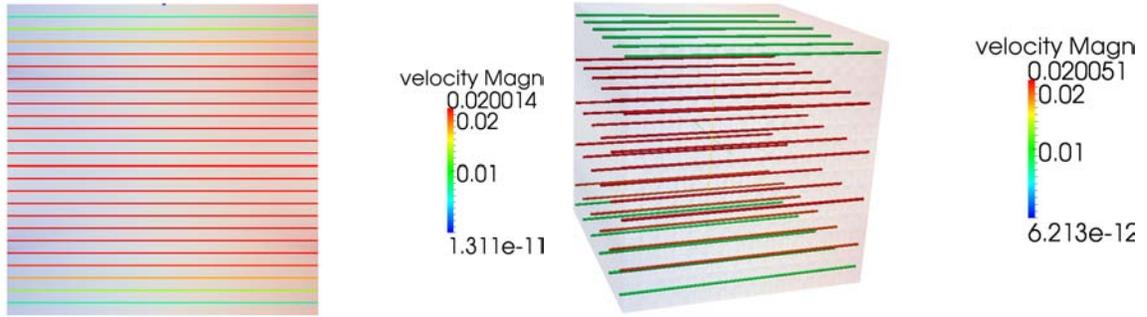


Figure 7.2: Streamlines and pressure for the two and three dimensional flow between two parallel plates. The pressure field is indicated by the background color, the streamlines are colored by the magnitude of the velocity field.

problem, only the matrix N in (4.21) needs to be updated before each iteration. In this respect, timings provided in Table 7.3 are divided into initial setup time (this includes setting up the interpolation matrices between the multigrid levels, determining the subdivisions of the domains and setting up the restriction matrices for each subdomain) and updating time (this includes the updating of the matrix N in the preconditioner, computing a factorization of the local monolithic system on each subdomain and computing a factorization for the direct solve on the coarsest level). All experiments are implemented in serial code. Timings for the Stokes type problem are obtained on a serial machine with an Intel Core i7 processor (2.6 GHz) and 8 GB of memory. Timings in Table 7.3 scale as expected except in the three dimensional case between $h = \frac{1}{16}$ and $h = \frac{1}{32}$, where the updating time increases significantly. This may be due to the high memory requirement of the fine mesh, causing memory-hard disc communication during the computation. Therefore, in Table 7.3 we also include timings for the three dimensional case obtained on a different machine, a Sun Microsystems SunFire X4600 with 20 AMD Opteron (TM) cores and 32 GB of memory. Timings for the fine meshes show improved scaling properties on this machine. Figure 7.2 shows the streamlines and pressure of this flow in two and three dimensions.

| Two Dimensional Experiments | | | | | |
|-----------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/8$ | 10 | 0.01 | 0.01 | 0.01 | 6 |
| $h = 1/16$ | 13 | 0.08 | 0.02 | 0.03 | 6 |
| $h = 1/32$ | 14 | 0.50 | 0.05 | 0.09 | 6 |
| $h = 1/64$ | 14 | 2.19 | 0.18 | 0.40 | 6 |
| $h = 1/128$ | 14 | 9.60 | 0.86 | 1.72 | 7 |
| $h = 1/256$ | 12 | 37.93 | 5.44 | 7.25 | 7 |

| Three Dimensional Experiments | | | | | |
|-------------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/4$ | 6 | 0.08 | 0.01 | 0.07 | 4 |
| $h = 1/8$ | 8 | 1.51 | 0.17 | 0.75 | 5 |
| $h = 1/16$ | 16 | 27.89 | 1.81 | 6.74 | 6 |
| $h = 1/32$ | 11 | 258.53 | 25.53 | 90.46 | 6 |

| Three Dimensional Experiments (*) | | | | | |
|-----------------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/4$ | 6 | 0.34 | 0.01 | 0.20 | 4 |
| $h = 1/8$ | 8 | 7.74 | 0.94 | 2.36 | 5 |
| $h = 1/16$ | 16 | 141.54 | 6.85 | 22.21 | 6 |
| $h = 1/32$ | 11 | 927.99 | 86.75 | 269.61 | 6 |

Table 7.3: The flow between two parallel plates, an analytical test case in two and three dimensions: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations. Three dimensional results in the third table are (marked by (*)) are obtained on a Sun Microsystems SunFire X4600.

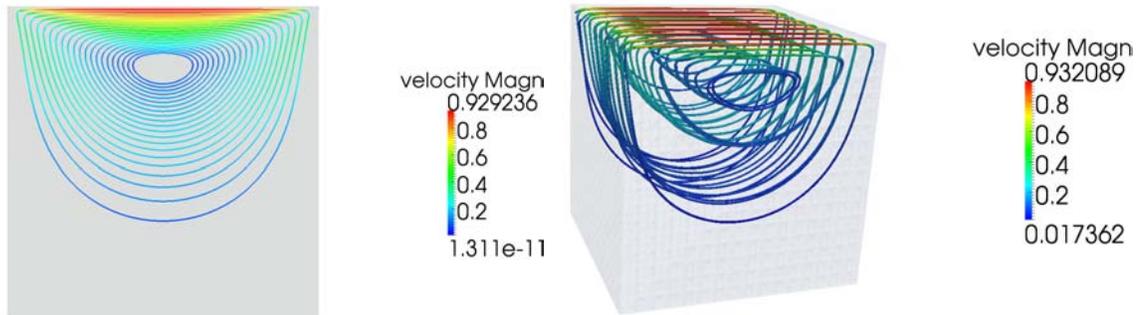


Figure 7.3: Streamlines and pressure field of the lid-driven cavity in two (left) and three (right) dimensions.

7.2.2 The Lid-Driven Cavity

We now test the linear solver on the lid-driven cavity benchmark introduced in Chapter 5. For the preconditioner we use Algorithm 6.5 with two smoothings in two dimensions and four smoothings in three dimensions as well as two inner iterations of FGMRES on each multigrid level. Table 7.4 shows the numerical results for this experiment. Results in the table indicate mesh independent convergence of the linear and nonlinear solver. Also, the total solution time increases by a factor of approximately 5 in two dimensions and approximately 10 in three dimensions (except between $h = \frac{1}{16}$ and $h = \frac{1}{32}$) for each additional multigrid level, while the corresponding size of the linear system increases by a factor of 4 and 8, respectively. Again we obtain improved scaling of our timings on a machine with more memory, displayed in Table 7.4. Streamlines and pressure distribution in two and three dimensions are shown in Figure 7.3.

7.2.3 The Steady Navier-Stokes type problem

In this subsection we apply the lid-driven cavity test case to the steady Navier-Stokes type problem. More precisely, we solve (2.4) with $\rho = 1$ and $\frac{\partial \mathbf{u}}{\partial t} \equiv 0$. Here all

| Two Dimensional Experiments | | | | | |
|-----------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/8$ | 8 | 0.01 | 0.01 | 0.01 | 4 |
| $h = 1/16$ | 11 | 0.07 | 0.01 | 0.02 | 5 |
| $h = 1/32$ | 12 | 0.42 | 0.04 | 0.10 | 5 |
| $h = 1/64$ | 12 | 1.87 | 0.17 | 0.40 | 5 |
| $h = 1/128$ | 12 | 8.22 | 0.80 | 1.71 | 5 |
| $h = 1/256$ | 11 | 34.91 | 5.44 | 7.29 | 4 |

| Three Dimensional Experiments | | | | | |
|-------------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/4$ | 3 | 0.04 | 0.01 | 0.06 | 3 |
| $h = 1/8$ | 5 | 0.94 | 0.17 | 0.74 | 4 |
| $h = 1/16$ | 8 | 13.94 | 1.80 | 6.63 | 5 |
| $h = 1/32$ | 5 | 167.21 | 25.48 | 89.12 | 5 |

| Three Dimensional Experiments (*) | | | | | |
|-----------------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/4$ | 3 | 0.17 | 0.01 | 0.21 | 3 |
| $h = 1/8$ | 5 | 4.63 | 0.49 | 2.37 | 4 |
| $h = 1/16$ | 8 | 71.30 | 6.69 | 21.66 | 5 |
| $h = 1/32$ | 5 | 419.88 | 86.94 | 200.62 | 5 |

Table 7.4: The lid-driven cavity flow in two and three dimensions: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations. The third table (marked by (*)) displays timings obtained on a Sun Microsystems SunFire X4600.

| Two Dimensional Experiments | | | | | |
|-----------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/8$ | 7 | 0.03 | 0.01 | 0.04 | 9 |
| $h = 1/16$ | 10 | 0.41 | 0.01 | 0.19 | 8 |
| $h = 1/32$ | 12 | 2.50 | 0.04 | 0.77 | 7 |
| $h = 1/64$ | 12 | 11.74 | 0.23 | 3.27 | 6 |
| $h = 1/128$ | 12 | 54.00 | 1.44 | 13.39 | 5 |
| $h = 1/256$ | 11 | 232.63 | 11.87 | 53.32 | 4 |

| Three Dimensional Experiments | | | | | |
|-------------------------------|------------|--------------|-----------|--------------|----------------|
| mesh | # lin. its | CPU time (s) | setup (s) | updating (s) | # nonlin. its. |
| $h = 1/4$ | 6 | 0.59 | 0.02 | 0.61 | 12 |
| $h = 1/8$ | 6 | 9.28 | 0.12 | 7.33 | 17 |
| $h = 1/16$ | 8 | 101.76 | 1.77 | 58.29 | 15 |
| $h = 1/32$ | 7 | 747.84 | 47.53 | 440.48 | 4 |

Table 7.5: The lid-driven cavity flow for the Navier-Stokes type problem: Number of linear iterations, CPU time for solving the linear system, setup and updating time for the preconditioner (all in seconds) and the total number of Picard iterations.

specifications are the same as in the previous subsection except that we now impose

$$\mathbf{u} = \begin{bmatrix} 50 \\ 0 \\ 0 \end{bmatrix} \text{ if } y = 1$$

corresponding to a Reynolds number of $Re = 50$. We now tighten the nonlinear tolerance to 10^{-4} to ensure accurate solutions. Numerical results are shown in Table 7.5. Due to a high memory requirement for solving the Navier-Stokes type equations, we perform the three dimensional experiments only on the Sun Microsystems Sun-Fire X4600. Again we observe mesh independent convergence of the linear solver, the number of nonlinear iterations even decreases with growing mesh size. Plots of streamlines and pressure in Figure 7.4.

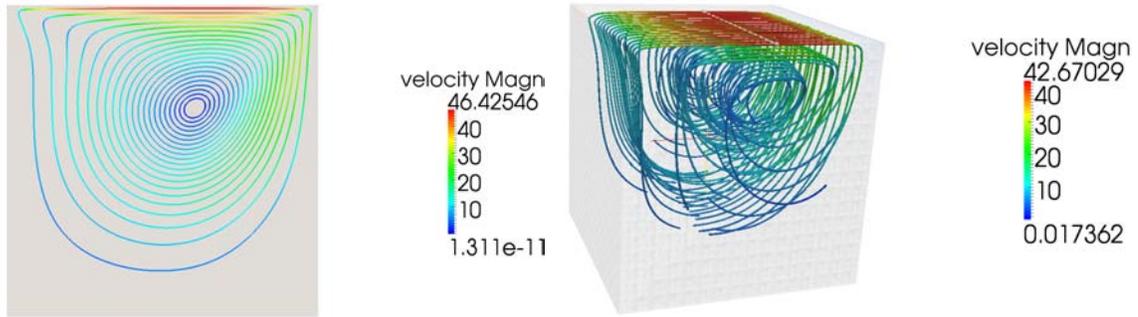


Figure 7.4: Velocity and pressure field of the lid-driven cavity on the Navier-Stokes type problem.

7.3 Applications in Hemodynamics

7.3.1 Motivation

In this section we discuss some potential applications of the research presented here to real-life problems involving blood flow simulations. In particular, we are investigating the applicability of Bingham fluids – or other types of visco-plastic fluids – to blood flow through an aneurysm.

An *aneurysm* is a gradual expansion of a part of an artery, occurring over a prolonged period of time. Two types of aneurysms exist [43]:

1. *fusiform aneurysms* are dilations of the cylindrical diameter of the vessel, and
2. *saccular aneurysms* are ball-like bulges on one side of the cylindrical vessel.

The affected part of the arterial wall stretches and becomes thinner and weaker. As a consequence, untreated aneurysms are prone to rupture, leading to a possible severe blood loss. In particular, a rupture of *cerebral aneurysms* – aneurysms occurring in the brain – may be fatal. The formation of cerebral aneurysms is considered to be caused by a complex interplay including biomechanical properties of the vessel wall and the forces acting on the wall, originating from the flow of blood through the vessel [25, 68, 89, 96]. Cerebral aneurysms tend to occur in regions with particular

features, such as bifurcations and curves, supporting the hypothesis that blood flow has a major involvement in their formation [72]. The purpose of the present research involving simulations of the flow of blood through arteries is to develop a thorough understanding of the nature of this involvement. These results could help anticipating the probability of a rupture as well as developing possible treatments of non-ruptured aneurysms [93]. It has been shown that the rupture of cerebral aneurysms is closely correlated to certain geometrical properties of vessel, such as its curvature, radius and its wall-shear stress [72].

Figure 7.5, left, shows a blood vessel with a cerebral aneurysm. The image in Figure 7.5 is obtained when injecting a contrast material into the vessel and tracing it after it has distributed through the blood stream. Notice that only a part of the domain is captured through this procedure, the material does not reach the center and the area around the boundary of the saccular aneurysm. A possible explanation of this phenomenon could be that the blood flow in this area exhibits visco-plastic effects, that is, plug regions exist in parts of the vessel which are not reachable by the contrast liquid. This observation motivated the idea of considering the Bingham model for simulating the blood flow in this situation. Similarly, the Casson and Herschel-Bulkley model from Chapter 1 could be considered. The applicability of these models for the flow of blood through an aneurysm still needs to be fully understood.

7.3.2 A Simulation on an Idealized Geometry

Due to the complexity of geometries involving hemodynamics and cardiovascular modelling, numerical simulations are often performed on idealized geometries. For this simulation we use a significantly simplified version of the domain in Figure 7.5. Here we apply the unsteady Navier-Stokes type Bingham problem on a cylindrical domain with a sphere attached to it. The set can be specified as $D = \{(x, y, z) \in \mathbb{R}^3 | 0 \leq x \leq 10, y^2 + z^2 \leq 1\} \cup \{(x, y, z) \in \mathbb{R}^3 | x^2 + (y + \frac{3}{2})^2 + (z - 5)^2 \leq 1\}$, as a time interval

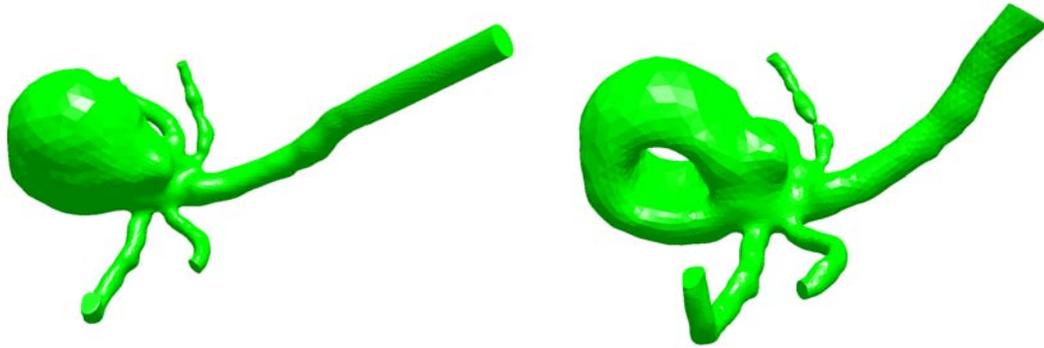


Figure 7.5: Blood vessel with aneurysm. Left: the original domain. Right: domain obtained when tracing the injected contrast material. Images by Frank Tong, Emory School of Medicine, Emory University.

we choose $I = [0, 1.5]$. In the Bingham fluid flow equations, we choose $\mu = 1$ and $\tau_s = 1$. A major challenge associated with using geometric multigrid preconditioning described in Section 6.6 with more complex shaped geometries is to represent the coarsest mesh accurately. We handle this by discretizing the domain with second order isoparametric finite elements. Using elements of higher order has the effect that the “curved” shape of the domain is captured well during the refinement process of the geometric multigrid. Figure 7.6 shows the shape of the idealized geometry on three levels of geometric multigrid. Note that the approximation of the domain improves with each additional level of multigrid.

We use multigrid preconditioning on three levels in this experiment with four smoothing steps and two inner FGMRES iterations. In the Bingham fluid equations we take $\mu = 1$ and $\tau_s = 1$. On the inflow, we prescribe a parabolic Dirichlet boundary condition by

$$u_1 = \begin{cases} -60t(y^2 + z^2), & \text{if } 0 \leq t \leq \frac{1}{2}, \\ -30(y^2 + z^2), & \text{if } \frac{1}{2} < t \leq \frac{3}{2}. \end{cases}, \quad u_2 \equiv u_3 \equiv 0.$$

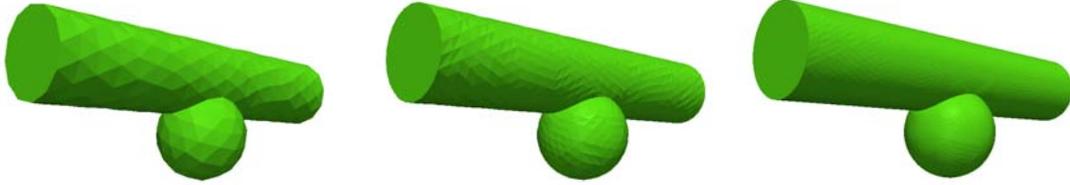


Figure 7.6: Idealized blood vessel with aneurysm on three geometric multigrid levels. Left: The coarsest level (280 elements), center: one level of refinement (2,240 elements), right: two levels of refinement (17,920 elements).

| Fine level | | Intermediate level | | General Information | |
|--------------|---------|--------------------|-------|------------------------------|-------|
| # subd.: | 72 | # subd.: | 12 | # Picard its (per time st.): | 5 |
| size subd.: | 100-140 | size subd.: | 74-98 | CPU (s) (per t.s.): | 53.41 |
| size overl.: | 2,791 | size overl.: | 342 | setup (s): | 0.80 |
| | | | | updating: | 72.36 |
| | | | | # linear its: | 6 |

Table 7.6: Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment.

These inflow boundary conditions do not change with time in the last two thirds of the time interval. This choice ensures that the solution reaches a steady-state at the end of the time interval. On the walls, we prescribe a no-slip condition and assume homogeneous Neumann boundary conditions on the outflow. The time is discretized with a time step of $\Delta t = 0.1$. Table 7.6 shows more details of how the parameters for this preconditioner are chosen as well as the numerical results for this experiment. Note that also in this “non-academic” test case with a more complex geometry the solver converges within a small number of iterations. Figure 7.7 shows the streamlines and pressure of this flow after a steady state has been reached. The Reynolds number is chosen sufficiently high so that a recirculation occurs in the sphere below the cylinder, as can be seen in the figure.

7.3.3 Simulations on a Real Aneurysms

We now test our solver on domains originating from real arteries. The images we use in this subsection are obtained from the AneuriskWeb page at Emory University

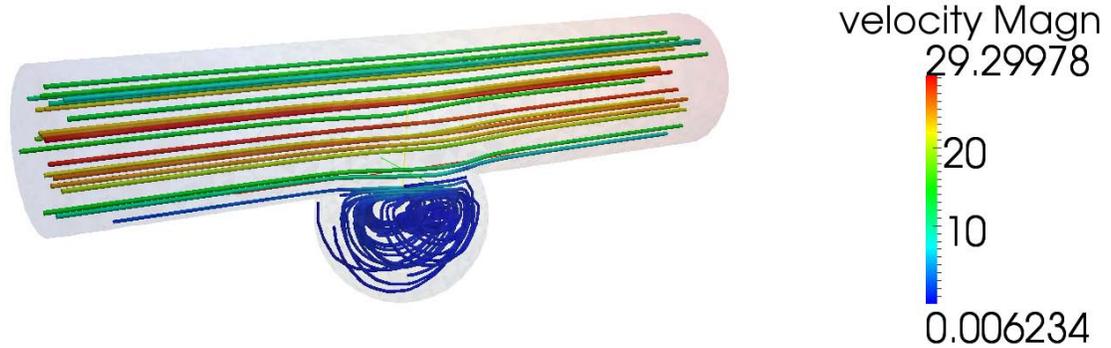


Figure 7.7: Streamlines of the unsteady Navier-Stokes type problem in a cylindrical domain with an attached sphere.

(see [4]) and originate from DICOM [36]. The surface of the geometries are generated via the `vmtk` software (see e.g. [75]). Based on the surface, we generate a mesh using `NETGEN` [86], which is then used to perform our computations in `MFEM`. The visualization of our solutions is done in `ParaView` [91].

We present two simulations in this subsection. In the first one, the domain is a part of a vessel having an aneurysm. Again we consider the time interval $I = [0, 1.5]$ and take the Dirichlet inflow boundary condition

$$u_1 = \begin{cases} 40t \cdot \boldsymbol{\eta}, & \text{for } 0 \leq t \leq \frac{1}{2}, \\ 20 \cdot \boldsymbol{\eta} & \text{for } \frac{1}{2} < t \leq \frac{3}{2}, \end{cases}, \quad u_2 \equiv 0,$$

where $\boldsymbol{\eta}$ is the (inward) unit normal to the surface of the inflow. Again we choose a time-dependent inflow boundary condition for the first third of the time interval and a constant boundary condition for the rest of the interval, so that the solution reaches a steady-state before $t = 1.5$. On the outflow, we assume homogeneous Neumann conditions. This choice of boundary conditions is extremely simplified, with no intention of mimicking a realistic situation where the inflow boundary condition is determined by a heart beat. Instead, we wish to get some understanding of how

| Fine level | | Intermediate level | | General Information | |
|--------------|--------|--------------------|--------|------------------------------|--------|
| # subd.: | 288 | # subd.: | 48 | # Picard its (per time st.): | 5 |
| size subd.: | 94-135 | size subd.: | 80-105 | CPU (s) (per t.s.): | 284.80 |
| size overl.: | 10,771 | size overl.: | 1396 | setup (s): | 4.91 |
| | | | | updating: | 230.96 |
| | | | | # linear its: | 7 |

Table 7.7: Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment on a real blood vessel with aneurysm.

Bingham fluids behave in this domain and to test the performance of our solver in complex geometries. We choose $\mu = 1$ and $\tau_s = 1$ in the Bingham equations for simplicity. As in the previous test case we use three levels of geometric multigrid and discretize the domain with second order isoparametric finite elements to capture the details of the geometry during the mesh refinement. The time discretization is $\Delta t = 0.1$ and we use five Picard iterations per time step. Table 7.7 displays details on the choice of parameters for the solver as well as numerical results of this experiment. Note that again the solver is convergent within a small number of iterations. Figure 7.8 shows a streamline plot of the solution after reaching a steady state, indicating a recirculation of the fluid in the area of the aneurysm.

In the second experiment, the computational domain is the cerebral aneurysm from Figure 7.5, left. This experiment is challenging due to the very complex shape of the geometry. Once again we use second order isoparametric elements for the discretization and two levels of geometric multigrid. Because of the complexity of the geometry, the coarsest grid has a relatively high number of elements, as a consequence the direct solve on the coarsest mesh is larger than in the previous experiments. The time interval is $I = [0, 1.5]$ discretized with $\Delta t = 0.1$. The Dirichlet inflow boundary condition is

$$u_1 = \begin{cases} 240t \cdot \boldsymbol{\eta}, & \text{for } 0 \leq t \leq \frac{1}{2}, \\ 120 \cdot \boldsymbol{\eta} & \text{for } \frac{1}{2} < t \leq \frac{3}{2}, \end{cases}, \quad u_2 \equiv 0,$$

with a unit normal $\boldsymbol{\eta}$ to the inflow surface. We prescribe a no-slip condition on the wall

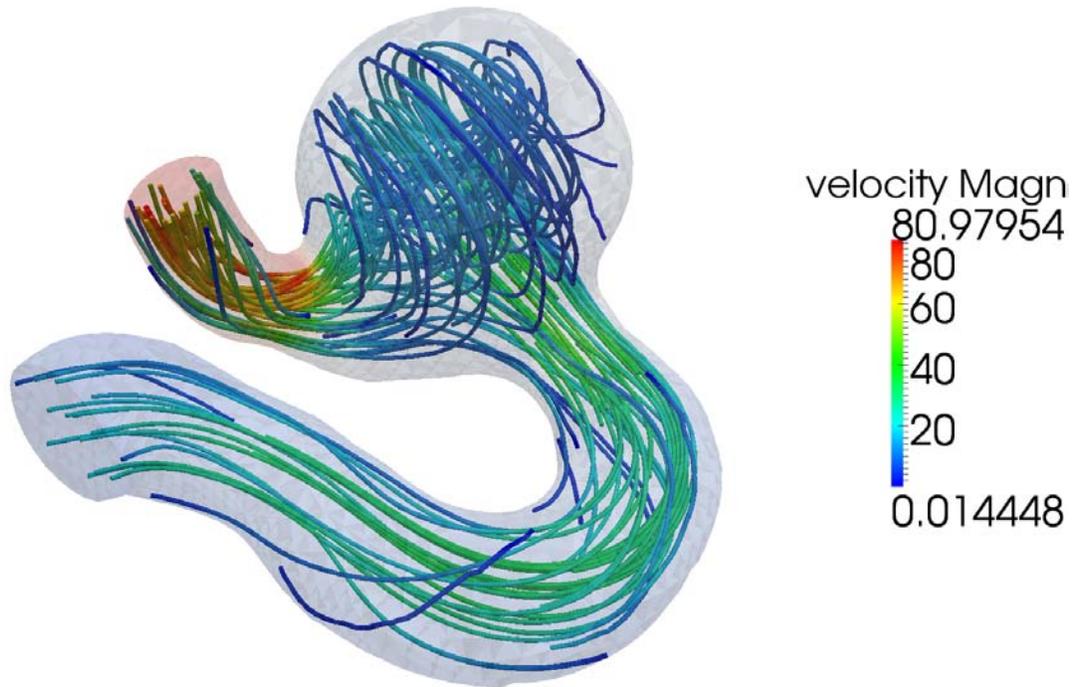


Figure 7.8: Streamlines of a Bingham fluid flow through a blood vessel with an aneurysm.

and homogeneous Neumann boundary conditions on the four outflow boundaries. At each time step we iterate with four Picard iterations. Table 7.8 shows more detailed specifications of the preconditioner as well as numerical results for this experiment. We obtain again convergence in a small number of iterations. Streamlines of the solution once the steady state has been reached are shown in Figure 7.9. The figure indicates a recirculation of the flow inside the aneurysm, the flow velocity is small inside the bulge.

| Fine level | | General Information | |
|--------------|---------|------------------------------|--------|
| # subd.: | 48 | # Picard its (per time st.): | 5 |
| size subd.: | 709-932 | CPU (s) (per t.s.): | 154.39 |
| size overl.: | 981 | setup (s): | 2.69 |
| | | updating: | 321.15 |
| | | # linear its: | 6 |

Table 7.8: Numerical results and specifications of the preconditioner for the unsteady Navier-Stokes experiment on the geometry from Figure 7.5, left.

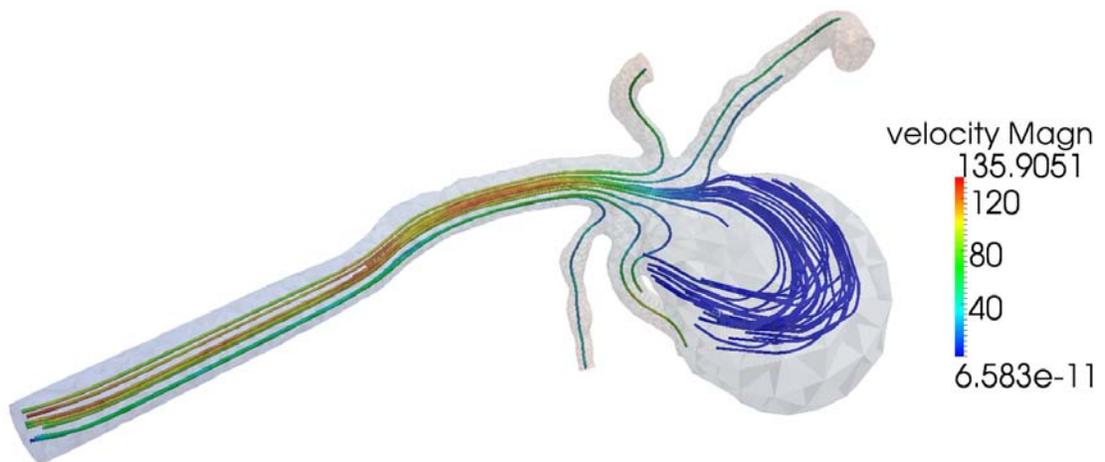


Figure 7.9: Streamlines of a Bingham fluid flow through the geometry from Figure 7.5, left.

Chapter 8

Conclusion and Future Work

8.1 Conclusion

In this thesis we have introduced a new formulation for the regularized Bingham fluid flow equations. This new formulation has the advantage of being numerically more robust when the regularization parameter ε gets smaller. We proved well-posedness results for the weak form of the problem and discussed algebraic properties of the discretized equations. In Chapter 5 we have performed several numerical experiments using a finite element method. These experiments show that the number of nonlinear iterations is significantly reduced for the new formulation compared to the classical formulation in primitive variables. The mixed formulation does not become singular in the plug region, so that – even though the theory in this thesis covers only the regularized case – numerical experiments also demonstrate good convergence results for the non-regularized case. For the non-regularized case the mixed formulation is found to compare favorably with ALG based on the formulation by Lion and Glowinski. The drawback of our approach is the augmentation of the problem. For this reason, in Chapter 6 we have proposed an efficient way for solving the linear systems that arise from the discretization and linearization of the non-regularized mixed for-

mulation. These linear systems are solved with a flexible Krylov subspace method preconditioned by a geometric multigrid algorithm. The preconditioner is computed with the regularized problem and is then used for solving the non-regularized Bingham system. This utilization of the regularization parameter ε is novel in the sense that it serves as a preconditioning parameter as opposed to a parameter for regularization purposes. Our numerical experiments presented in Chapter 7 indicate mesh independent convergence of the linear solver in a small number of iterations. Timings in Chapter 6 are obtained here on serial machines, but may be significantly improved on a parallel architecture.

8.2 Future Work

The numerical methods presented in this thesis should be applied to large-scale problems on parallel architectures. We emphasize here in particular the application to problems in computational hemodynamics. The applicability of the Bingham model – or similar models for visco-plastic materials from Chapter 1 – should be investigated. For the efficient solution of the linearized system, many possible research directions exist. The geometric multilevel preconditioner from Chapter 6 has both strengths and limitations. In contrast to its algebraic counterpart, the fine grids in geometric multigrid are the result of a refinement of a coarse mesh. This makes the geometric variant intuitive and projection and interpolation operators between multigrid levels arise naturally from the underlying problem. In the case of algebraic multigrid, the coarse levels are determined using the sparsity pattern of the linear system matrix [20]. Many different ways to do this exist and the problem of finding a “good” coarse level representation of the linear system is often a non-trivial task. However, geometric multigrid has some limitations. Geometric multigrid methods can only be applied to specific meshes satisfying specific requirements. In particular, the fine level

mesh needs to be the result of a refinement of a coarse mesh. This becomes particularly challenging if the computational domain is very complex, since the coarsest mesh may fail to capture the fine details of the domain. For this reason, considering algebraic multigrid methods for solving the linearized and discretized Bingham equations may be worthwhile. We want to point out two other preconditioning techniques that should be investigated in the future:

1. The *incomplete LU* (ILU) factorization method attempts to factorize the matrix \mathcal{A} into $\mathcal{A} = LU$ while dropping entries in L and U that are below a certain user specified dropping tolerance [35]. The ILU factorization is not problem specific and can be applied to very general linear systems with non-singular system matrices. A possible approach for Bingham fluids would be to compute an ILU factorization of \mathcal{A} corresponding to the initial guess \mathbf{u}_0 for the nonlinear iterations and reuse the factorization in the proceeding nonlinear iterations until the performance of the linear solver has deteriorated by a specified amount. The obtained factorization is then discarded and recomputed for the current system. Presumably, the factorization will need to be discarded more frequently for the first few nonlinear iterations than for higher iterations.
2. Inexact factorization techniques presented in Section 6.5 have been very efficient for solving the Navier-Stokes equations, the applicability to Non-Newtonian fluids should be investigated.

Concerning the nonlinear solver, the idea of adding an auxiliary variable to the system could be applied to the fluid flow equations of other types of visco-plastic materials, such as Casson or Herschel-Bulkley fluids. The effectiveness of this procedure should be investigated.

Appendix A

Background

This chapter provides an overview of some basic concepts of numerical and functional analysis that have been of significance in this thesis.

A.1 Solving Linear Systems of Equations

Consider the linear system

$$\mathcal{A}\mathbf{x} = \mathbf{b}, \tag{A.1}$$

with $\mathcal{A} \in \mathbb{R}^{n \times n}$ and vectors $\mathbf{x}, \mathbf{b} \in \mathbb{R}^n$. We assume here that \mathcal{A} is nonsingular. Various methods for solving a linear system of the form (6.1) exist.

A.1.1 Direct Methods

Direct methods, such as Gaussian elimination with partial pivoting, provide a robust and reliable way for solving linear equations and can be applied to very general systems. Their time and memory requirement is predictable [38]. A major drawback of direct methods, however, is their scalability for large systems [11]. In many linear systems arising from real-life applications, including the discretization of partial differential equations, the dimension n is very large and \mathcal{A} is very sparse. The

sparsity of the matrix is crucial for practical reasons, since in numerical codes only the non-zero entries need to be stored. This optimizes the use of memory resources substantially. Direct methods however have the tendency to “destroy” the sparsity of \mathcal{A} , since many zero entries become non-zero during the solving process. This process is called *fill-in* [5]. As a result, this increases the memory requirements significantly and makes the procedure unaffordable for large systems. Also, Gaussian elimination scales with $\mathcal{O}(n^3)$ [8] which is not convenient if n is very large. When solving large and sparse linear systems, iterative schemes are usually the method of choice.

A.1.2 Stationary Iterative Methods

Stationary iterative methods are iterative schemes of the form

$$\mathbf{x}^{(k+1)} = T\mathbf{x}^{(k)} + \mathbf{c}, \quad k = 0, 1, 2, \dots \quad (\text{A.2})$$

where T is the fixed iteration matrix, \mathbf{c} is a fixed vector and $\mathbf{x}^{(0)}$ is an initial guess. Popular examples of stationary iterative methods are the Jacobi method or Gauss-Seidel, which are based on a splitting $\mathcal{A} = \mathcal{B} - \mathcal{C}$, with \mathcal{B} nonsingular, of the matrix \mathcal{A} . The iterative scheme to solve (6.1) is then given by

$$\mathbf{x}^{(k)} = (I - \mathcal{B}^{-1}\mathcal{A})\mathbf{x}^{(k-1)} + \mathcal{B}^{-1}\mathbf{b},$$

which is of the form (A.2). The method converges if the spectral radius of the iteration matrix T is less than one: $\rho(T) < 1$ [59]. Stationary iterative methods are seldom used as a main solver for large linear systems since their convergence is often very slow [8].

A.1.3 Krylov Subspace Methods

We start explaining the concept of Krylov methods by introducing the general projection method, following [82].

The idea of *projection methods* is to find an approximate solution of (6.1) in an m -dimensional subspace \mathcal{K} of \mathbb{R}^n , with $m < n$. Typically, this approximation is obtained by imposing m independent orthogonality conditions. More precisely, the residual vector $\mathbf{f} - \mathbf{A}\mathbf{x}$ is constrained to be orthogonal to m linear independent vectors. This gives rise to another m -dimensional subspace \mathcal{M} of \mathbb{R}^n , the *subspace of constraints*. The projection method in its basic form can be expressed as

Algorithm A.1 Projection Method

while (convergence == false) **do**

 select a pair of subspaces \mathcal{K} and \mathcal{M} ;

 choose bases $V = [v_1, v_2, \dots, v_m]$, $W = [w_1, w_2, \dots, w_m]$ for \mathcal{M} ;

$\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{x}$;

$\mathbf{y} = (W^T \mathbf{A}V)^{-1} W^T \mathbf{r}$;

$\mathbf{x} = \mathbf{x} + V\mathbf{y}$;

 check convergence;

end while

Krylov subspace methods are a particular kind of projection method where the subspace \mathcal{K} is given by the Krylov subspace

$$\mathcal{K}_m(\mathcal{A}, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \mathcal{A}^2\mathbf{r}_0, \dots, \mathcal{A}^{m-1}\mathbf{r}_0\},$$

where $\mathbf{r}_0 = \mathbf{f} - \mathcal{A}\mathbf{x}_0$. The different types of Krylov methods arise from different choices of the subspace \mathcal{M} . Note that the approximation of the solution of (6.1) by a Krylov method has the form

$$\mathcal{A}^{-1}\mathbf{f} \approx \mathbf{x}_m = \mathbf{x}_0 + q_{m-1}(\mathcal{A})\mathbf{r}_0,$$

and in the particular case of $\mathbf{x}_0 = \mathbf{0}$ we have that

$$\mathcal{A}^{-1}\mathbf{f} \approx q_{m-1}(\mathcal{A})\mathbf{f}. \quad (\text{A.3})$$

This means that $\mathcal{A}^{-1}\mathbf{f}$ is approximated by $q_{m-1}(\mathcal{A})\mathbf{f}$. Numerous Krylov subspace methods exist, however we want to mention here three well-known schemes that have become quite popular [62]:

1. The Conjugate Gradient Method [55] (CG) is used when the matrix \mathcal{A} is symmetric positive definite. Here $\mathcal{K} = \mathcal{M} = K_m(\mathcal{A}, \mathbf{r}_0)$.
2. The Minimum Residual Method (MINRES) [69] is used when \mathcal{A} is symmetric. We take $\mathcal{K} = \mathcal{K}_m(\mathcal{A}, \mathbf{r}_0)$ and $\mathcal{M} = \mathcal{A}\mathcal{K}_m(\mathcal{A}, \mathbf{r}_0)$,
3. the Generalized Minimum Residual Method (GMRES) [83] is used for general linear systems, in particular when $\mathcal{A} \neq \mathcal{A}^T$. In this case, $\mathcal{K} = \mathcal{K}_m(\mathcal{A}, \mathbf{r}_0)$ and $\mathcal{M} = \mathcal{A}\mathcal{K}_m(\mathcal{A}, \mathbf{r}_0)$.

A.2 Preconditioning

When using Krylov subspace methods on large and sparse linear systems, using *preconditioning* is inevitable. When solving partial differential equations, the finer the mesh (i.e. the larger the resulting linear system), the more Krylov iterations are usually needed for achieving a given tolerance. The idea behind preconditioning is to solve instead of (6.1) the equivalent system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}. \quad (\text{A.4})$$

The matrix \mathcal{P} is nonsingular and should be chosen so that $\mathcal{P}^{-1}\mathcal{A}$ has in some sense “better” algebraic properties than \mathcal{A} so that the linear solver for system (A.4) con-

verges in a small number of Krylov iterations. In the optimal case, the number of iterations remains constant with growing system size. In the case where \mathcal{A} is *normal*, the convergence behavior of CG, MINRES and GMRES is determined entirely by the spectrum of \mathcal{A} and fast convergence can be achieved by choosing \mathcal{P} so that the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ cluster around one. In the case when \mathcal{A} is non-normal, clustering of the eigenvalues alone may not be sufficient for improving convergence of GMRES. A rigorous understanding of this case is still largely open [62]. When choosing $\mathcal{P} = \mathcal{A}$, any Krylov subspace method applied to (A.4) will converge in exactly one iteration. However, applying \mathcal{P} is obviously as difficult as solving (6.1). Choosing $\mathcal{P} = I$ makes the application of \mathcal{P} trivial, however the preconditioned method will converge in exactly the same number of iterations as when solving the unpreconditioned method (6.1). The challenge in preconditioning is to find the right trade-off between these two extreme cases. A good preconditioner should [11]

- be relatively easy to apply,
- be relatively easy to construct and
- significantly reduce the number of iterations of a Krylov method needed for solving the linear system with a given tolerance.

A large variety of preconditioners exist. Some preconditioners, such as those based on an incomplete factorization of \mathcal{A} , are very general and can be applied to almost all linear systems [11]. Most preconditioners, however, are designed for specific linear systems arising from a specific type of problem and require a solid understanding of the spectrum of \mathcal{A} and the physical problem represented by the linear system. An overview over common techniques for preconditioning can be found in [11].

A.3 Nonlinear Preconditioning

As mentioned in the previous subsection, multigrid methods are a powerful tool for preconditioning an iterative solver. The smoother in Algorithm 6.2 is often a stationary iterative method, since their behavior is independent of the choice of the initial guess \mathbf{x}_0 (the iteration matrix T in (A.2) constant and does not depend on the initial guess). This makes the preconditioner \mathcal{P} in (A.4) a linear operator. However, in some cases Krylov subspace methods are used as smoothers, such as our multilevel preconditioner from Chapter 6. Following (A.3), a Krylov method approximates the inverse of \mathcal{A} using a polynomial q of degree $m - 1$. However, q depends on the choice of the initial guess \mathbf{x}_0 , resulting in a different approximation of \mathcal{A}^{-1} for different initial guesses \mathbf{x}_0 . This means that a preconditioner based on a fixed number of Krylov iterations results in a nonlinear mapping and the preconditioner changes along the iterations of the outer Krylov subspace method. The nonlinearity of Krylov subspace methods is addressed in more detail in [100]. To handle this changing preconditioner, the outer Krylov method needs to be modified to ensure convergence. We mention here one method which is suitable for nonlinear preconditioning, the *flexible GMRES method* [81] (FGMRES). Since this method was used in Chapter 6, we report here the algorithm for flexible GMRES [82]:

Algorithm A.2 FGMRES

- 1: compute $\mathbf{r}_0 = \mathbf{b} - \mathcal{A}\mathbf{x}_0$, $\beta = \|\mathbf{r}_0\|_2$, and $\mathbf{v}_1 = \mathbf{r}_0/\beta$;
 - 2: **for** $j = 1, 2, \dots, m$ **do**
 - 3: compute $\mathbf{z}_j = \mathcal{P}_j^{-1}\mathbf{v}_j$;
 - 4: compute $\mathbf{w} = \mathcal{A}\mathbf{z}_j$;
 - 5: **for** $i = 1, 2, \dots, j$ **do**
 - 6: $h_{i,j} = (\mathbf{w}, \mathbf{v}_i)$;
 - 7: **end for**
 - 8: compute $h_{j+1,j} = \|\mathbf{w}\|_2$ and $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$;
 - 9: define $Z_m = [\mathbf{z}_1, \dots, \mathbf{z}_m]$, $\overline{H}_m = \{h_{i,j}\}_{1 \leq i \leq j+1; 1 \leq j \leq m}$;
 - 10: **end for**
 - 11: compute $\mathbf{y}_m = \operatorname{argmin}_y \|\beta \mathbf{e}_1 - \overline{H}_m \mathbf{y}\|_2$ and $\mathbf{x}_m = \mathbf{x}_0 + Z_m \mathbf{y}_m$;
 - 12: if satisfied stop, else set $\mathbf{x}_0 \leftarrow \mathbf{x}_m$ and go to 1.
-

A.4 Sobolev and Lebesgue Spaces

In Chapter 1 we introduced our notation for Sobolev and Lebesgue spaces. In this section we give some details on these functional spaces. Let throughout this section $1 \leq p \leq \infty$.

Definition A.1 We denote by $L^p(\Omega)$ the set of all functions $f : \Omega \rightarrow \mathbb{R}$, with Ω being an open set, such that

$$\left(\int_{\Omega} |f|^p \right)^{\frac{1}{p}} < \infty.$$

We define the corresponding norm on $L^p(\Omega)$ by

$$\|f\|_p = \left(\int_{\Omega} |f|^p \right)^{\frac{1}{p}}.$$

With $L^\infty(\Omega)$ we denote the set of functions $f : \Omega \rightarrow \mathbb{R}$ satisfying

$$\|f\|_\infty = \operatorname{ess\,sup}_{\mathbf{x} \in \Omega} |f(\mathbf{x})| < \infty.$$

L^p spaces are often called Lebesgue spaces. The functions in Lebesgue spaces are, to be precise, representatives of equivalence classes where the equivalence relation is “equality almost everywhere”, that is, they differ only on a set of measure zero [44]. An important inequality associated with these spaces is the *Hölder inequality*: For $f \in L^p(\Omega)$ and $g \in L^q(\Omega)$ we have that

$$\left| \int_{\Omega} fg d\Omega \right| \leq \|f\|_p \|g\|_q. \quad (\text{A.5})$$

For the special case $p = q = \frac{1}{2}$, this becomes the *Cauchy*, or *Cauchy-Schwarz* inequality

$$\left| \int_{\Omega} fg d\Omega \right| \leq \|f\|_2 \|g\|_2 \quad (\text{A.6})$$

for $f, g \in L^2(\Omega)$.

Definition A.2 We denote by $H^k(\Omega)$ the set of all functions $f : \Omega \rightarrow \mathbb{R}$ in $L^2(\Omega)$ such that

$$\partial^\alpha f \in L^2(\Omega) \text{ for all } \alpha \text{ with } |\alpha| \leq k.$$

We define the corresponding norm on $H^k(\Omega)$ by

$$\|f\|_k = \left(\sum_{|\alpha| \leq k} |\partial^\alpha f|^2 \right)^{\frac{1}{2}}.$$

We refer to the H^k spaces as *Sobolev spaces*.

Here α is a multi-index and derivatives are to be understood in a weak sense. We may define a scalar product on $H^k(\Omega)$ by

$$(u, v) = \sum_{|\alpha| \leq k} \int_{\Omega} D^\alpha u D^\alpha v d\Omega,$$

making the $H^k(\Omega)$ spaces *Hilbert spaces*.

Bibliography

- [1] *Analysis of semi-staggered finite-difference method with application to Bingham flows*, Computer Methods in Applied Mechanics and Engineering, 198 (2009), pp. 975 – 985.
- [2] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *An approximate minimum degree ordering algorithm*, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 886–905.
- [3] P. R. AMESTOY, T. A. DAVIS, AND I. S. DUFF, *Algorithm 837: AMD, an approximate minimum degree ordering algorithm*, ACM Transactions on Mathematical Software, 30 (2004), pp. 381–388.
- [4] ANEURISK-TEAM, *AneuriskWeb project website*. Department of Math&CS, Emory University, <http://ecm2.mathcs.emory.edu/aneuriskweb>, 2012.
- [5] H. ANTIA, *Numerical Methods for Scientists and Engineers*, Birkhäuser, Basel, 2002.
- [6] A. APOSPORIDIS, E. HABER, M. A. OLSHANSKII, AND A. VENEZIANI, *A mixed formulation of the Bingham fluid flow problem: Analysis and numerical solution*, Computer Methods in Applied Mechanics and Engineering, 200 (2011), pp. 2434 – 2446.

- [7] G. ASTARITA, *Letter to the editor: “the engineering reality of the yield stress”*, Journal of Rheology, 34 (1990), pp. 275–277.
- [8] O. AXELSSON, *Iterative Solution Methods*, Cambridge University Press, New York, 1996.
- [9] H. A. BARNES AND K. WALTERS, *The yield stress myth?*, Rheologica Acta, 24 (1985), pp. 323–326.
- [10] I. BASOV AND V. SHELUKHIN, *Nonhomogeneous incompressible Bingham viscoplastic as a limit of nonlinear fluids*, Journal of Non-Newtonian Fluid Mechanics, 142 (2007), pp. 95 – 103.
- [11] M. BENZI, *Preconditioning techniques for large linear systems: A survey*, Journal of Computational Physics, 182 (2002), pp. 418 – 477.
- [12] M. BENZI AND X.-P. GUO, *A dimensional split preconditioner for Stokes and linearized Navier-Stokes equations*, Applied Numerical Mathematics, 61 (2011), pp. 66 – 76.
- [13] M. BENZI, M. NG, Q. NIU, AND Z. WANG, *A relaxed dimensional factorization preconditioner for the incompressible Navier-Stokes equations*, Journal of Computational Physics, 230 (2011), pp. 6185 – 6202.
- [14] M. BENZI AND M. A. OLSHANSKII, *An Augmented Lagrangian-based approach to the Oseen problem*, SIAM Journal on Scientific Computing, 28 (2006), pp. 2095–2113.
- [15] M. BENZI, M. A. OLSHANSKII, AND Z. WANG, *Modified augmented Lagrangian preconditioners for the incompressible Navier-Stokes equations*, International Journal for Numerical Methods in Fluids, 66 (2011), pp. 486–508.

- [16] M. BERCOVIER AND M. ENGELMAN, *A finite-element method for incompressible non-Newtonian flows*, Journal of Computational Physics, 36 (1980), pp. 313–326.
- [17] A. N. BERIS, J. A. TSAMOPOULOS, R. C. ARMSTRONG, AND R. A. BROWN, *Creeping motion of a sphere through a Bingham plastic*, Journal of Fluid Mechanics, 158 (1985), pp. 219–244.
- [18] L. C. BERSELLI, L. DIENING, AND M. RŮŽIČKA, *Existence of Strong Solutions for Incompressible Fluids with Shear Dependent Viscosities*, Journal of Mathematical Fluid Mechanics, 12 (2010), pp. 101–132.
- [19] G. BLAIR, *A Survey of General and Applied Rheology*, Pitman, London, 1944.
- [20] W. BRIGGS, V. HENSON, AND S. MCCORMICK, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [21] A. BROOKS AND T. HUGHES, *Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199–259.
- [22] E. BRUJAN, *Cavitation in Non-Newtonian Fluids: With Biomedical and Bioengineering Applications*, Springer, Berlin, 2010.
- [23] R. BYRON-BIRD, G. C. DAI, AND B. J. YARUSSO, *The rheology and flow of viscoplastic materials*, Reviews in Chemical Engineering, 12 (1983), pp. 2–70.
- [24] X.-C. CAI AND M. SARKIS, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM Journal on Scientific Computing, 21 (1999), pp. 792–797.

- [25] J. CEBRAL, F. MUT, J. WEIR, AND C. PUTMAN, *Association of hemodynamic characteristics and cerebral aneurysm rupture*, American Journal of Neuroradiology, 32 (2010), pp. 264 – 270.
- [26] T. F. CHAN, G. H. GOLUB, AND P. MULET, *A nonlinear primal-dual method for total variation-based image restoration*, SIAM Journal on Scientific Computing, 20 (1999), pp. 1964–1977.
- [27] T. A. DAVIS, *Algorithm 832: UMFPACK V4.3—an unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, 30 (2004), pp. 196–199.
- [28] T. A. DAVIS, *A column pre-ordering strategy for the unsymmetric-pattern multifrontal method*, ACM Transactions on Mathematical Software, 30 (2004), pp. 165–195.
- [29] T. A. DAVIS, *Algorithm 849: A concise sparse Cholesky factorization package*, ACM Transactions on Mathematical Software, 31 (2005), pp. 587–591.
- [30] T. A. DAVIS, *Direct Methods for Sparse Linear Systems*, Fundamentals of Algorithms, Society for Industrial and Applied Mathematics, Philadelphia, 2006.
- [31] T. A. DAVIS AND I. S. DUFF, *An unsymmetric-pattern multifrontal method for sparse LU factorization*, SIAM Journal on Matrix Analysis and Applications, 18 (1997), pp. 140–158.
- [32] T. A. DAVIS AND I. S. DUFF, *A combined unifrontal/multifrontal method for unsymmetric sparse matrices*, ACM Transactions on Mathematical Software, 25 (1999), pp. 1–20.

- [33] E. J. DEAN AND R. GLOWINSKI, *Operator-splitting methods for the simulation of Bingham visco-plastic flow*, Chinese Annals of Mathematics, 23 (2002), pp. 187–204.
- [34] E. J. DEAN, R. GLOWINSKI, AND G. GUIDOBONI, *On the numerical simulation of Bingham visco-plastic flow: Old and new results*, Journal of Non-Newtonian Fluid Mechanics, 142 (2007), pp. 36 – 62.
- [35] J. DIAZ, *Mathematics for Large Scale Computing*, Lecture Notes in Pure and Applied Mathematics, Taylor & Francis, New York, 1989.
- [36] DICOM, *Digital imaging and communications in medicine*.
<http://medical.nema.org/>.
- [37] D. DORAISWAMY, *The origins of rheology: A short historical excursion*, Rheological Bulletin, 71 (2002), pp. 1–9.
- [38] I. DUFF, A. ERISMAN, AND J. REID, *Direct Methods for Sparse Matrices*, Numerical Mathematics and Scientific Computation, Oxford University Press, Oxford, 1986.
- [39] G. DUVAUT AND J. L. LIONS, *Inequalities in Mechanics and Physics*, Springer, Berlin, 1976.
- [40] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.
- [41] H. C. ELMAN AND R. S. TUMINARO, *Boundary conditions in approximate commutator preconditioners for the Navier-Stokes equations*, Electronic Transactions on Numerical Analysis, 35 (2009), pp. 257–280.

- [42] L. C. EVANS, *Partial Differential Equations (Graduate Studies in Mathematics, V. 19)*, American Mathematical Society, Providence, 1998.
- [43] L. FORMAGGIA, A. QUARTERONI, AND A. VENEZIANI, *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, Modeling, Simulation & Applications, Springer, Milan, 2009.
- [44] L. FORMAGGIA, F. SALERI, AND A. VENEZIANI, *Solving Numerical PDEs: Problems, Applications, Exercises*, Unitext / La Matematica Per Il 3+2, Springer, Milan, 2012.
- [45] I. FRIGAARD AND C. NOUAR, *On the usage of viscosity regularisation methods for visco-plastic fluid flow computation*, Journal of Non-Newtonian Fluid Mechanics, 127 (2005), pp. 1 – 26.
- [46] I. FRIGAARD AND O. SCHERZER, *Herschel-Bulkley diffusion filtering: non-Newtonian fluid mechanics in image processing*, ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik, 86 (2006), pp. 474–494.
- [47] G. P. GALDI, A. M. ROBERTSON, R. RANNACHER, S. TUREK, A. M. ROBERTSON, A. SEQUEIRA, AND M. V. KAMENEVA, *Hemorheology*, vol. 37 of Oberwolfach Seminars, Birkhäuser, Basel, 2008.
- [48] A. GAUTHIER, F. SALERI, AND A. VENEZIANI, *A fast preconditioner for the incompressible Navier-Stokes equations*, Computing and Visualization in Science, 6 (2004), pp. 105–112.
- [49] J. M. GINDER, *Rheology controlled by magnetic fields*, Encyclopedia of Applied Physics, 1 (1996).

- [50] V. GIRAULT AND P. RAVIART, *Finite element methods for Navier-Stokes equations: theory and algorithms*, Springer Series in Computational Mathematics, Springer, Berlin, 1986.
- [51] P. P. GRINEVICH AND M. A. OLSHANSKII, *An iterative method for the Stokes-type problem with variable viscosity*, SIAM Journal on Scientific Computing, 31 (2009), pp. 3959–3978.
- [52] W. HACKBUSCH, *Multi-Grid Methods and Applications*, Springer Series in Computational Mathematics, Springer, Berlin, 2003.
- [53] J. P. HARNETT AND R. Y. Z. HU, *The yield stress - an engineering reality*, Journal of Rheology, 33 (1989), pp. 671–679.
- [54] T. HAYAT, M. FAROOQ, Z. IQBAL, AND A. HENDI, *Stretched flow of Casson fluid with variable thermal conductivity*, Walailak Journal of Science and Technology (WJST), 10 (2012).
- [55] M. R. HESTENES AND E. STIEFEL, *Methods of Conjugate Gradients for solving linear systems*, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409–436.
- [56] P. HILD, I. R. IONESCU, T. LACHAND-ROBERT, AND I. ROȘCA, *The blocking of an inhomogeneous Bingham fluid. Applications to landslides*, ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 36 (2002), pp. 1013–1026.
- [57] J. HRON, A. QUAZZI, AND S. TUREK, *A Computational Comparison of Two FEM Solvers for Nonlinear Incompressible Flow*, Ergebnisberichte angewandte Mathematik, Universität Dortmund, Fachbereich Mathematik, Dortmund, 2003.

- [58] G. KARYPIS AND V. KUMAR, *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0*, 2009.
- [59] C. KELLEY, *Iterative Methods for Linear and Nonlinear Equations*, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, Philadelphia, 1987.
- [60] D. J. KLINGENBERG, *Magnetorheology: Applications and challenges*, AIChE Journal, 47 (2001), pp. 246–249.
- [61] R. KRESS, *Linear Integral Equations*, no. v. 82 in Applied Mathematical Sciences, Springer, New York, 1999.
- [62] J. LIESEN AND P. TICHÝ, *Convergence analysis of Krylov subspace methods*, GAMM-Mitteilungen, 27 (2004), pp. 153–173.
- [63] MFEM, *Finite element discretization library*. Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, <http://code.google.com/p/mfem/>, 2012.
- [64] E. MITOULIS, *Flows of viscoplastic materials: models and computations*, in Rheology Reviews, D. M. Binding, N. E. Hudson, and R. Keunings, eds., Glasgow, 2007, Universities Design & Print, pp. 135–178.
- [65] E. MITSOULIS AND T. ZISIS, *Flow of Bingham plastics in a lid-driven square cavity*, Journal of Non-Newtonian Fluid Mechanics, 101 (2001), pp. 173–180.
- [66] E. A. MURAVLEVA AND M. A. OLSHANSKII, *Two finite-difference schemes for the calculation of Bingham fluid flows in a cavity*, Russian Journal of Numerical Analysis and Mathematical Modelling, 23 (2008), pp. 615–634.

- [67] I. NEWTON, T. LESEUR, F. JACQUIER, AND J. WRIGHT, *Philosophiæ naturalis principia mathematica*, no. v. 1-2 in *Philosophiæ naturalis principia mathematica*, ex prelo academico, typis A. et J.M. Duncan, London, 1822.
- [68] A. NIXON, M. GUNEL, AND B. SUMPIO, *The critical role of hemodynamics in the development of cerebral vascular disease*, *Journal of Neurosurgery*, 112 (2010), pp. 1240–1253.
- [69] C. C. PAIGE AND M. A. SAUNDERS, *Solution of sparse indefinite systems of linear equations*, *SIAM Journal on Numerical Analysis*, 12 (1975), pp. 617–629.
- [70] T. C. PAPANASTASIOU, *Flow of materials with yield*, *Journal of Rheology*, 31 (1987), pp. 385–404.
- [71] T. C. PAPANASTASIOU AND A. G. BOUDOUVIS, *Flows of viscoplastic materials: models and computations*, *Computers & Structures*, 64 (1997), pp. 677–694.
- [72] T. PASSERINI, L. SANGALLI, S. VANTINI, M. PICCINELLI, S. BACIGALUPPI, L. ANTIGA, E. BOCCARDI, P. SECCHI, AND A. VENEZIANI, *An integrated statistical investigation of internal carotid arteries of patients affected by cerebral aneurysms*, *Cardiovascular Engineering and Technology*, 3 (2012), pp. 26–40.
- [73] A. PERELOMOVA, *Acoustic heating produced in the thermoviscous flow of a Bingham plastic*, *Central European Journal of Physics*, 9 (2011), pp. 138–145.
- [74] J. PEROT, *An analysis of the fractional step method*, *Journal of Computational Physics*, 108 (1993), pp. 51 – 58.
- [75] M. PICCINELLI, A. VENEZIANI, D. A. STEINMAN, A. REMUZZI, AND L. ANTIGA, *A framework for geometric analysis of vascular structures: Application to cerebral aneurysms*, *IEEE Transactions on Medical Imaging*, 28 (2009), pp. 1141–1155.

- [76] A. PUTZ, I. FRIGAARD, AND D. MARTINEZ, *On the lubrication paradox and the use of regularisation methods for lubrication flows*, Journal of Non-Newtonian Fluid Mechanics, 163 (2009), pp. 62 – 77.
- [77] A. QUARTERONI, F. SALERI, AND A. VENEZIANI, *Analysis of the Yosida method for the incompressible Navier-Stokes equations*, Journal de Mathématiques Pures et Appliquées, 78 (1999), pp. 473 – 503.
- [78] A. QUARTERONI, F. SALERI, AND A. VENEZIANI, *Factorization methods for the numerical approximation of Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 188 (2000), pp. 505 – 526.
- [79] K. RAJU, *Fluid Mechanics, Heat Transfer, and Mass Transfer: Chemical Engineering Practice*, Wiley, Hoboken, 2011.
- [80] N. ROQUET AND P. SARAMITO, *An adaptive finite element method for Bingham fluid flows around a cylinder*, Computer Methods in Applied Mechanics and Engineering, 192 (2003), pp. 3317 – 3341.
- [81] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM Journal on Scientific Computing, 14 (1993), pp. 461–469.
- [82] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, Society for Industrial and Applied Mathematics, Philadelphia, 2nd ed., 2003.
- [83] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM Journal on Scientific Statistical Computing, 7 (1986), pp. 856–869.
- [84] M. SAHIN AND H. J. WILSON, *A semi-staggered dilation-free finite volume method for the numerical solution of viscoelastic fluid flows on all-hexahedral elements*, Journal of Non-Newtonian Fluid Mechanics, 147 (2007), pp. 79 – 91.

- [85] H. SCHMITT, *Numerical simulation of Bingham fluid flow using prox-regularization*, Journal of Optimization Theory and Applications, 106 (2000), pp. 603–626.
- [86] J. SCHÖBERL, J. GERSTMAYR, AND R. GAISBAUER, *NETGEN - automatic 3d tetrahedral mesh generator*. <http://www.hpfem.jku.at/netgen/>, 2003.
- [87] J. SCHÖBERL, R. SIMON, AND W. ZULEHNER, *A robust multigrid method for elliptic optimal control problems*, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1482–1503.
- [88] J. SCHURZ, *The yield stress - An empirical reality*, Rheologica Acta, 29 (1990), pp. 170–171.
- [89] D. SFORZA, C. PUTMAN, AND J. R. CEBRAL, *Hemodynamics of cerebral aneurysms*, Annual Review of Fluid Mechanics, 41 (2009), pp. 91–107.
- [90] B. SMITH, P. BJØRSTAD, AND W. GROPP, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, New York, 2004.
- [91] A. SQUILLACOTE, *The ParaView Guide: A Parallel Visualization Application*, Kitware, Clifton Park, 2007.
- [92] J. STEFFE, *Rheological Methods in Food Process Engineering*, Freeman Press, East Lansing, 1996.
- [93] H. TAKAO, Y. MURAYAMA, Y. QIAN, AND A. MOHAMED, *O-018 a new hemodynamic parameter: energy loss to anticipate aneurysm rupture*, Journal of NeuroInterventional Surgery, 2 (Suppl 1):A8 (2010).

- [94] R. I. TANNER AND J. F. MILTHORPE, *Numerical simulation of the flow of fluids with yield stress*, in *Numerical Methods in Laminar and Turbulent Flow*, C. Taylor, J. A. Johnson, and W. R. Smith, eds., Seattle, 1983, pp. 680–690.
- [95] A. TOSELLI AND O. WIDLUND, *Domain Decomposition Methods - Algorithms and Theory*, Springer Series in Computational Mathematics, Springer, Berlin, 2004.
- [96] J. VAN GIJNM, R. S. KERR, AND G. J. RINKEL, *Subarachnoid haemorrhage*, *Lancet*, 369 (2007), pp. 306–318.
- [97] P. VASSILEVSKI, *Multilevel Block Factorization Preconditioners: Matrix-based Analysis and Algorithms for Solving Finite Element Equations*, Springer, New York, 2008.
- [98] G. VINAY, A. WACHS, AND J.-F. AGASSANT, *Numerical simulation of weakly compressible Bingham flows: The restart of pipeline flows of waxy crude oils*, *Journal of Non-Newtonian Fluid Mechanics*, 136 (2006), pp. 93 – 105.
- [99] G. VINAY, A. WACHS, AND I. FRIGAARD, *Start-up transients and efficient computation of isothermal waxy crude oil flows*, *Journal of Non-Newtonian Fluid Mechanics*, 143 (2007), pp. 141 – 156.
- [100] A. J. WATHEN AND T. REES, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, *Electronic Transactions on Numerical Analysis*, 34 (2009), pp. 125–135.
- [101] K. X. WHIPPLE, *Open-channel flow of Bingham fluids: Applications in debris-flow research*, *The Journal of Geology*, 105 (1997), pp. 243–262.

- [102] H. YANG AND W. ZULEHNER, *Numerical simulation of fluid-structure interaction problems on hybrid meshes with algebraic multigrid methods*, Journal of Computational and Applied Mathematics, 235 (2011), pp. 5367 – 5379.
- [103] YOUTUBE, *Pool filled with Non-Newtonian fluid*. www.youtube.com/watch?v=f2XQ97XHjVw, accessed on October 6th, 2012.
- [104] YOUTUBE, *Stuck in cornstarch pool*. <http://www.youtube.com/watch?v=Lb9kt1z3jAA>, accessed on October 6th, 2012.