NONLINEAR CONTROL FOR NON-NEWTONIAN FLOWS

by

AZIZAH ALRASHIDI

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Mathematics
College of Engineering and Physical Sciences
The University of Birmingham
January 2015

Abstract

PDE-constrained optimisation is an important area in the field of numerical analysis, with problems arising in a wide variety of applications including optimal design, optimal control and parameter estimation. The aim of such problems is to minimize a functional $J(u, d)$ whilst adhering to constraints posed by a system of partial differential equations (PDE), with u and d used respectively to denote the state and control of the system.

In this thesis, we describe the steady-state generalised Stokes equations for incompressible fluids. We proceed to derive the weak formulation of the problem, and show that the resulting system may be written in terms of a mixed formulation of the Stokes problem. Based on this formulation, the problem is discretised through use of the Galerkin finite element method, before investigating control problems based on the generalised Stokes equations, along with numerical experimentation.

This work will be used to achieve the main aim of this thesis, namely the exploration and investigation of solution methods for optimal control problems constrained by non-Newtonian flow. Ultimately, an iterative solution method designed for such problems coupled with an appropriate preconditioning strategy will be described and analysed, and used to produce effective numerical results.

Contents

Acknowledgements

First and foremost, I would like to express my thanks and appreciation to my supervisor, Daniel Loghin. He has been very supportive, not only in an academic capacity, but also with any issue that I have found during my time in Birmingham. I would not have been able to reach this point without his knowledge, inspiration and support.

I would also like to express my deepest thanks to the person who initially inspired me to undertake a PhD, namely my husband Essa Alrashidi. He has supported me in every aspect of my life, not only taking care of my six children but also helping me with pretty much everything that I have needed.

Also, my thanks go to my wonderful children, Bodour, Abdulaziz, Omar, Wrood, Ahmad and my youngest son Lazam. I plan to spend plenty of time with you from now on to make up for the past four years!

I would also like to thanks a number of colleagues within the department who have helped me during my studies. Particular thanks goes to Thomas Reeve for helping in the early stages of my PhD, notably with help understanding LaTeX, and to James Turner in the later stages of my PhD, who was able to provide help despite having a thesis of his own to submit. My thanks also go to Samia Riaz, Chunlei Xu, Sudaba Mohammad and Nina Embleton, as well as other students that I have collaborated with during my time here in Birmingham.

Thanks also go to the School of Mathematics, particularly the Applied Mathematics group for the organisation of the weekly seminar series, and also lunch time applied mathematics seminars specifically designed for postgraduate students. These seminars allowed me to meet and present my work to other students within the department.

Finally, I gratefully acknowledge my scholarship provided by the Public Authority for Applied and Education Training (PAEET), allowing me the opportunity to undertake and complete my PhD studies.

List of Figures

List of Tables

Chapter 1

Introduction

Optimisation problems constrained by partial differential equations (PDEs) arise in a wide variety of important applications such as optimal design, optimal control, and parameter estimation. These types of problems are referred to as PDE-constrained optimisation problems. In the simulation of real-world problems, the state variables require solutions to their governing PDEs by using relevant data from certain control variables. The state variables of the system typically correspond to the displacement, velocity, temperature, electric field and magnetic field, for instance, whereas the control variables are generally represented, for example, by the geometry, coefficients, initial conditions, boundary conditions and source functions. Determination of such variables is the main motivation in order to satisfy certain given aims in performance described through the objective function whilst adhering to both equality and inequality constraints on the behaviour of the system. The control variables are usually described in the form of equality constraints; these equations will be referred to as state equations throughout this thesis.

The general form of a PDE-constrained optimisation problem can be represented as:

$$\begin{aligned} \min_{u,d} \quad & J(u,d) \\ \text{subject to} \quad & \mathbf{c}(u,d) = 0 \\ & \mathbf{h}(u,d) \geqslant 0, \end{aligned} \tag{1.1}$$

where u are the state variables, d the control variables, and J the objective function. Furthermore, $\mathbf{c}$ represents the state equations, and $\mathbf{h}$ the inequality constraints. With respect to the type of objective function and control variables, the optimisation of problems using constrained PDEs, as in equation (1.1), can lead to problems of optimal design, optimal control, or inverse problems with design, control, or inversion variables, respectively represented by control variables. Such PDE related problems and their solutions exist in a number of engineering and scientific fields such as aerodynamics, atmospheric sciences, industrial chemical processes, the environment, geosciences, homeland security, infrastructure, manufacturing, medicine and physics, to name a few. Furthermore, while we usually have a well-posed simulation problem (given d, find u from $\mathbf{c}(u,d) = 0$) , the optimisation problem in equation (1.1) can be ill-posed. Generally, we vary the input quantity (or control) in order to aim at a desired property of an output quantity (or state), and monitor it with respect to some objective function J. The objective functional is usually formulated in a manner such that the desired property is achieved on the minimum of J.

The control of PDEs has the state u as the quantity to be determined in order to solve the PDE, where the control can be an input function described on either the boundary, the whole domain or parts of these. The distributed control problem refers to the situation where the changeable quantity has a domain distribution. In contrast, the boundary control problem corresponds to the case where only boundary conditions of the PDE are changeable.

Recent years have seen research into a wide range of optimal control problems with PDE constraints [43, 63, 62, 59, 54, 49, 74], with examples including Poisson's equation [59, 70, 54, 60], both steady Stokes [63, 59, 54, 55, 48] and unsteady Stokes [71, 54], Navier-Stokes [45, 56] as well as others [57, 58, 21, 22].

However, the literature for investigations into optimal control problems for non-Newtonian flows is fairly sparse. Slawig [69] has produced work in the two-dimensional stationary case, while White [77] has investigated the control of a parabolic equation with a power law differential operator. Non-Newtonian fluid flows in the field of structural optimisation has been considered by Barrett and Liu [8].

To highlight the main issues, consider the following distributed optimal flow control problem subject to the Stokes equation for incompressible fluids on a bounded domain $\Omega \in \mathbb{R}^2$

$$\begin{aligned} \min_{\vec{u},\vec{f}} \qquad & J(\vec{u},\vec{f}) = \frac{1}{2}\left\|\vec{u}-\vec{u}^d\right\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\left\|\vec{f}\right\|^2_{L^2(\Omega)} \\ \text{subject to} \quad & -\nu\Delta\vec{u}+\nabla p = \vec{f} \quad \text{in } \Omega, \\ & \nabla\cdot\vec{u} = 0 \quad \text{in } \Omega, \\ & \vec{u} = \vec{u}_D \quad \text{on } \partial\Omega, \end{aligned} \tag{1.2}$$

where $\vec{u}$ represents the velocity of the fluid, the scalar p represents the pressure, ν is the viscosity, $\vec{f}$ a domain force and $\vec{u}_D$ is a boundary source. Ω represents the domain and $\partial\Omega$ its boundary. In the simulation problem, the variables $\vec{f},\Omega,\nu$ and $\vec{u}_D$ are known and present, where we seek the state $\vec{u}$. In comparison, the optimisation problem requires the determination of certain variables, for example $\vec{f}$ (distributed control), $\vec{u}_D$ (boundary control), or Ω (shape or topology optimisation), in order to minimize some functional of these variables via the control variable and resulting state. In particular, in example (1.2), the only control variable corresponds to the distributed source $\vec{f}$; each of ν, Ω, $\vec{u}_D$ are known. Equation (1.2) includes a Tikhonov term as the cost functional in the second term as a regularising factor in order to remedy an ill-posed situation.

Discretisation of the Stokes equations through use of the Galerkin finite element method is well understood and has been considered within a number of works, including [4, 32, 17]. Furthermore, established solution methods for the resulting saddle point system are prevalent in the literature, not only for the Stokes equations [32, 67, 68], but also for general saddle point problems [79, 50, 80].

When formulating PDE-constrained optimisation problems, we can choose to either discretise first followed by optimisation, or vice versa. In the *discretise-then-optimise* approach, we first discretise the objective function and the PDE constraints and then solve the discrete optimisation problem. In the *optimise-then-discretise* approach, we first consider the necessary conditions for the continuous optimisation problem, followed by discretisation of the resulting system. In the literature, there are conflicting views about which is the best strategy to follow (see, for example, [25]). In this work, we have chosen to discretise first, then optimize. We discretise the objective function J and the PDE using the finite element method.

In this work, the focus will be on the Stokes equations and their generalised versions with specific models for viscosity (such as the power-law model). In the linear case, the well-posedness of a problem and its finite element approximation are well established. Baranger and Najib [6] extended existing results to the nonlinear problem when viscosity adheres to either the power or Carreau laws.

Numerical examples based on jumps in the viscosity will be presented for the Stokes equations, covering the cases of both piecewise constant and also variable viscosity. An investigation into the maximum and minimum eigenvalues of both the full and preconditioned Schur complement matrices suggests a detachment of certain eigenvalues from the rest of the spectrum. We therefore consider an adapted preconditioner based on deflation, which aims to replace the smallest eigenvalues with 1, along with numerical investigations for two test problems.

Using this presentation, we then consider the task of controlling the generalised Stokes equations. A matrix-vector system is then formed through consideration of the first order optimality conditions, and will be solved using preconditioned GMRES. Five preconditioners will be presented and described based on different approximations for relevant matrices. Numerical experimentation will be presented indicating the dependence of each of our preconditioners on the mesh parameter. Further work will consider use of the inner-outer GMRES method [64], from which three solution methods will be described. Numerical results will be provided for two examples based on driven cavity flow and pipe flow problems.

The structure of the thesis can be outlined as follows:

In Chapter 2, we recall some definitions and notation for Sobolev spaces and weak derivatives.

In Chapter 3, we introduce optimisation problems constrained by partial differential equations along with analytical groundwork and optimality theory. We present relevant existence and uniqueness results in researching optimal solutions and produce optimal conditions and optimisation algorithm.

In Chapter 4, we describe the steady-state generalised Stokes equations for incompressible fluids. We derive the weak formulation of the problem and write the formulation as a mixed formulation of the Stokes problem, for which conditions for well-posedness will be discussed. We show that the mixed weak formulation for the Stokes problem fulfils these conditions. We also consider the discretization of the incompressible Stokes equations and formulate the problem in terms of a minimisation problem.

In Chapter 5, we introduce the mixed finite element method that we use to discretise our problem, arriving at a saddle point system which we look to investigate further.

Chapter 6 begins with a general consideration of different solution methods for systems of linear equations. This is followed by an investigation into appropriate preconditioning strategies for saddle point systems, including the notion of deflated preconditioning along with relevant results from the literature. The results from this chapter will then be used in Chapter 7, where we describe the generalised Stokes equations for incompressible fluids. The associated weak formulation will be outlined, where we consider linearisation through use of Picard iterations due to the nonlinearity within the problem. After discretization, the resulting system will then be be described and analysed through numerical experimentation, using preconditioning approaches based on the previous chapter.

In Chapter 8, we consider the problem of controlling the generalised Stokes equations. The distributed control problem will be described, along with the associated discrete form of the problem. By writing down the Lagrangian, a saddle point matrix-vector system is formed from the resulting first order optimality conditions. Five different block upper triangular preconditioning approaches will be considered, both of which consider approximations to the Schur complement, with two of the preconditioners involving a further approximation for the (1,1) block of the system matrix. Both of these preconditioners will then be used in Chapter 9, where numerical results are provided for distributed control problems.

Finally, in Chapter 10 we provide a summary of the thesis and indicate directions for future research.

Chapter 2

Preliminaries

In this chapter, we review some basic definitions, notations and basic theorems of the Sobolev spaces that we will use throughout this thesis. We also introduce the concept of the weak derivative [2, 34].

2.1 Sobolev spaces

We recall that the support of a function u is the closure of the set where u does not vanish:

Definition 2.1.1 *A real valued function u in Ω has support*

$$\textit{supp } u = \overline{\{\mathbf{x} \in \Omega \backslash\, u(\mathbf{x}) \neq 0\}}.$$

If supp $u \subset \Omega$ is a closed and bounded subset of $\mathbb{R}^n$, then we say u has compact support in Ω.

Definition 2.1.2 *An n-tuple $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_n)$, $\alpha_j \in \mathbb{N} \cup \{0\}$ is called a multi index of order $k := |\boldsymbol{\alpha}| = \alpha_1 + ... + \alpha_n$. If $\mathbf{x} \in \mathbb{R}^n$, we denote by $\mathbf{x}^{\boldsymbol{\alpha}}$ the product $x_1^{\alpha_1} ... x_n^{\alpha_n}$.*

Let $D_j = \frac{\partial}{\partial x_j}$, the partial differential operator of order k can be defined as

$$D^{\boldsymbol{\alpha}} = D_1^{\alpha_1}...D_n^{\alpha_n} = \frac{\partial^{|\boldsymbol{\alpha}|}}{\partial x_1^{\alpha_1}...\partial x_n^{\alpha_n}}.$$

Definition 2.1.3 *Let $\boldsymbol{\alpha}$ be a multi index of order k.*

- *The set of continuous, real valued functions defined on Ω which are k times differentiable denoted by $C^k(\Omega)$ is*

$$C^k(\Omega) := \{u : \Omega \to \mathbb{R} :\ D^{\boldsymbol{\alpha}}u \text{ continuous on } \Omega, \text{ for all } \boldsymbol{\alpha} \text{ with } |\boldsymbol{\alpha}| \leqslant k\}.$$

- *The set of continuous, real valued functions defined on Ω which are k times differentiable and whose support is in Ω is denoted by $C_0^k(\Omega)$.*

- *The set of continuous, real valued functions defined on Ω which are infinitely differentiable and whose support is in Ω is denoted by $C_0^\infty(\Omega)$*

$$C_0^\infty(\Omega) = \bigcap_{k\in\mathbb{N}\cup\{0\}} C_0^k(\Omega).$$

2.1.1 L^p-spaces

Let Ω be a bounded open set in $\mathbb{R}^n$ and $u \in \Omega$. We denote the Lebesgue integral of u by $\int_\Omega u(\mathbf{x})\ \mathrm{d}\mathbf{x}$. If $\int_\Omega u(\mathbf{x})\ \mathrm{d}\mathbf{x} < \infty$ then we say u is (Lebesgue-) integrable.

Definition 2.1.4 *Let $1 \leqslant p < \infty$ be a real number. The space of real valued functions whose absolute value raised to the p^{th} power is integrable is*

$$L^p(\Omega) := \left\{ u(\mathbf{x}) : \Omega \to \mathbb{R} : \int_\Omega |u(\mathbf{x})|^p \,\mathrm{d}\mathbf{x} < \infty \right\}.$$

$L^p(\Omega)$ spaces are Banach spaces when equipped with the norm

$$\|u(\mathbf{x})\|_{L^p(\Omega)} := \left(\int_\Omega |u(\mathbf{x})|^p \,\mathrm{d}\mathbf{x}\right)^{\frac{1}{p}}.$$

Therefore

$$L^p(\Omega) = \Big\{u(\mathbf{x}) : \Omega \to \mathbb{R} :\ \|u\|_{L^p(\Omega)} < \infty\Big\}.$$

For the case $p = \infty$ define

$$\|u\|_{L^\infty(\Omega)} = \operatorname*{ess\,sup}_{\mathbf{x}\in\Omega} |u(\mathbf{x})|\,,$$

where ess sup denotes the essential supremum, i.e., the lowest upper bound over Ω excluding subsets of Ω of Lebesgue measure zero. Then the above definition for $L^p(\Omega)$ spaces can be extended to the case $p = \infty$:

$$L^\infty(\Omega) := \Big\{u(\mathbf{x}) : \Omega \to \mathbb{R} :\ \|u\|_{L^\infty(\Omega)} < \infty\Big\}.$$

The case $p = 2$ is a particular instance of $L^p(\Omega)$ where $L^2(\Omega)$ is a Hilbert space when equipped with the inner product

$$(u, v) = \int_\Omega u(\mathbf{x})v(\mathbf{x})\,\mathrm{d}\mathbf{x}.$$

Note that $\|u\|_{L^2(\Omega)} = (u, u)^{\frac{1}{2}}$.

Theorem 2.1.1 (Cauchy-Schwarz inequality) *Let u, $v \in L^2(\Omega)$. Then*

$$|(u, v)| \leqslant \|u\|_{L^2(\Omega)}\,\|v\|_{L^2(\Omega)}\,. \tag{2.1}$$

2.1.2 Generalised (weak) derivatives

Let $u \in C^k(\Omega)$, $w \in C_0^\infty(\Omega)$, and let $\boldsymbol{\alpha}$ be a multi index of order k. The following integration by parts holds

$$\int_\Omega D^{\boldsymbol{\alpha}} u(\mathbf{x}) w(\mathbf{x}) \, \mathrm{d}\mathbf{x} = (-1)^{|\boldsymbol{\alpha}|} \int_\Omega u(\mathbf{x}) D^{\boldsymbol{\alpha}} w(\mathbf{x}) \, \mathrm{d}\mathbf{x},$$

where all terms involving integrals over the boundary of Ω, which arise in the course of integrating by parts, have disappeared because w and all of its derivatives are identically zero on $\partial\Omega$. The previous identity is the basis for the definition of the weak derivative. In the case where $u \notin C^k(\Omega)$, the integration on the left is not defined, however the one on the right is. We use this fact to define the concept of a generalised derivative. First, we define the space of locally integrable functions by

$$L^1_{loc}(\Omega) := \{u : u \in L^1(U), \ \forall U \text{ with } \overline{U} \subset \Omega\}.$$

Definition 2.1.5 *Let $u \in L^1_{loc}(\Omega)$. The function $u^{\boldsymbol{\alpha}}(\mathbf{x}) \in L^1_{loc}(\Omega)$ defined by*

$$\int_\Omega u^{\boldsymbol{\alpha}}(\mathbf{x}) w(\mathbf{x}) \, d\mathbf{x} = (-1)^{|\boldsymbol{\alpha}|} \int_\Omega u(\mathbf{x}) D^{\boldsymbol{\alpha}} w(\mathbf{x}) \, d\mathbf{x}, \quad \forall w \in C_0^\infty(\Omega)$$

is called the generalised (or weak) derivative of u of order k.

Using the above identities we could define the spaces of weakly differentiable functions, known as Sobolev spaces.

2.1.3 $W^{k,p}$ and H^k spaces

Let $\boldsymbol{\alpha}$ be multi index of order $k > 0$, and $p \in [1, \infty]$. We define the Sobolev space of order k, denoted by $W^{k,p}(\Omega)$, which is the set of L^p functions with derivatives of order up to and including k also

$$W^{k,p}(\Omega) := \{u \in L^p(\Omega) : \ D^{\boldsymbol{\alpha}} u \in L^p(\Omega) \text{ for all } \boldsymbol{\alpha} \text{ with } |\boldsymbol{\alpha}| \leqslant k\}.$$

The Sobolev spaces $W^{k,p}(\Omega)$ are Banach spaces when equipped with the norm

$$\|u\|_{W^{k,p}(\Omega)} := \left(\sum_{|\boldsymbol{\alpha}| \leqslant k} \|D^{\boldsymbol{\alpha}} u\|^p_{L^p(\Omega)} \right)^{\frac{1}{p}}, \ 1 \leqslant p < \infty.$$

For the case $p = \infty$, the corresponding norm is

$$\|u\|_{W^{k,\infty}(\Omega)} := \sum_{|\boldsymbol{\alpha}| \leqslant k} \|D^{\boldsymbol{\alpha}} u\|_{L^\infty(\Omega)}.$$

The definition of the Sobolev norms can be more explicit by considering the following seminorm

$$|u|_{W^{k,p}(\Omega)} := \left(\sum_{|\boldsymbol{\alpha}| = k} \|D^{\boldsymbol{\alpha}} u\|^p_{L^p(\Omega)} \right)^{\frac{1}{p}}.$$

We can rewrite the Sobolev norms in terms of its seminorms:

$$\|u\|_{W^{k,p}(\Omega)} = \left(\sum_{j=0}^{k} |u|^p_{W^{j,p}(\Omega)} \right)^{\frac{1}{p}}.$$

The case $p = 2$, is special for Sobolev spaces as it was for L^p-spaces. These spaces are used when solving partial differential equations, so they deserve a special notation. The spaces

$$H^k(\Omega) := W^{k,2}(\Omega)$$

are Hilbert spaces when equipped with the inner product

$$(u, w)_{H^k(\Omega)} = \sum_{|\boldsymbol{\alpha}| \leqslant k} (D^{\boldsymbol{\alpha}} u, D^{\boldsymbol{\alpha}} w)_{L^2(\Omega)}.$$

We write the seminorms on $H^k(\Omega)$ as

$$|u|_{k,\Omega} := \left(\sum_{|\boldsymbol{\alpha}|=k} \|D^{\boldsymbol{\alpha}} u\|^2_{L^2(\Omega)} \right)^{\frac{1}{2}}$$

with the corresponding norms on $H^k(\Omega)$

$$\|u\|_{H^k(\Omega)} = \left(\sum_{j=0}^{k} |u|^2_{j,\Omega} (\Omega) \right)^{\frac{1}{2}}.$$

Some interesting values for k:

- $k = 0$

$$\|u\|_{H^0(\Omega)} = |u|_{0,\Omega} = \|u\|_{L^2(\Omega)}.$$

- $k = 1$

$$|u|_{1,\Omega} = |\nabla u|_{0,\Omega} = \|\nabla u\|_{L^2(\Omega)}$$

$$\|u\|_{H^1(\Omega)} = \left(|u|^2_{0,\Omega} + |u|^2_{1,\Omega} \right)^{\frac{1}{2}} = \left(\|u\|^2_{L^2(\Omega)} + \|\nabla u\|^2_{L^2(\Omega)} \right)^{\frac{1}{2}}.$$

- $k = 2$

$$\|u\|_{H^2(\Omega)} = \left(\|u\|^2_{L^2(\Omega)} + \|\nabla u\|^2_{L^2(\Omega)} + |u|^2_{2,\Omega} \right)^{\frac{1}{2}}.$$

Finally, we state the following useful theorem.

Theorem 2.1.2 (Poincaré-Friedrichs inequality) *Let $\Omega \subset \mathbb{R}^n$ be a bounded open set* *with a sufficiently smooth boundary $\partial\Omega$. Then there exists a constant $C = C(\Omega)$ such that*

$$\|u\|_{L^2(\Omega)} = |u|_{0,\Omega} \leqslant C(\Omega)\,|u|_{1,\Omega} \quad \forall u \in H_0^1(\Omega). \tag{2.2}$$

Chapter 3

PDE-constrained optimisation problems

In this chapter, we introduce optimisation problems constrained by a partial differential equation (PDE) by presenting analytical groundwork and optimality theory. The matter of PDE-constrained optimisation problems along with associated existence and uniqueness theorems has been discussed in the literature, for example in the textbooks of Lions [49], Hinze [43] and Tröltzsch [74]. We will present the existence results, uniqueness results in researching optimal solutions, produce optimal conditions and optimisation algorithms. The form of the problems under consideration here are as follows:

$$\begin{aligned} \min_{w \in W} \quad & J(w) \\ \text{subject to} \quad & e(w) = 0, \\ & c(w) \in K, \;\; w \in C, \end{aligned} \tag{3.1}$$

where $J : W \to \mathbb{R}$ is the objective function, $e : W \to Z$ and $c : W \to R$ are operators where W, Z and R are real Banach spaces, $K \subset R$ is a closed convex cone, and $C \subset W$ is a closed convex set. Usually, the spaces W, Z and R are generalised function spaces and the operator equation $e(w) = 0$ represents a PDE or a system of PDEs. The constraint:

$$c(w) \in K$$

is an abstract inequality constraint. It can often be more efficient, in a bound constrained optimisation problem, to omit inequality constraints and to write constraints as $w \in C$, where $C \subset W$ is a closed convex set. Problems are then considered as follows:

$$\begin{aligned} \min_{w \in W} \quad & J(w) \\ \text{subject to} \quad & e(w) = 0, \;\; w \in C. \end{aligned} \tag{3.2}$$

In order to link with optimisation in finite dimensions, one can consider:

$$W = \mathbb{R}^n, \qquad Z = \mathbb{R}^l, \qquad R = \mathbb{R}^m, \qquad K = (-\infty, 0]^m, \qquad C = \mathbb{R}^n,$$

so that the problem in (3.1) transforms into a problem of nonlinear optimisation

$$\begin{aligned} \min_{w \in W} \quad & J(w) \\ \text{subject to} \quad & e(w) = 0, \\ & c(w) \leqslant 0. \end{aligned} \tag{3.3}$$

Optimisation of optimal control problems with PDE constraints can, typically, lead to separation of the optimisation variable into two parts which produces a second structure, so that there is state $u \in U$ and control $y \in Y$ (both U and Y are Banach spaces). Hence,

$W = U \times Y$, $w = (u, y)$ and the problem becomes

$$\begin{aligned} \min_{u \in U, y \in Y} \quad & J(u, y) \\ \text{subject to} \quad & e(u, y) = 0, \\ & c(u, y) \in K. \end{aligned} \tag{3.4}$$

Definition 3.0.6 *A state-control pair $(\overline{u}, \overline{y}) \in U \times Y$ is called optimal for (3.4), if $e(\overline{u}, \overline{y}) = 0$, $c(\overline{u}, \overline{y}) \in K$ and*

$$J(\overline{u}, \overline{y}) \leqslant J(u, y) \qquad \forall (u, y) \in U \times Y, \quad e(u, y) = 0 \textit{ and } c(u, y) \in K.$$

3.1 Existence of solutions

Let the optimal objective function value be denoted by J^*, which is finite and attainable by using the properties of the problem. We then consider a minimising sequence $(w^k)_{k \in \mathbb{N}}$, i.e., $e(w^k) = 0$, $c(w^k) \leqslant 0$, $J(w^k) \to J^*$ and prove that $(w^k)_{k \in \mathbb{N}}$ is bounded. Next, one can conclude that $(w^k)_{k \in \mathbb{N}}$ contains a convergent subsequence $(w^k)_K \to \overline{w}$, as a result of boundedness. If we assume the continuity of J, e, c then one can note that

$$\begin{aligned} J(\overline{w}) &= \lim_{k \to \infty} J(w^k) = J^* \\ c(\overline{w}) &= \lim_{k \to \infty} c(w^k) \leqslant 0, \\ e(\overline{w}) &= \lim_{k \to \infty} e(w^k) = 0. \end{aligned}$$

Hence, $\overline{w}$ solves the problem. For more details we refer the reader to [43, Chapter 1].

3.2 Existence result for some optimal controls

Definition 3.2.1 *Let X be a Banach space. We say that a sequence $(x_k) \subset U$ converges weakly to some $x \in X$, written $x_k \rightharpoonup x$, if*

$$\lim_{k \to \infty} f(x_k) = f(x) \quad \forall f \in X^*,$$

where X^* is the dual space of X.

Definition 3.2.2 *Let X be a Banach space. A function f is sequentially weakly lower semicontinuous if*

$$f(x) \leqslant \lim_{k \to \infty} \inf f(x_k),$$

for all sequences (x_k) such that $x_k \rightharpoonup x$.

We will present below some theorems for the existence of the optimal controls.

3.2.1 Linear quadratic optimisation problem

Consider the following linear quadratic optimisation problem of the form

$$\min_{(u,y) \in U \times Y} J(u, y)$$

subject to

$$Au + By = g, \quad u \in U_{ad},\, y \in Y_{ad}, \tag{3.5}$$

where the objective function $J(u, y) = \frac{1}{2} \left\| Qu - u^d \right\|_H^2 + \frac{1}{2}\gamma \left\| y \right\|_Y^2$, H, Y are Hilbert spaces, U, Z are Banach spaces, $u^d \in H$, $g \in Z$, $A \in \mathcal{L}(U, Z)$, $B \in \mathcal{L}(Y, Z)$, $Q \in \mathcal{L}(U, H)$. Both the regularization parameter γ and the desired state u^d are given.

We have the following existence result for (3.5)

Theorem 3.2.1 *[43] Let the following assumptions hold*

- *$\gamma \geqslant 0$, $Y_{ad} \subset Y$ is convex, closed and in the case $\gamma = 0$ bounded.*
- *$U_{ad} \subset U$ is convex and closed, such that (3.5) has a feasible point.*
- *$A \in \mathcal{L}(U, Z)$ has a bounded inverse.*

Then the problem (3.5) has an optimal solution $(\overline{u}, \overline{y})$. If $\gamma > 0$ then the solution is unique.

3.2.2 Nonlinear optimization problems

The existence result can be extended to nonlinear problems

$$\min_{u \in U, y \in Y} J(u, y) \quad \text{subject to} \qquad e(u, y) = 0, \;\; u \in U_{ad}, \;\; y \in Y_{ad}, \tag{3.6}$$

where $J : U \times Y \to \mathbb{R}$, $e : U \times Y \to Z$ are continuous with a Banach space Z and reflexive Banach spaces Y, U.

Theorem 3.2.2 *[43] Let the following assumptions hold*

- *$Y_{ad} \subset Y$ is convex, closed and bounded.*
- *$U_{ad} \subset U$ is convex and closed, such that (3.6) has a feasible point.*
- *The state equation $e(u, y) = 0$ has a bounded solution operator $y \in Y_{ad} \mapsto u(y) \in U$.*
- *$(u, y) \in U \times Y \mapsto e(u, y) \in Z$ is continuous under weak convergence.*
- *J is sequentially weakly lower semicontinuous.*

Then the problem (3.6) has an optimal solution $(\overline{u}, \overline{y})$.

The proofs can be found in [43].

3.3 Unique solutions

In general, it is known that if the objective function is not convex then a unique minimum of the objective function can not be expected. However, strict convexity of objective function does guarantee uniqueness. Therefore, we start to present the basic definitions of convex sets and functions. A convex set in $\mathbb{R}^n$ is a set that contains all the points of any line segment joining two points of the set. To be more clear, we introduce some definitions of a convex set and function.

Definition 3.3.1 *A set $S \subseteq \mathbb{R}^n$ is convex if and only if $\forall\, x, y\ \in S$ and $\lambda \in [0,1]$:*

$$\lambda x + (1-\lambda)y \in S.$$

Definition 3.3.2 *Let $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$ and S be a convex set. The function f is convex if and only if $\forall\, x, y\ \in S$ and $\lambda \in [0,1]$:*

$$f\left(\lambda x + (1-\lambda)y\right) \leqslant \lambda f(x) + (1-\lambda) f(y).$$

Definition 3.3.3 *Let $f : S \subseteq \mathbb{R}^n \to \mathbb{R}$ and S be a convex set. The function f is strictly convex if and only if $\forall\, x, y\ \in S$ and $\lambda \in [0,1]$:*

$$f\left(\lambda x + (1-\lambda)y\right) < \lambda f(x) + (1-\lambda) f(y).$$

According to [43], unique solutions to the problem (3.3) can be obtained. For this, J requires strict convexity, e must be linear and c_i must be convex in order to maintain a unique solution.

3.4 Optimality conditions

Let all previous considerations be taken into account. If we assume that the functions J, c and e are continuously differentiable and that the constraints satisfy a regularity condition on the constraints called constraint qualification (CQ) at the solution, then the first order optimality conditions hold at the solution $\overline{w}$ as in the following:

Karush-Kuhn-Tucker (KKT) conditions

Lagrange multipliers $\overline{\theta} \in \mathbb{R}^l$ and $\overline{\lambda} \in \mathbb{R}^m$ exist such that $(\overline{w}, \overline{\theta}, \overline{\lambda})$ solves the following KKT system:

$$\begin{aligned} \nabla J(\overline{w}) + c'(\overline{w})^T\overline{\lambda} + e'(\overline{w})^T\overline{\theta} &= 0, \\ e(\overline{w}) &= 0, \\ c(\overline{w}) \leqslant 0, \quad \overline{\lambda} \geqslant 0, \quad c(\overline{w})^T\overline{\lambda} &= 0, \end{aligned} \tag{3.7}$$

where the column vector $\nabla J(w) = J'(w)^T \in \mathbb{R}^n$ is the gradient of J and $c'(w) \in \mathbb{R}^{m\times n}$, $e'(w) \in \mathbb{R}^{l\times n}$ are the Jacobian matrices of c and e respectively [43].

3.5 Optimisation algorithms

The solutions to the KKT system are the foundations of up-to-date optimisation algorithms and for problems with no inequality constraints, the KKT system reduces to an $(n + l) \times (n + l)$ equation system, as follows

$$G(w, \theta) := \begin{pmatrix} \nabla J(w) + e'(w)^T\theta \\ e(w) \end{pmatrix} = 0. \tag{3.8}$$

We can solve this nonlinear system and find the points that satisfy the KKT conditions.

Lagrange-Newton method

In applying Newton's method to (3.8) we obtain the Lagrange-Newton algorithm, which is a potent algorithm for equality constrained optimisation.

$$\left\{\begin{array}{l} \textbf{For } k = 0, 1, \ldots \\ 1.\textbf{STOP} \text{ if } G(w^k, \theta^k) = 0. \\ 2.\text{Compute } s^k = \begin{pmatrix} s_w^k \\ \\ s_\theta^k \end{pmatrix} \text{ by solving} \\ G'(w^k, \theta^k) s^k = \ -G(w^k, \theta^k) \\ \ \text{and set } w^{k+1} := w^k + s_w^k,\ \theta^{k+1} := \theta^k + s_\theta^k. \\ \textbf{End} \end{array}\right. \tag{3.9}$$

3.6 Summary

In this chapter, we introduced the notion of PDE-constrained optimisation problems. We presented the first order optimality conditions. We also presented the main existence and uniqueness results that we need to have a unique solution to our PDE-constrained optimisation problems later in Chapter 8.

Chapter 4

Stokes problem

The Stokes equation models fluid flow which has a very small velocity, in other words, a very high viscosity. In the next section, we consider the Newtonian case of Stokes problem which is linear, however the non-Newtonian case will be presented in Chapter 7.

4.1 The Stokes problem

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain. The steady-state generalised Stokes equations for incompressible fluids are given by the system of partial differential equations [72, 32]

$$
\begin{aligned}
-\mathrm{div}\,\boldsymbol{\sigma} &= \vec{f} \quad \text{in } \Omega, && (4.1a)\\
\mathrm{div}\,\vec{u} &= 0 \quad \text{in } \Omega, && (4.1b)
\end{aligned}
$$

where the stress tensor

$$\boldsymbol{\sigma} = -pI + \boldsymbol{\sigma}_E, \tag{4.2}$$

the extra stress tensor $\boldsymbol{\sigma}_E$ is a function of shear rate $\varepsilon(\vec{u})$,

$$\varepsilon(\vec{u}) = \frac{1}{2}\left(\nabla\vec{u} + (\nabla\vec{u})^T\right) \text{ and } |\varepsilon(\vec{u})| = \left(\sum_{i,j=1}^{2}(\varepsilon_{i,j}(\vec{u}))^2\right)^{\frac{1}{2}},$$

where the vector notation $\vec{\Box} = (\Box_1, \Box_2)$, where $\Box_1$ and $\Box_2$ are horizontal and vertical components respectively. $\vec{f} : \Omega \to \mathbb{R}^2$ is a given function, while the vector variable $\vec{u}$ represents the velocity of the fluid, and the scalar function p represents the pressure. The first equation (4.1a) is called the momentum equation and represents conservation of the momentum of the fluid, while the second equation (4.1b) is called the mass conservation equation, representing the incompressibility of the fluid. The boundary value problem considered is equation (4.1) posed on a two dimensional domain Ω, with boundary conditions on $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ given by

$$\begin{cases} \vec{u} &= g \quad \text{on } \partial\Omega_D, \\ \mathbf{n} \cdot \boldsymbol{\sigma} &= h \quad \text{on } \partial\Omega_N, \end{cases} \tag{4.3}$$

where $\mathbf{n}$ is the outward-pointing normal to the boundary. If $\partial\Omega = \partial\Omega_D$ then the boundary condition is of Dirichlet type, and the boundary value problem is referred to as the Dirichlet problem for the Stokes equation.

For $\boldsymbol{\sigma}_E = 2\nu\varepsilon(\vec{u})$, equation (4.1) becomes Newtonian, with ν denoting the viscosity, which remains constant in this chapter. In case (4.1) is a linear Stokes system and it can be written as

$$\begin{cases} -\nu\Delta\vec{u} + \nabla p &= \vec{f} \quad \text{in } \Omega, \\ \nabla \cdot \vec{u} &= 0 \quad \text{in } \Omega. \end{cases} \tag{4.4}$$

The aim is to find the velocity and the pressure of the viscous flow with some boundary conditions on the boundary $\partial\Omega$ of the domain Ω.

4.2 Weak formulation

In this section, we consider Dirichlet boundary conditions (other conditions can be found in[32]):

$$\begin{aligned} -\nu\Delta\vec{u} + \nabla p &= \vec{f} \quad \text{in } \Omega, & (4.5a)\\ \nabla\cdot\vec{u} &= 0 \quad \text{in } \Omega, & (4.5b)\\ \vec{u} &= 0 \quad \text{on } \partial\Omega. & (4.5c) \end{aligned}$$

Let $\Omega \subset \mathbb{R}^2$ be an open bounded set and assume $\vec{f} \in [L^2(\Omega)]^2$. The classical solution of the problem (4.5) is defined as the following:

Definition 4.2.1 *A pair $(\vec{u}, p) \in (C^2(\Omega) \cap C^0(\overline{\Omega}))^2 \times C^1(\Omega)$ satisfying the Stokes equations (4.5) is called a classical solution.*

To derive the weak formulation of the Stokes equations, we multiply the first two equations (4.5a) and (4.5b) by test functions $\vec{v} \in V$ and $q \in Q$ respectively, where V and Q are suitable spaces to be defined later, then integrate over the domain Ω:

$$\int_\Omega (-\nu\Delta\vec{u} + \nabla p)\,\vec{v}\,d\Omega = \int_\Omega \vec{f}\,\vec{v}\,d\Omega, \tag{4.6}$$

$$\int_\Omega (\nabla\cdot\vec{u})\,q\,d\Omega \,=\, 0. \tag{4.7}$$

By integrating by parts the left hand side for equation (4.6), we get

$$-\nu\int_\Omega \Delta\vec{u}\,\vec{v}\,d\Omega = \nu\int_\Omega \nabla\vec{u} : \nabla\vec{v}\,d\Omega - \nu\int_{\partial\Omega} (\mathbf{n}\cdot\nabla\vec{u})\cdot\vec{v}\,d\Omega, \tag{4.8}$$

$$\int_\Omega \nabla p \cdot \vec{v}\, d\Omega \;=\; -\int_\Omega p \nabla \cdot \vec{v}\, d\Omega + \int_{\partial\Omega} p\, \mathbf{n} \cdot \vec{v}\, d\Omega. \tag{4.9}$$

combining (4.6), (4.8) and (4.9) gives

$$\nu \int_\Omega \nabla \vec{u} : \nabla \vec{v}\, d\Omega - \int_\Omega p \nabla \cdot \vec{v}\, d\Omega = \int_\Omega \vec{f} \cdot \vec{v}\, d\Omega \tag{4.10}$$

for all $\vec{v} \in V = (H_0^1(\Omega))^2$. Note that here $\nabla \vec{u} : \nabla \vec{v} := \sum_{i,j} \frac{\partial u_i}{\partial x_j} \frac{\partial u_i}{\partial x_j}$, the componentwise scalar product. Since there are no derivatives on the pressure and test function q on the left-hand side of (4.10) and (4.7) respectively, the appropriate space for p and q is $L^2(\Omega)$. However, for the Stokes equations with pure Dirichlet boundary condition, the pressure is unique up to an additive constant. In order to ensure the uniqueness, we impose the condition

$$\int_\Omega p\, d\Omega = 0. \tag{4.11}$$

Therefore we consider the following space:

$$Q := \left\{ q \in L^2(\Omega) : \int_\Omega q = 0 \right\}.$$

The mixed weak formulation of the Stokes problem (4.5) can be established as follows:

$$\begin{cases} \text{Find } \vec{u} \in V \text{ and } p \in Q, \text{ such that for all } \vec{v} \in V \text{ and } q \in Q, \\ \nu \int_\Omega \nabla \vec{u} : \nabla \vec{v}\, d\Omega - \int_\Omega p \nabla \cdot \vec{v}\, d\Omega = \int_\Omega \vec{f} \cdot \vec{v}\, d\Omega, \\ -\int_\Omega (\nabla \cdot \vec{u}) q\, d\Omega = 0. \end{cases} \tag{4.12}$$

This can be written in the form:

$$\begin{cases} \text{Find } \vec{u} \in V \text{ and } p \in Q, \text{ such that for all } \vec{v} \in V \text{ and } q \in Q, \\ a(\vec{u},\vec{v}) + b(\vec{v},p) = l(\vec{v}), \\ b(\vec{u},q) = 0. \end{cases} \tag{4.13}$$

where

$$\begin{aligned} a(\vec{u},\vec{v}) &= \nu \int_\Omega \nabla\vec{u} : \nabla\vec{v} \; d\Omega, \\ b(\vec{v},p) &= -\int_\Omega p \, \nabla \cdot \vec{v} \; d\Omega, \\ l(\vec{v}) &= \int_\Omega \vec{f} \cdot \vec{v} \; d\Omega. \end{aligned}$$

The derivation of the weak formulation indicates that any solution of (4.5) satisfies (4.12). Now the question that will arise is if problem (4.12) is well posed. To address this matter we will present the mixed formulation problem in the next section.

4.3 Mixed formulation

Consider the following abstract mixed formulation

$$\begin{cases} \text{Find } \vec{u} \in V \text{ and } p \in Q, \text{ such that for all } \vec{v} \in V \text{ and } q \in Q, \\ a(\vec{u},\vec{v}) + b(\vec{v},p) = l(\vec{v}), \\ b(\vec{u},q) = 0. \end{cases} \tag{4.14}$$

The well-posedness of (4.14) is decided in the following theorem which was proved by Brezzi for this type of problems as presented in [16],

Theorem 4.3.1 *The problem (4.14) has a unique solution if the following conditions hold*

(i) $a(\cdot,\cdot)$ *and* $b(\cdot,\cdot)$ *are continuous bilinear forms on* $(V \times V)$ *and* $(V \times Q)$ *respectively,*

i.e., the following inequalities are assumed to hold:

$$a(\vec{u},\vec{v}) \leqslant C_a \left\|\vec{u}\right\|_V \left\|\vec{v}\right\|_V \; \forall \vec{u},\vec{v} \in V \tag{4.15}$$

and

$$b(\vec{v},p) \leqslant C_b \left\|\vec{v}\right\|_V \left\|p\right\|_Q \; \forall \vec{v} \in V,\, p \in Q. \tag{4.16}$$

(ii) the bilinear form $a(\cdot,\cdot)$ is coercive over V, i.e

$$a(\vec{v},\vec{v}) \geqslant \alpha \left\|\vec{v}\right\|_V^2 \;\; \forall \vec{v} \in V,$$

where $\alpha \; > 0$.

(iii) the bilinear form $b(\cdot,\cdot)$ satisfies the inf-sup *condition, i.e.*

$$\inf_{q \in Q} \sup_{\vec{v} \in V} \frac{b(\vec{v},q)}{\left\|\vec{v}\right\|_V \left\|q\right\|_Q} = \beta.$$

where $\beta \; > 0$.

(iv) The linear form $l(\vec{v})$ is bounded.

The inf-sup condition is also called the LBB condition as it refers to similar contributions by Ladyzhenskaya, Brezzi and Babuška. The mixed weak formulation for the Stokes problem (4.12) fulfils the conditions of Theorem (4.3.1) w.r.t. $\left\|\cdot\right\|_V$ and $\left\|\cdot\right\|_Q$ with V, Q defined in Section 4.2. In particular, we have

- Continuity of $a(\cdot,\cdot)$(by using Cauchy-Schwarz inequality (2.1)):

$$\begin{aligned} a(\vec{u},\vec{v}) &= (\nabla\vec{u},\nabla\vec{v}) \\ &\leqslant \left\|\nabla\vec{u}\right\|_{L^2(\Omega)} \left\|\nabla\vec{v}\right\|_{L^2(\Omega)} \\ &= \left|\vec{u}\right|_{1,\Omega} \left|\vec{v}\right|_{1,\Omega} \end{aligned}$$

- Continuity of $b(\cdot,\cdot)$:
$$\begin{aligned} b(\vec{v},p) &= (p,\nabla\vec{v}) \\ &\leqslant \|p\|_{L^2(\Omega)}\,\|\nabla\vec{v}\|_{L^2(\Omega)} \\ &= \|p\|_{L^2(\Omega)}\,|\vec{v}|_{1,\Omega} \end{aligned}$$

- Coercivity of $a(\cdot,\cdot)$ on V (by using Poincaré-Friedrichs inequality (2.2)):
$$\begin{aligned} a(\vec{u},\vec{u}) &= (\nabla\vec{u},\nabla\vec{u}) \\ &= \|\nabla\vec{u}\|^2_{L^2(\Omega)} = \tfrac{1}{2}\|\nabla\vec{u}\|^2_{L^2(\Omega)} + \tfrac{1}{2}\|\nabla\vec{u}\|^2_{L^2(\Omega)} \\ &\geqslant \tfrac{1}{2}\|\nabla\vec{u}\|^2_{L^2(\Omega)} + \tfrac{1}{2C(\Omega)}\|\vec{u}\|^2_{L^2(\Omega)} \\ &\geqslant \tfrac{1}{2}\min\Big\{1,\tfrac{1}{C(\Omega)}\Big\}\Big(\|\nabla\vec{u}\|^2_{L^2(\Omega)} + \|\vec{u}\|^2_{L^2(\Omega)}\Big) \\ &= \tfrac{1}{2}\min\Big\{1,\tfrac{1}{C(\Omega)}\Big\}\|\vec{u}\|^2_{[H^1(\Omega)]^2} \end{aligned}$$

- To verify the inf-sup condition we present the following theorem

Theorem 4.3.2 *Let $q \in L^2_0$. Then there exists $\vec{w} \in [H^1(\Omega)]^2$ with $-div\,\vec{w} = q$ and a constant $c > 0$ such that*
$$\|\vec{w}\|_{[H^1(\Omega)]^2} \leqslant c\,\|q\|_{L^2(\Omega)}\,.$$

As a consequence, we have
$$\sup_{\vec{v}\in[H^1(\Omega)]^2}\frac{b(\vec{v},q)}{\|\vec{v}\|_{H^1(\Omega)}} \geqslant \frac{b(\vec{w},q)}{\|\vec{w}\|_{H^1(\Omega)}} \geqslant \frac{\|q\|^2_{L^2(\Omega)}}{\|\vec{w}\|_{H^1(\Omega)}} \geqslant \frac{1}{c}\|q\|_{L^2(\Omega)}\,,$$

which implies the inf-sup condition.

4.4 Discrete weak formulation

The finite element discretisation of partial differential equations is based on its weak formulation to obtain the discrete formulation. A discrete weak formulation of Stokes problem is defined by using finite dimensional velocity space $V_h \subset V$ and pressure space $Q_h \subset Q$. Then the discrete mixed weak formulation of (4.14) is given by:

$$\begin{cases} \text{Find } \vec{u}_h \in V_h \text{ and } p_h \in Q_h \text{ such that for all } \vec{v}_h \in V_h \text{ and } q_h \in Q_h \\ a(\vec{u}_h, \vec{v}_h) + b(\vec{v}_h, p_h) = (\vec{f}_h, \vec{v}_h), \\ b(\vec{u}_h, q_h) = 0. \end{cases} \tag{4.17}$$

We have the following existence and uniqueness theorem and error estimates, as given in [16]

Theorem 4.4.1 *Assume that the following conditions hold*

(i) the bilinear form $a(\cdot,\cdot)$ is coercive over V_h, i.e. there exists $\alpha^* > 0$ such that

$$a(\vec{v}_h, \vec{v}_h) \geqslant \alpha^* \left\| \vec{v}_h \right\|_{V_h}^2 \ \ \forall \vec{v}_h \in V_h,$$

(ii) the bilinear form $b(\cdot,\cdot)$ satisfies the inf-sup condition, i.e. there exists a constant $\beta^* > 0$ such that

$$\inf_{q_h \in Q_h} \sup_{\vec{v}_h \in V_h} \frac{b(\vec{v}_h, q_h)}{\left\| \vec{v}_h \right\|_{V_h} \left\| q_h \right\|_{Q_h}} \geqslant \beta^*.$$

Then the problem (4.17) has a unique solution $(\vec{u}_h, p_h) \in (V_h, Q_h)$ for all $h > 0$. Moreover if $(\vec{u}, p)$ is the solution of (4.14), then there exists a constant $C > 0$, depending only on C_a (4.15), C_b (4.16), α^* and β^* such that

$$\left\| \vec{u} - \vec{u}_h \right\|_V + \left\| p - p_h \right\|_Q \leqslant C \left\{ \inf_{\vec{v}_h \in V_h} \left\| \vec{u} - \vec{v}_h \right\|_V + \inf_{q_h \in Q_h} \left\| p - q_h \right\|_Q \right\}.$$

4.5 Compatible spaces

It is well known that for the mixed finite formulation of the Stokes problem the inf sup condition does not always hold and that we need to ensure that the pair of finite element spaces (V_h, Q_h) is compatible in this sense and the corresponding discretisation is well posed. This condition is also called the compatibility condition and the pair (V_h, Q_h) satisfying this condition is called a compatible pair. In [17], Brezzi and Fortin have published the necessary and sufficient conditions for the stability of the mixed finite element discretisation of the Stokes problem. A pair of finite element spaces (V_h, Q_h) satisfying these conditions is called stable, and so is the corresponding discretisation. Some stable mixed finite element spaces are listed in the next chapter. A full study of other stable and unstable approximations can be found in [17] and [37].

4.6 Test problems

The following two examples of two-dimensional Stokes flow will be used within this thesis as test problems.

Example 1 ***(Driven cavity test problem)***

Let $\Omega = [0,1]^2$. Consider the Stokes problem

$$\begin{cases} -\Delta\vec{u} + \nabla p &= \vec{f} \qquad \textit{in } \Omega, \\ \nabla \cdot \vec{u} &= 0 \qquad \textit{in } \Omega, \\ \vec{u} &= \vec{u}_D \quad \textit{on } \partial\Omega, \end{cases} \tag{4.18}$$

where

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 16(x - x^2)^2 \\ 0 \end{pmatrix} & \textit{on } y = 1 \\ \mathbf{0} & \textit{otherwise} \end{cases}$$

The above example represents a driven flow in a cavity. It is a classic test problem which is used in fluid dynamics. The geometry of this test problem is illustrated in Figure 4.1 for testing a numerical simulation. It has a square flow model which moves from left to right. The boundary conditions for the velocity are $\vec{u}_D$ on the top (lid) which moves in the

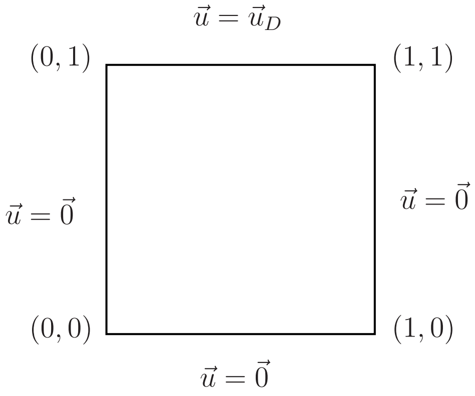


Figure 4.1: Geometry of the driven cavity test problem

x-direction and no slip condition at the rest of the boundaries. It has a simple geometry and straight forward flow structure, which makes this example an attractive test case. The velocity $\vec{u}$ and the pressure p are the solution of the driven cavity flow in a square domain $[0,1]^2$. Figure 4.2 shows the velocity in horizontal and vertical components u_1, u_2 and pressure p. The other test example is a pipe flow, where the domain is a horizontal pipe as follows:

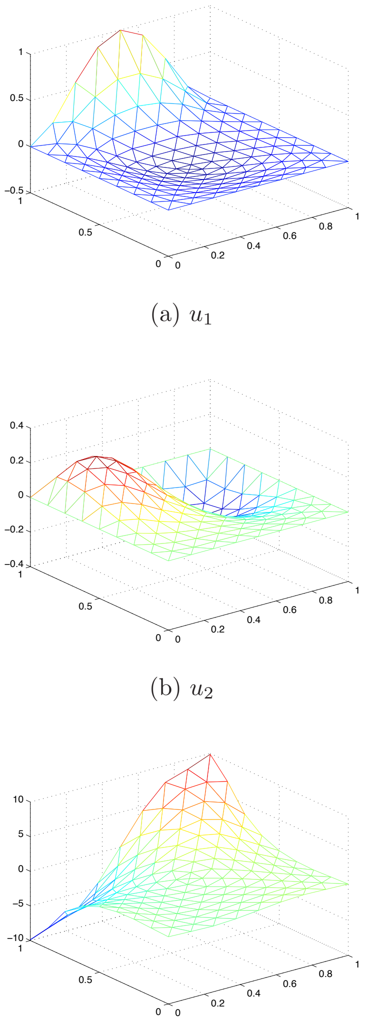
Example 2 ***(Pipe flow test problem)***

Let $\Omega = [0,4] \times [0,1]$, consider the Stokes problem

$$\begin{cases} -\Delta \vec{u} + \nabla p &= \vec{f} \qquad in\ \Omega, \\ \nabla \cdot \vec{u} &= 0 \qquad in\ \Omega, \\ \vec{u} &= \vec{u}_D \quad on\ \partial\Omega_D, \\ \frac{\partial \vec{u}}{\partial n} &= 0 \qquad on\ \partial\Omega_N, \end{cases} \tag{4.19}$$

where

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 4(y-y^2) \\ 0 \end{pmatrix} & on\ x = 0 \\ \mathbf{0} & on\ y = 0,\ y = 1, \end{cases}$$



(a) u_1

(b) u_2

(c) p

Figure 4.2: The velocity components u_1, u_2 and the pressure p for the driven cavity test problem.

and $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$

The geometry of this test problem is illustrated in Figure 4.3. The resulting velocity has a parabolic profile in the horizontal direction, displaying gradual decay from a maximum at the center to zero at both pipe walls and outflow boundary condition (at $x = 4$). Vertically, the velocity component is zero. For the pressure p, there is a pressure drop

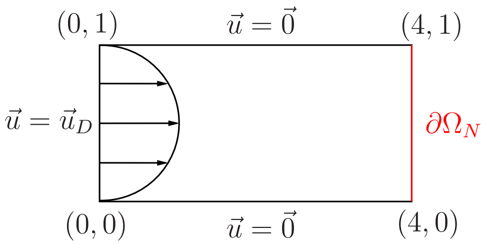


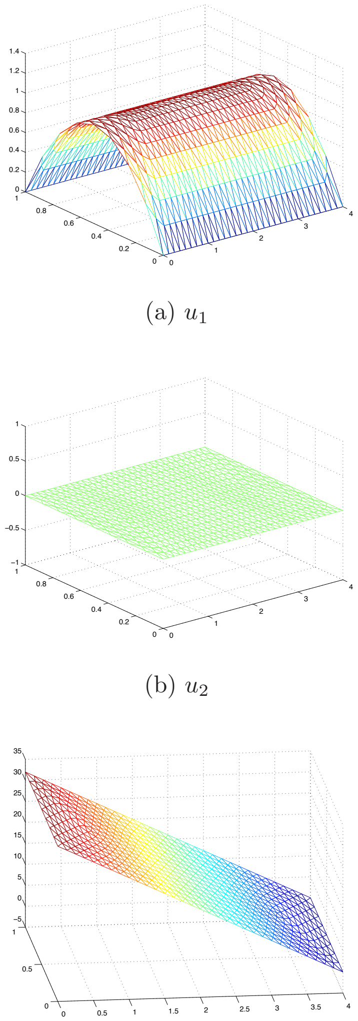
Figure 4.3: Geometry of pipe flow test problem

occurs along the pipe. Figure 4.4 shows the velocity in horizontal and vertical components u_1, u_2 and pressure p. Figure 4.5 shows the streamlines for the driven cavity and the pipe flow test problems..

4.7 Summary

In conclusion, we have derived the weak formulation of the Stokes equations. We stated the existence and uniqueness theorem and error estimates for this problem. Furthermore, we presented the necessary and sufficient conditions for the stability of the mixed finite element discretisation of the Stokes problem.

In the next chapter, we introduce the method that we use to discretise our weak formulation.



(a) u_1

(b) u_2

(c) p

Figure 4.4: The velocity components u_1, u_2 and the pressure p for the pipe flow test problem.

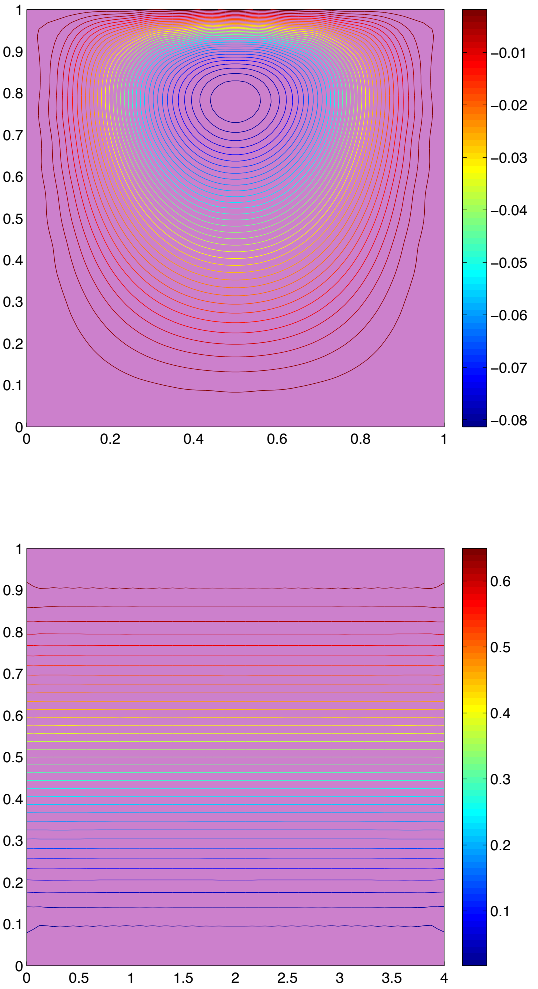


Figure 4.5: The streamlines for $\vec{u}$ for the driven cavity and the pipe flow test problems.

Chapter 5

Finite Element Method For Stokes

In this chapter, we want to present the method that we used to discretise our problem. We have various choices for the discretisation of the Stokes equations. The finite element method (FEM) is considered as one of the well established and convenient methods for solving partial differential equations numerically [23, 27]. Moreover, the Stokes problem is well understood when discretised with the FEM [4, 18, 17, 37, 28, 32]. Also for these problems, we have an error analysis already available unlike other discretisation methods [75, 3]. By taking all these considerations into account, we have chosen the FEM to discretise the Stokes equations. We also use this method to discretise our optimisation problems in later chapters.

The general steps to solve a given differential equation using the finite element method are basically the following

1. Set up a weak formulation of the differential equation.
2. Set up the discrete weak formulation by restricting the weak formulation to a finite dimensional setting.
3. Setup (linear) system of equations and solve the discrete problem.

5.1 Discretised weak formulation

Recall the weak formulation of the Stokes equations

$$\begin{cases} \text{Find } \vec{u} \in V \text{ and } p \in Q, \text{ such that for all } \vec{v} \in V \text{ and } q \in Q, \\ \nu \displaystyle\int_\Omega \nabla \vec{u} : \nabla \vec{v} \; dxdy - \int_\Omega p \, \nabla \cdot \vec{v} \; dxdy = \int_\Omega \vec{f} \cdot \vec{v} \; dxdy, \\ -\displaystyle\int_\Omega (\nabla \cdot \vec{u}) q \; dxdy = 0. \end{cases} \tag{5.1}$$

In order to discretise the weak formulation, we define the finite dimensional spaces $V_h \subset V$ and $Q_h \subset Q$ and consider the following discrete problem

$$\begin{cases} \text{Find } \vec{u}_h \in V_h \text{ and } p_h \in Q_h, \text{ such that for all } \vec{v}_h \in V_h \text{ and } q_h \in Q_h, \\ a(\vec{u}_h, \vec{v}_h) + b(\vec{v}_h, p_h) = l(\vec{v}_h), \\ b(\vec{u}_h, q_h) = 0. \end{cases} \tag{5.2}$$

where

$$\begin{aligned} a(\vec{u}_h, \vec{v}_h) &= \nu \int_\Omega \nabla \vec{u}_h : \nabla \vec{v}_h \; dxdy, \\ b(\vec{v}_h, p_h) &= -\int_\Omega p_h \, \nabla \cdot \vec{v}_h \; dxdy, \\ l(\vec{v}_h) &= \int_\Omega \vec{f}_h \cdot \vec{v}_h \; dxdy. \end{aligned}$$

In our problem, we look for $(\vec{u}_h, p_h) \in V_h \times Q_h$ where V_h, Q_h are spaces of continuous piecewise polynomials associated with a subdivision $\mathcal{T}_h$. Let $\mathcal{T}_h$ be a set of disjoint simplices of Ω such that $\{T_i\}_{1 \leqslant i \leqslant N} \equiv \mathcal{T}_h$ and $\cup_{i=1}^{N} T_i = \Omega_h$, where Ω_h is an approximation of Ω since Ω is not always a polygon (in $\mathbb{R}^2$) or a polyhedron (in $\mathbb{R}^3$). The finite element spaces of velocity and pressure V_h, Q_h are chosen to be

$$\begin{aligned} V_h &= \left\{ \vec{v}_h : \Omega_h \mid \vec{v}_{h|T_i} \in \mathcal{P}_k(T_i) \right\} \cap C^0(\Omega_h), \\ Q_h &= \left\{ q_h : \Omega_h \mid q_{h|T_i} \in \mathcal{P}_l(T_i) \right\} \cap C^0(\Omega_h), \end{aligned} \tag{5.3}$$

where $\mathcal{P}_k(T_i)$ denotes the space of polynomials of degree k on T_i , $k > l \geqslant 0$. Some commonly used compatible pairs of spaces are listed below [4, 17] .

1. $\mathcal{P}_2 - \mathcal{P}_1$ elements.

   This combination involves a quadratic polynomial approximation for velocity and linear polynomial approximation for pressure (see Figure 5.1):

$$\begin{aligned} V_h &= \left\{\vec{v}_h : \Omega_h \mid \vec{v}_{h|T_i} \in \mathcal{P}_2(T_i)\right\}, \\ Q_h &= \left\{q_h : \Omega_h \mid q_{h|T_i} \in \mathcal{P}_1(T_i)\right\}. \end{aligned} \tag{5.4}$$

2. $\mathcal{P}_2 - \mathcal{P}_0$ elements.

   The $\mathcal{P}_2 - \mathcal{P}_0$ pair is another stable approximation where the velocity is approximated by a quadratic polynomial and the pressure by a piecewise constant approximation.

$$\begin{aligned} V_h &= \left\{\vec{v}_h : \Omega_h \mid \vec{v}_{h|T_i} \in \mathcal{P}_2(T_i)\right\}, \\ Q_h &= \left\{q_h : \Omega_h \mid q_{h|T_i} \in \mathcal{P}_0(T_i)\right\}. \end{aligned} \tag{5.5}$$

These spaces are also graphically denoted by indicating the degree of the polynomial approximation on a generic simplex.

Let

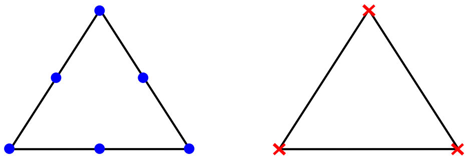


Figure 5.1: The $\mathcal{P}_2$ and $\mathcal{P}_1$ triangle elements.

$$\left\{\vec{\phi}_i : i = 1, \dots, n_u\right\}$$

$$\{\psi_i : i = 1, \dots, n_p\}$$

denote basis for V_h and Q_h respectively. Then

$$\vec{u}_h = \sum_{i=1}^{n_u} u_i \, \vec{\phi}_i \,, \quad p_h = \sum_{j=1}^{n_p} p_j \, \psi_j \tag{5.6}$$

and the weak formulation becomes for the choice $\vec{v}_h = \vec{\phi}_j$, $q_h = \psi_k$,

$$\begin{aligned} \sum_{i=1}^{n_u} u_i \, a(\vec{\phi}_i, \vec{\phi}_j) - \sum_{l=1}^{n_p} p_l \, b(\psi_l, \vec{\phi}_j) = & \quad \textstyle\sum_{i=1}^{n_u} l(f_i, \vec{\phi}_i), \\ \sum_{i=1}^{n_u} u_i \, b(\psi_k, \vec{\phi}_i) = & \qquad\qquad 0. \end{aligned}$$

.

which in matrix form reads

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \tag{5.7}$$

where the matrix $\mathbf{A}$ is a symmetric and positive definite square matrix, which is called the stiffness matrix, and the matrix $\mathbf{B}$ is called the divergence matrix:

$$\mathbf{A}_{ij} = a(\vec{\phi}_i, \vec{\phi}_j)$$

$$\mathbf{B}_{ki} = b(\psi_k, \vec{\phi}_i)$$

In order to evaluate **A** we write

$$\begin{aligned}
a(\vec{\phi}_i, \vec{\phi}_j) &= \nu \int_\Omega \nabla\vec{\phi}_i : \nabla\vec{\phi}_j \; dxdy \\
&= \sum_{T_k \in \mathcal{T}_h} \nu \int_{T_k} \nabla\vec{\phi}_i : \nabla\vec{\phi}_j \; dxdy \\
&= \nu \sum_{k=1}^{N} A_{ij}^{(k)},
\end{aligned}$$

where $A_{ij}^{(k)}$ is a sparse matrix containing contribution corresponding to the support of $\vec{\phi}_i$ and $\vec{\phi}_j$ jointly, i.e.

$$\begin{aligned}
\mathbf{A}_{ij}^{(k)} &= \int_{\text{supp } \vec{\phi}_i \cap \text{ supp } \vec{\phi}_j} \nabla\vec{\phi}_i : \nabla\vec{\phi}_j \; dxdy \\
&= \sum_{T_k \in \mathcal{T}_h} \nu \int_{T_k} \nabla\vec{\phi}_i : \nabla\vec{\phi}_j \; dxdy \\
&= \nu \sum_{T \in \tau_{ij}} \int_{T} \nabla\vec{\phi}_i : \nabla\vec{\phi}_j \; dxdy,
\end{aligned}$$

where

$$\tau_{ij} = \left\{ T : \, T \in \text{supp } \vec{\phi}_i \cap \text{ supp } \vec{\phi}_j \right\}.$$

It is only when this intersection is nonempty that we get the entries $\mathbf{A}_{ij}$, as a consequence the matrix **A** will be largely sparse.

5.2 Basis elements

We construct $\vec{\phi}_i$ and ψ_j by locally constructing a basis for polynomial spaces $\mathcal{P}_k$ and $\mathcal{P}_l$ on an element T by imposing the conditions

$$\phi_i(\mathbf{x}_k) = \delta_{ik}$$

$$\psi_i(\mathbf{x}_l) = \delta_{jl},$$

where $\mathbf{x}_k$ is a node of element T. The basis for V_h has elements for the form

$$\vec{\phi}_i = \left\{ \begin{pmatrix} \phi_1 \\ 0 \end{pmatrix}, \ldots, \begin{pmatrix} \phi_n \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ \phi_1 \end{pmatrix}, \ldots \begin{pmatrix} 0 \\ \phi_n \end{pmatrix} \right\}.$$

This local basis is known as the set of shape functions. The implementation of the FEM is usually done using certain techniques. We describe one in next section. We focus on the grid of triangular simplices ($\mathbb{R}^2$).

5.3 Transformation to the reference element

The idea here is to map an arbitrary triangle element in the (x, y)-plane to another one which has a simpler geometry in a computational sense, denoted by (ξ, η)-plane. Consider an arbitrary triangle Δ_k with vertex nodes $(x_i, y_i),\ i = 1, 2, 3$ which is mapped by a linear transformation to the reference (canonical) triangle, denoted by Δ_E with vertices at (0,0), (1,0), (0,1) (see Figure 5.2). The mapping is defined for all points $(x, y) \in \Delta_k$, as in

$$x(\xi, \eta) = x_1\chi_1(\xi, \eta) + x_2\chi_2(\xi, \eta) + x_3\chi_3(\xi, \eta), \tag{5.8}$$

$$y(\xi, \eta) = y_1\chi_1(\xi, \eta) + y_2\chi_2(\xi, \eta) + y_3\chi_3(\xi, \eta), \tag{5.9}$$

where

$$\begin{aligned} \chi_1(\xi, \eta) &= 1 - \xi - \eta \\ \chi_2(\xi, \eta) &= \xi \\ \chi_3(\xi, \eta) &= \eta \end{aligned} \tag{5.10}$$

are the basis functions defined on the reference element. As an aside, curve-sided elements can be produced via analogous mapping as in $\mathcal{P}_2$ element basis functions of the reference triangles. Evidently, polynomial mapping onto Δ_k from a reference element needs to be differentiable, hence, with a differentiable function $\varphi(\xi,\eta)$, derivative transformation is via

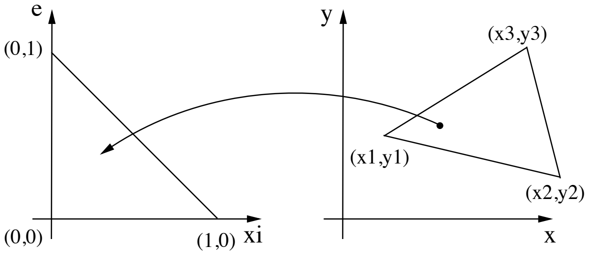


Figure 5.2: The mapping to canonical triangle.

$$\begin{bmatrix} \dfrac{\partial\varphi}{\partial\xi} \\ \dfrac{\partial\varphi}{\partial\eta} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial\xi} & \dfrac{\partial y}{\partial\xi} \\ \dfrac{\partial x}{\partial\eta} & \dfrac{\partial y}{\partial\eta} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{bmatrix}. \tag{5.11}$$

In order to facilitate calculation of the Jacobian matrix in (5.11) we can substitute (5.10) into (5.8)-(5.9) and differentiate to obtain

$$J = \frac{\partial(x,y)}{\partial(\xi,\eta)} = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix}. \tag{5.12}$$

Hence in this case, J is a constant matrix over the reference element and the determinant

$$|J| = \begin{vmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{vmatrix} = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2\ \text{area}(\Delta_k) \tag{5.13}$$

is the ratio of the area of the mapped element Δ_k with respect to reference element Δ_E. It is noteworthy that $|J_k(\xi,\eta)| \neq 0\,\forall(\xi,\eta) \in \Delta_E$, which guarantees a unique and differentiable inverse mapping from Δ_k onto the reference element Δ_E. The inversion of the derivative transformation (5.11) is given by

$$\begin{bmatrix} \dfrac{\partial\varphi}{\partial x} \\ \dfrac{\partial\varphi}{\partial y} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\eta}{\partial x} \\ \dfrac{\partial\xi}{\partial y} & \dfrac{\partial\eta}{\partial y} \end{bmatrix} \begin{bmatrix} \dfrac{\partial\varphi}{\partial\xi} \\ \dfrac{\partial\varphi}{\partial\eta} \end{bmatrix}. \tag{5.14}$$

Hence, derivatives of functions, defined on Δ_k, satisfy

$$\begin{bmatrix} \dfrac{\partial\xi}{\partial x} & \dfrac{\partial\xi}{\partial y} \\ \dfrac{\partial\eta}{\partial x} & \dfrac{\partial\eta}{\partial y} \end{bmatrix} = \frac{1}{|J|} \begin{bmatrix} y_3 - y_1 & x_1 - x_3 \\ y_1 - y_2 & x_2 - x_1 \end{bmatrix}. \tag{5.15}$$

5.4 Finite element assembly

In order to obtain a generalised form of the stiffness matrix consider

$$\mathbf{A}_{ij}^{(k)} = \int_{\Delta_K} \nabla\phi_i \nabla\phi_j \, dxdy = \int_{\Delta_K} \left(\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x} + \frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y} \right) dxdy.$$

Mapping to the reference element Δ_E, we have

$$\begin{aligned} \mathbf{A}_{ij}^{(k)} = \int_{\Delta_E} \Bigg[& \left(\frac{\partial\xi}{\partial x}\frac{\partial\widehat{\varphi}_{\pi_K(i)}}{\partial\xi} + \frac{\partial\eta}{\partial x}\frac{\partial\widehat{\varphi}_{\pi_K(i)}}{\partial\eta} \right) \left(\frac{\partial\xi}{\partial x}\frac{\partial\widehat{\varphi}_{\pi_K(j)}}{\partial\xi} + \frac{\partial\eta}{\partial x}\frac{\partial\widehat{\varphi}_{\pi_K(j)}}{\partial\eta} \right) \\ & + \left(\frac{\partial\xi}{\partial y}\frac{\partial\widehat{\varphi}_{\pi_K(i)}}{\partial\xi} + \frac{\partial\eta}{\partial y}\frac{\partial\widehat{\varphi}_{\pi_K(i)}}{\partial\eta} \right) \left(\frac{\partial\xi}{\partial y}\frac{\partial\widehat{\varphi}_{\pi_K(j)}}{\partial\xi} + \frac{\partial\eta}{\partial y}\frac{\partial\widehat{\varphi}_{\pi_K(j)}}{\partial\eta} \right) \Bigg] |J| \, d\xi d\eta, \end{aligned} \tag{5.16}$$

where $\widehat{\varphi}_{\pi_K(i)}(\xi,\eta) = \phi_i\left(x(\xi,\eta), y(\xi,\eta)\right)$ are the reference basis functions and π_K is a mapping from global to local degree of freedom numbering.

Rearranging the terms, we have

$$\mathbf{A}_{ij}^{(k)} = \int_{\Delta_E} \left[\left(\frac{\partial \xi}{\partial x}^2 + \frac{\partial \xi}{\partial y}^2 \right) \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \xi} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \xi} + \left(\frac{\partial \xi}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial \xi}{\partial y} \frac{\partial \eta}{\partial y} \right) \left(\frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \xi} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \eta} + \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \eta} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \xi} \right) \right.$$
$$\left. + \left(\frac{\partial \eta}{\partial x}^2 + \frac{\partial \eta}{\partial y}^2 \right) \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \eta} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \eta} \right] |J| \, d\xi d\eta$$

$$= |J| \left(\frac{\partial \xi}{\partial x}^2 + \frac{\partial \xi}{\partial y}^2 \right) \int_{\Delta_E} \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \xi} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \xi} d\xi d\eta$$
$$+ |J| \left(\frac{\partial \xi}{\partial x} \frac{\partial \eta}{\partial x} + \frac{\partial \xi}{\partial y} \frac{\partial \eta}{\partial y} \right) \int_{\Delta_E} \left(\frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \xi} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \eta} + \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \eta} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \xi} \right) d\xi d\eta$$
$$+ |J| \left(\frac{\partial \eta}{\partial x}^2 + \frac{\partial \eta}{\partial y}^2 \right) \int_{\Delta_E} \frac{\partial \widehat{\varphi}_{\pi_K(i)}}{\partial \eta} \frac{\partial \widehat{\varphi}_{\pi_K(j)}}{\partial \eta} d\xi d\eta,$$

which is expressed using local derivatives of the element basis functions. The local stiffness matrix can be computed, because the above three integrals involve derivatives of known functions.

Finally, by assembling the $\mathbf{A}_{ij}^{(k)}$ contributions on each Δ_K the global stiffness matrix is

$$\mathbf{A} = \sum_{\Delta_k} \mathbf{A}_{ij}^{(k)}.$$

For the divergence matrix $\mathbf{B}$, the assembly is done similarly.

5.5 The right hand side approximation

Assuming the forcing term can be approximated as

$$f(x, y) \approx f_h(x, y) = \sum_i f(x_i, y_i) \phi_i,$$

the right hand side is approximated by

$$\int_{\Delta_K} f_h(x,y)\phi_j(x,y)dxdy = \int_{\Delta_K} f(x_i,y_i)\phi_i(x,y)\phi_j(x,y)dxdy.$$

Transforming to the reference element, we have

$$\int_{\Delta_K} f(x_i,y_i)\phi_i(x,y)\phi_j(x,y)dxdy = f(x_i,y_i)\int_{\Delta_E} \widehat{\varphi}_{\pi_K(i)}(\xi,\eta)\widehat{\varphi}_{\pi_K(j)}(\xi,\eta)\,|J|\,d\xi d\eta,$$

where the entries of the elemental mass matrix can be computed at the beginning of the assembly:

$$M_{ij}^{(k)} = \int_{\Delta_E} \widehat{\varphi}_{\pi_K(i)}(\xi,\eta)\widehat{\varphi}_{\pi_K(j)}(\xi,\eta)\,|J|\,d\xi d\eta.$$

Globally, from the $M_{ij}^{(k)}$ contributions on each Δ_K we have

$$M = \sum_{\Delta_k} M_{ij}^{(k)}$$

Hence

$$(f,\phi_i) \approx (M\mathbf{f})_i$$

where $\mathbf{f}_i = f(x_i,y_i)$ and M is the mass matrix.

5.6 Summary

We have illustrated how the finite element method is used to discretise the Stokes equations. We obtain a linear system of equations involving large sparse matrices. We have shown how to assemble these matrices. In the following, we use all these approaches to discretise the control problem subject to the Stokes equations in the next chapter.

Chapter 6

Solution Methods for Linear Systems

Consider the linear system

$$Kx = b, \tag{6.1}$$

with $K \in \mathbb{R}^{n \times n}$, $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{b} \in \mathbb{R}^n$. Traditionally, solutions have been determined to such systems through the use of direct solution methods, providing a solution to (6.1) within $O(n^3)$ operations. Whilst these methods (such as Gaussian elimination) are robust, from a computational perspective using direct methods for obtaining solutions to systems of the form (6.1) involving a dense system matrix of size $n \times n$ will be expensive, requiring $O(n^3)$ flops. Additionally, direct solution methods must be run to completion for a solution to be obtained. However, the matrices involved in (6.1) generally have certain structural properties which can be exploited. For instance, systems of the form (6.1) may arise from a finite element discretisation of a partial differential equation, where the matrix K will be large, but also sparse. Techniques aiming to take advantage of such structures have been considered, with the development of a number of sparse direct solution methods [29].

For this work, iterative solution methods will be used which naturally exploit sparsity patterns present in matrix-vector systems. Projection methods that look to obtain solutions to (6.1) within a subspace of $\mathbb{R}^n$ of the form

$$\mathcal{K}_m(K, \mathbf{b}) := \left\{\mathbf{b}, K\mathbf{b}, K^2\mathbf{b}, K^3\mathbf{b}, \ldots, K^{n-1}\mathbf{b}\right\} \qquad (m \leqslant n), \tag{6.2}$$

are widely used and referred to as Krylov subspace methods, with (6.2) known as a Krylov subspace. A number of approaches that aim to determine solutions to (6.1) using Krylov subspaces are present in the literature. In the case where K is symmetric positive definite, the conjugate gradient method [42] may be used. For indefinite and non-symmetric K, the GMRES (Generalised Minimum Residual) method [65] is typically used, and will be the solution method of choice within this thesis. The aim of the method is to minimise the norm of the residual $\mathbf{r} = \mathbf{b} - A\mathbf{x}$ by writing $\mathbf{x} = V\mathbf{y}$, with $\mathbf{y} \in \mathbb{R}^m$ and the matrix $V \in \mathbb{R}^{n\times m}$ denoting an orthonormal basis of $\mathcal{K}_m$. This basis is obtained through the Arnoldi process, essentially using Gram-Schmidt orthogonalisation on the Krylov space $\mathcal{K}(K, \mathbf{r}_0)$ with $\mathbf{r}_0$ representing the initial residual. The vector $\mathbf{y}$ is obtained by solving a $(m + 1) \times m$ least squares problem, which is achieved sequentially by using Givens rotations at each iterative step.

6.1 Saddle point system

Let the matrix K have the following 2×2 block structure

$$K = \begin{bmatrix} A & B^T \\ B & C \end{bmatrix}. \tag{6.3}$$

Such a matrix is referred to as a saddle point matrix, playing an important role in the fields of both Numerical Analysis and Linear Algebra (see [11]). A saddle point system is a linear system of the form (6.1), where the structure of the matrix K is as in (6.3).

Systems of this type are underlying in a number of scientific and engineering disciplines, with applications found in computational fluid dynamics [30, 31], optimisation [36, 46] and optimal control [13].

If A is nonsingular, then the saddle point matrix K admits the following block triangular factorisation

$$K = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{6.4}$$

where $S := C - BA^{-1}B^T$ is the Schur complement of A in K. From (6.4), it is clear that K is nonsingular if and only if S is nonsingular.

In a number of situations, one can expect the eigenvalues of K to be widespread and unclustered. Therefore, the use of iterative methods for saddle point systems can require a substantial number of iterations in order to achieve convergence. In order to improve the spectral properties of our system, we consider employing an appropriate preconditioner. The following system

$$P^{-1}K\mathbf{x} = P^{-1}\mathbf{b}, \tag{6.5}$$

is referred to as a left preconditioned system, where P is a non singular preconditioner matrix. Additionally, the system

$$KP^{-1}\widehat{\mathbf{x}} = \mathbf{b}, \tag{6.6}$$

is known as a right preconditioned system, where $\widehat{\mathbf{x}} := P\mathbf{x}$.

In order to choose an efficient preconditioner, P has to be a good approximation to K, taking the structure of the problem into account. Moreover, the new system involving P should be more straightforward to solve. Therefore, we require a preconditioner that is not only cheap to construct, but that also produces a matrix $P^{-1}K$ with an efficient clustering of eigenvalues.

In the literature, a number of preconditioners have been studied based on the structure of K in (6.3). Examples include block triangular preconditioners [47, 51], block definite and indefinite preconditioners [46], and block and approximate Schur complement preconditioners [11, 30].

6.2 Block preconditioners

As mentioned in the previous section, block preconditioners are based on the properties of the linear system that we want to solve, hence the block factorisation (6.4). We consider preconditioners of the form

$$P = \begin{bmatrix} I & 0 \\ c_1 BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & c_2 A^{-1}B^T \\ 0 & I \end{bmatrix}, \tag{6.7}$$

where both $c_1, c_2 \in \mathbb{R}$. By choosing appropriate values of c_1 and c_2, a number of block preconditioners can be formulated. For example, in the case where $c_1 = c_2 = 0$, (6.7) will be reduced to a block diagonal preconditioner. Alternatively, if the product $c_1c_2 = 0$, we arrive at either a block upper or lower triangular preconditioner. We will examine these preconditioners in more detail in the special case corresponding to $C = 0$.

6.2.1 Block diagonal preconditioning

The block diagonal preconditioner has the form

$$P_1 = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}$$

Proposition 6.2.1 *[51] If the preconditioner* $P_1 = \begin{bmatrix} A & 0 \\ 0 & -BA^{-1}B^T \end{bmatrix}$ *is applied to the matrix* K *as in* (6.3) *with* $C = 0$ *then the preconditioned matrix* $\mathcal{T} := P_1^{-1}K$ *satisfies the* *equation*

$$\mathcal{T}\left(\mathcal{T}-I\right)\left(\mathcal{T}^2-\mathcal{T}-I\right)=0.$$

From the above proposition, it follows that $\mathcal{T}$ is diagonalisable and has at most four distinct eigenvalues, namely $0, 1$ and $\frac{1\pm\sqrt{5}}{2}$. In the case where $\mathcal{T}$ is nonsingular, the zero eigenvalue is no longer present and so the number reduces further to three. The result from the above proposition can be applied in the case of right preconditioning, or in general any centered preconditioned matrix of the form

$$\mathcal{T}=P_1^{-1}KP_2^{-1}, \qquad \text{where} \qquad P_1P_2=P. \tag{6.8}$$

For any vector $\mathbf{r}$, the Krylov subspace

$$\mathcal{K}_n(\mathcal{T},\mathbf{r}):=\left\{\mathbf{r},\mathcal{T}\mathbf{r},\mathcal{T}^2\mathbf{r},\mathcal{T}^3\mathbf{r},\ldots,\mathcal{T}^{n-1}\mathbf{r}\right\}$$

has dimension at most 3 if $\mathcal{T}$ is nonsingular, or 4 if $\mathcal{T}$ is singular. The number of iterations for any Krylov subspace method with Galerkin or optimality property does not exceed 3 [51, 44].

6.2.2 Block triangular preconditioning

Let

$$P_2:=\begin{bmatrix} A & B^T \\ 0 & S \end{bmatrix} \qquad \text{and} \qquad P_3:=\begin{bmatrix} A & B^T \\ 0 & -S \end{bmatrix}$$

Proposition 6.2.2 *If the preconditioner P_2 is applied to K, then $P_2^{-1}K$ and KP_2^{-1} have minimum polynomial $(\lambda-1)^2$ [51].*

Proposition 6.2.3 *If the preconditioner P_3 is applied to K, then $P_3^{-1}K$ and KP_3^{-1} have minimum polynomial $(\lambda+1)(\lambda-1)$ [51].*

Remark 6.2.1 *We note the following based on the the propositions above:*

(i) By using the preconditioner P_2, the preconditioned system has only a single eigenvalue of 1*. Where as using the preconditioner P_3 will lead to a preconditioned system with exactly two eigenvalues* 1*.*

(ii) Using the preconditioner P_2 or P_3 as opposed to P_1 requires an additional multiplication of a vector by B^T at every iteration.

In order to choose a particular preconditioner, it is necessary that the preconditioner should not be expensive to construct. All previous preconditioners involve the exact Schur complement, which will generally be a dense matrix. To have a practical preconditioner, both A and S should be approximated by $\widehat{A}$ and $\widehat{S}$ respectively. The choice of these approximations depends on the underlying problem. If both are chosen in an appropriate way, the approximated preconditioned system matrix will have most of its eigenvalues clustered around those of the original preconditioned system matrix.

6.3 Schur complement approximations

For the Stokes problem, Wathen [76] describes how the Schur complement is spectrally equivalent to the pressure mass matrix M_p by using the following result: From the inf-sup stability condition, there exists a constant $\gamma > 0$ independent of the mesh size satisfying the following inequality

$$\gamma p \leqslant \sup_{\vec{u} \in V} \frac{(p \, , \nabla \cdot \vec{u})}{\nabla \cdot \vec{u}} \leqslant \mathit{\Gamma} p,$$

where $\mathit{\Gamma} \leqslant d$ for $\Omega \subset \mathbb{R}^d$. The associated discrete matrix representation to the above inequality corresponds to

$$\gamma \left(\mathbf{p}^T M_p \mathbf{p} \right)^{1/2} \leqslant \max_{\mathbf{u}} \frac{\mathbf{p}^T B \mathbf{u}}{\left(\mathbf{u}^T A \mathbf{u} \right)^{1/2}} \leqslant \mathit{\Gamma} \left(\mathbf{p}^T M_p \mathbf{p} \right)^{1/2}.$$

By setting $\mathbf{w} = A^{1/2}\mathbf{u}$, we may write

$$\max_{\mathbf{u}} \frac{\mathbf{p}^T B\mathbf{u}}{\left(\mathbf{u}^T A\mathbf{u}\right)^{-1/2}} = \max_{\mathbf{w}} \frac{\mathbf{p}^T B A^{1/2}\mathbf{w}}{\left(\mathbf{w}^T \mathbf{w}\right)^{1/2}} = \left(\mathbf{p}^T B A^{-1} B^T \mathbf{p}\right)^{1/2},$$

and thus

$$\gamma \left(\mathbf{p}^T M_p \mathbf{p}\right)^{1/2} \leqslant \left(\mathbf{p}^T B A^{-1} B^T \mathbf{p}\right)^{1/2} \leqslant \mathit{\Gamma} \left(\mathbf{p}^T M_p \mathbf{p}\right)^{1/2}.$$

In the case of variable viscosity, there is an increase in the condition number of $M_p^{-1}S$ proportional to the ratio between maximum and minimum viscosity values. This relation is described by Grinevich and Olshanski in [39] through the following lemma

Lemma 6.3.1 *Assume the discrete inf-sup condition holds, i.e.*

$$\inf_{q_h \in Q_h} \sup_{v_h \in V_h} \frac{(q_h, div\ \vec{v}_h)}{q_h \nabla \vec{v}_h} \geqslant c_0,$$

with $q_h \in Q_h$ denoting an arbitrary pressure function. Then, the following inequality holds

$$c_0^2 \nu_{max}^{-1} M_p \leqslant S \leqslant \nu_{min}^{-1} M_p.$$

Proof: See [38] or [39].

Note: For two matrices A and B we write $A \geqslant B$ if $A - B$ is positive semidefinite.

From Lemma 6.3.1, it follows that

$$\text{cond}\left(\widehat{S}^{-1} S\right) \leqslant c_0^{-2} \frac{\nu_{\max}}{\nu_{\min}} \qquad \text{with } \widehat{S} = M_p. \tag{6.9}$$

Despite the pressure mass matrix M_p being well conditioned, the lack of sparsity present within its inverse suggests that storage and application of M_p^{-1} may be expensive. Therefore, alternatives that look to provide a suitable approximation to M_p should be considered. Wathen [76] suggests forming $\widehat{S}$ via the diagonal components of M_p, i.e. $\widehat{S} =$

$\mathrm{diag}\,(M_p)$. Based on this choice, if the triangulation satisfies the condition of a limit on the minimum angle of the triangle, then $\widehat{S}$ and M_p are spectrally equivalent with constants independent of the mesh size.

However, for problems that will be discussed within this thesis, the ratio between $\nu_{\max}$ and $\nu_{\min}$ is often significantly greater than 1, suggesting that the resulting preconditioner will become inefficient. Grinevich and Olshanski extend the preconditioning of the Schur complement to the case where the viscosity coefficient is no longer constant as is the case of the generalised Stokes equations. They suggest a preconditioner which takes into account the variable viscosity. Based on this, we define our preconditioner $M_\nu = (M_\nu)_{ij} \in \mathbb{R}^{m\times m}$ as a modified pressure mass matrix, with

$$(M_\nu)_{ij} := \left(\nu^{-1}\psi_j, \psi_i\right), \tag{6.10}$$

and modified L^2 space as follows

$$L^2_\nu(\Omega) := \left\{\, q \in L^2(\Omega) \,\middle|\, \left(q, \nu^{-1}\right) = 0 \right\}.$$

We now consider the effectiveness of the preconditioner M_ν. We are interested in the constants c_ν and C_ν in the following inequality

$$c_\nu M_\nu \leqslant S \leqslant C_\nu M_\nu \qquad \text{in the space } \; Q_h. \tag{6.11}$$

From Lemma 6.3.1, it is possible to derive an evaluation of c_ν, which is formulated in the following lemma

Lemma 6.3.2

$$c_\nu \geqslant c_0^2 \frac{\nu_{min}}{\nu_{max}},$$

where c_0 is the inf-sup constant.

Proof: See [38] or [39]

Lemma 6.3.3 *For a positive $\nu \in L^\infty(\Omega)$ and $\Omega \subset \mathbb{R}^d$, the upper bound in* (6.11) *holds with $C_\nu = d$.*

Proof: See [38] or [39].
This bound on c_ν leads to a similar estimate on the condition number of $M_\nu^{-1}S$ as seen in (6.9). However, numerical experimentation presented in later chapters will show that for particular coefficients ν in non-Newtonian flow, the effective condition number of M_ν^{-1} is bounded with respect to $\nu_{\min}\nu_{\max}^{-1}$.

It is known that the eigenvalues which are smallest in magnitude slow down the convergence of GMRES. Removing or deflating these isolated eigenvalues can enhance the convergence rate. In the next section, we will present a particular deflated preconditioner to do this job.

6.4 Deflated preconditioner

In [5], an adaptive preconditioner is presented for restarted GMRES. This adaptive preconditioner is attractive when a good preconditioner is already applied. The aim of this adaptive preconditioner is to remove the smallest k eigenvalues and replace them by 1. The eigenvalues of the preconditioned matrix will have k multiple eigenvalues equal to 1, as well as the remaining $n-k$ eigenvalues of the original matrix. The latter paper builds a preconditioner from spectral information gathered by the Arnoldi process after each iteration of the restarted GMRES algorithm. This method is based on determining an invariant subspace of the original matrix and shifting the associated eigenvalues that are close to the origin. This adaptive preconditioner essentially enhances the performance of the iterative solver by removing the influence of smaller eigenvalues [5]. To construct the deflated preconditioner, we need the information from the Arnoldi decomposition of an $n \times n$ matrix K. At each iterative step m, we have the relation

$$KV_m = V_m H_m + \mathbf{f}_m \mathbf{e}_m^T,$$

where $\mathbf{f}_m \in \mathbb{R}^n$ and $V_m \in \mathbb{R}^{n \times m}$ such that $V_m^T V_m = I_m$ and $V^T \mathbf{f}_m = 0$. The $\mathbf{e}_m$ is the m^{th} axis vector of appropriate dimension, I_m denotes the $m \times m$ identity matrix and $H_m \in \mathbb{R}^{m \times m}$ is an upper Hessenberg matrix. When $V_m \mathbf{e}_1 = \mathbf{r}_0 / \mathbf{r}_0$, the columns of V_m span the Krylov subspace $\mathcal{K}_m (K, \mathbf{r}_0)$. Let $V_k \in \mathbb{R}^{n \times k}$ be the matrix which consists of the first k columns $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ of V_m, and let the columns of the matrix W_{n-k} span the orthogonal complement of span $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$. As $W_{n-k}^T W_{n-k} = I_{n-k}$, the columns of the matrix $[V_k \; W_{n-k}]$ form an orthogonal basis of $\mathbb{R}^n$. The inverse of the matrix

$$M_{def} = V_k H_k V_k^T + W_{n-k} W_{n-k}^T, \tag{6.12}$$

will be used as a deflated preconditioner, with $k << n$. We describe this inverse below [5].

Proposition 6.4.1 *Let $Q \in \mathbb{R}^{n \times n}$ be an orthogonal matrix partitioned as $Q = [V \; W]$, where the submatrix V consists of the first k columns of Q and the submatrix W the remaining columns. Assume that $H = V^T K V$ is nonsingular. Then, the matrix*

$$M_{def} := VHV^T + WW^T,$$

is nonsingular, with inverse given by

$$M_{def}^{-1} = VH^{-1}V^T + WW^T.$$

Proof: The matrix M_{def} can be expressed as

$$M_{\text{def}} = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} H & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix}, \tag{6.13}$$

and therefore

$$M_{\text{def}}^{-1} = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} H^{-1} & 0 \\ 0 & I_{n-k} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix}, \tag{6.14}$$

which shows the result

$$M_{\text{def}}^{-1} = VH^{-1}V^T + WW^T.$$

□

In Proposition 6.4.1, when the span of the columns of the matrix

$$V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\},$$

represent an invariant subspace of $\mathcal{K}$, the eigenvalues of $M_{def}^{-1}K$ can be found in terms of the eigenvalues of K. This relation is expressed in the following corollary

Corollary 6.4.1 *Let the matrices V, W and H be as in Proposition 6.4.1, and assume, moreover, that the columns of the matrix V span an invariant subspace of $\mathcal{K}$ associated with the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_k$. Then*

$$\lambda\left(M_{def}^{-1}K\right) = \{\,\lambda_{k+1}, \lambda_{k+2}, \ldots, \lambda_n, 1, \ldots, 1\,\},$$

where the eigenvalue 1 has multiplicity at least k.

Proof: The matrix K is similar to

$$\widetilde{K} = \begin{bmatrix} V^T \\ W^T \end{bmatrix} K \begin{bmatrix} V & W \end{bmatrix} = \begin{bmatrix} H & \widetilde{K}_{12} \\ 0 & \widetilde{K}_{22} \end{bmatrix},$$

and

$$\lambda\left(\widetilde{K}_{22}\right) = \{\lambda_{k+1}, \lambda_{k+2}, \dots, \lambda_n\}\,. \tag{6.15}$$

From (6.15) and the representation of M_{def}^{-1} in (6.14) we get

$$M_{\text{def}}^{-1}K = \begin{bmatrix} V & W \end{bmatrix} \begin{bmatrix} I_k & H^{-1}\widetilde{K}_{12} \\ 0 & \widetilde{K}_{22} \end{bmatrix} \begin{bmatrix} V^T \\ W^T \end{bmatrix}.$$

Then the spectrum of the matrix $M_{\text{def}}^{-1}K$ consists of $\lambda\left(\widetilde{K}_{22}\right)$ and at least k eigenvalues equal to 1. □

In the case of right preconditioning, the result is shown in [33].

6.5 Summary

Within this chapter, a concise background into solution methods for linear systems has been provided, with a description of the iterative approach we plan to use throughout this thesis, namely GMRES. Furthermore, we have discussed appropriate preconditioning strategies for saddle point problems based on the system arising from use of the mixed finite element method for the Stokes equations described in the previous chapter. Both block diagonal and block upper triangular preconditioners have been considered, with relevant results from the literature analysed. Finally, we have introduced the notion of deflated preconditioning in order to enhance our solution.

Chapter 7

Generalised Stokes equations

Newtonian fluids have a linear relation between the shear stress and the shear rate. In contrast, in non Newtonian fluid, this relation is not linear any more. There are many models used in practice to describe the extra stress tensor $\boldsymbol{\sigma}_E = \nu(\varepsilon(\vec{u}))\varepsilon(\vec{u})$ (4.2). Depending on the form of the function $\nu(.)$, these fluids can be subdivided into three different types: pseudoplastic (or shear-thinning), dilatant (shear- thickening) and viscoplastic. Some of the better known models are listed below.

7.1 Constitutive models

1. **The power law (Ostwald-deWaele)**

   One of the most widely used is the power law model, where $\boldsymbol{\sigma}_E$ is written as following:

   $$\nu = 2\nu_0 \left|\varepsilon(\vec{u})\right|^{\alpha-1}$$

   where

   $$|\varepsilon(\vec{u})| = \left(\sum_{i,j=1}^{2} (\varepsilon_{i,j}(\vec{u}))^2\right)^{\frac{1}{2}},$$

   ν_0 and $\alpha > 0$ are parameters characteristic of each fluid. Here, α represents the flow behaviour index and $\nu_0 > 0$ is the consistency index.

For $\alpha = 1$, the power law model describes the Newtonian fluid which we presented in Chapter 4 and non Newtonian for $\alpha \neq 1$. When $0 < \alpha < 1$, the model exhibits a decrease in viscosity with increasing shear stress which describes the shear-thinning fluids. When $1 < \alpha < \infty$ the model exhibits an increase in viscosity with increasing shear stress which describes the shear-thickening fluids.

2. **Carreau model**

   Another popular model is that due to Carreau [19]. The constitutive equation for the Carreau model is given via the following expression for the viscosity

$$\frac{\nu - \nu_\infty}{\nu_0 - \nu_\infty} = [1 + |\varepsilon(\vec{u})|^2]^{\frac{\alpha-1}{2}},$$

   where ν_∞ is the infinite shear rate viscosity, ν_0 is the zero shear rate viscosity and α the exponential constant. When $\alpha = 1$, the Carreau model corresponds to Newtonian fluid with $\nu = \nu_0$.

3. **Cross model**

   Another model with a wide acceptance is due to Cross [26]. In simple shear, the model is written as:

$$\frac{\nu - \nu_\infty}{\nu_0 - \nu_\infty} = \frac{1}{1 + k\,|\varepsilon(\vec{u})|^\alpha},$$

   where $\alpha < 1$ and k are fitting parameters. ν_∞ and ν_0 are the limiting values of the apparent viscosity at high and low shear rate, respectively. As $k \to 0$, this model describe the Newtonian fluid behaviour. Similarly, the Cross model reduces to the power law model when $\nu \ll \nu_0$ and $\nu \gg \nu_\infty$.

More descriptions of such models can be found in many books [20], [24], [53], [40] and in the review paper [14].

7.2 Generalised Stokes equations

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain with boundary $\partial\Omega$. Recall the steady-state generalised Stokes equations for incompressible fluids are given by the system of partial differential equations

$$\begin{aligned} -\operatorname{div}\boldsymbol{\sigma} &= \vec{f} \quad \text{in } \Omega, && (7.1a)\\ \operatorname{div}\vec{u} &= 0 \quad \text{in } \Omega, && (7.1b)\\ \vec{u} &= \vec{u}_D \quad \text{on } \partial\Omega_D, && (7.1c)\\ \mathbf{n}\cdot\boldsymbol{\sigma} &= 0 \quad \text{on } \partial\Omega_N, && (7.1d) \end{aligned}$$

where the stress tensor

$$\boldsymbol{\sigma} = -pI + \boldsymbol{\sigma}_E,$$

the extra stress tensor $\boldsymbol{\sigma}_E$ is a function of shear rate $\varepsilon(\vec{u})$,

$$\boldsymbol{\sigma}_E = \nu(\varepsilon(\vec{u}))\varepsilon(\vec{u}),$$

and

$$\varepsilon(\vec{u}) = \frac{1}{2}\left(\nabla\vec{u} + (\nabla\vec{u})^T\right),$$

$\vec{f} : \Omega \to \mathbb{R}^2$ is a given function, $\vec{u}$ is the velocity of the fluid, and p is the pressure.

7.3 Weak formulation

First, define the following spaces

$$\begin{aligned} V_E &= \left\{ \vec{w} \in (H^1(\Omega))^2 : \vec{w}_{|\partial\Omega_D} = \vec{u}_D \right\}, \\ V_0 &= \left\{ \vec{w} \in (H^1(\Omega))^2 : \vec{w}_{|\partial\Omega_D} = \mathbf{0} \right\}, \\ Q &= L^2(\Omega). \end{aligned}$$

We use the same approach as in Chapter 4 to derive the weak formulation by multiplying (7.1a) and (7.1b) by arbitrary functions $\vec{v} \in V_0, q \in Q$, respectively, and integrating by part to the left hand side. We get

$$-\int_{\partial\Omega} \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \vec{v}\ ds + \int_{\Omega} \boldsymbol{\sigma} : \nabla\vec{u}\ d\Omega = \int_{\Omega} \vec{f} \cdot \vec{v}\ d\Omega \tag{7.2a}$$

$$\int_{\Omega} q \text{ div } \vec{u}\ d\Omega = 0. \tag{7.2b}$$

In equation (7.2a), the term $\int_{\partial\Omega} \mathbf{n} \cdot \boldsymbol{\sigma} \cdot \vec{v}\ ds = 0$ since $\vec{v} = 0$ on $\partial\Omega_D$ and $\mathbf{n} \cdot \boldsymbol{\sigma} = 0$ on $\partial\Omega_N$. By substituting $\boldsymbol{\sigma}$ in the first equation we can state the following weak (or variational) formulation of the generalised Stokes problem (7.1):

$$\left\{ \begin{array}{l} \text{Find } (\vec{u}, p) \in V_E \times Q \text{ such that for all } (\vec{v}, q) \in V_0 \times Q, \\ \displaystyle\int_{\Omega} \nu(\varepsilon(\vec{u}))\varepsilon(\vec{u}) : \varepsilon(\vec{v}) \ \ d\Omega - \int_{\Omega} p \text{ div}\,\vec{v} \ \ d\Omega = \int_{\Omega} \vec{f} \cdot \vec{v} \ \ d\Omega, \\ \displaystyle\int_{\Omega} q \text{ div}\,\vec{u} \ \ d\Omega = 0. \end{array} \right. \tag{7.3}$$

Existence and uniqueness for generalised Stokes equations (7.1) was shown by Baranger and Najibin [6] for both the power law and Carreau models. A particular finite element discretisation is discussed in [8]. For simplicity of presentation we consider that $\vec{u}_D = \mathbf{0}$ so that $V_E = V_0$.

Note that problem (7.3) is nonlinear, due to the dependence of ν on $\vec{u}$. We need first to linearize the problem. We choose the following Picard iteration: given $(\vec{u}^0, p^0)$ as an initial guess, solve the following sequence of linear problems until convergence

$$\begin{cases} \textbf{For } m = 0, 1, \ldots \text{ solve until convergence} \\ \text{Find } (\vec{u}^{m+1}, p^{m+1}) \in V_0 \times Q \text{ such that for all } (\vec{v}, q) \in V_0 \times Q, \\ \displaystyle\int_\Omega \nu(\varepsilon(\vec{u}^m))\varepsilon(\vec{u}^{m+1}) : \varepsilon(\vec{v}) d\Omega - \int_\Omega p^{m+1} \mathrm{div}\, \vec{v}\; d\Omega = \int_\Omega \vec{f} \cdot \vec{v}\; d\Omega. \\ \displaystyle\int_\Omega q\ \mathrm{div}\, \vec{u}^{m+1}\ d\Omega = 0 \\ \textbf{End} \end{cases} \tag{7.4}$$

We will use the finite element method to solve the above sequence of linear problems. Let Ω_h be a partition of Ω into simplices of diameter no greater than h. Let $V_0^h \in V_0$ and $Q^h \in Q$ be finite dimensional subspaces, with bases $\{\vec{\phi}_i\}$, $\{\psi_j\}$ of continuous piecewise polynomial functions, respectively, defined on the partition Ω_h. We consider the following discrete weak formulation corresponding to formulation (7.3)

$$\begin{cases} \text{Find } (\vec{u}_h, p_h) \in V_0^h \times Q^h \text{ such that for all } (\vec{v}_h, q_h) \in V_0^h \times Q^h, \\ \displaystyle\int_\Omega \nu(\varepsilon(\vec{w}_h))\varepsilon(\vec{u}_h) : \varepsilon(\vec{v}_h)\ d\Omega - \int_\Omega p_h \mathrm{div}\, \vec{v}_h\ d\Omega = \int_\Omega \vec{f} \cdot \vec{v}_h\ d\Omega. \\ \displaystyle\int_\Omega q_h\ \mathrm{div}\, \vec{u}_h\ d\Omega = 0. \end{cases} \tag{7.5}$$

Using the expansions

$$\vec{u}_h(\mathbf{x}) = \sum_{i=1}^{N} u_i \vec{\phi}_i(\mathbf{x}), \quad p_h(\mathbf{x}) = \sum_{j=1}^{M} p_j \psi_j(\mathbf{x})$$

we obtain the discrete form of the Picard iteration (7.4):

$$\begin{cases} \textbf{For } m = 0, 1, \dots \text{ solve until convergence} \\ \begin{pmatrix} \mathbf{A}_m(\vec{u}) & \mathbf{B}^T \\ \mathbf{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}^{m+1} \\ \mathbf{p}^{m+1} \end{pmatrix} = \begin{pmatrix} \vec{f} \\ 0 \end{pmatrix}. \\ \textbf{End} \end{cases} \tag{7.6}$$

where

$$[\mathbf{A}_m(\vec{u}_h)]_{ij} = \int_\Omega \nu(\varepsilon(\vec{u}_h^m))\, \varepsilon(\vec{\phi}_j) : \varepsilon(\vec{\phi}_i)\ d\Omega,$$

$$[\mathbf{B}]_{jk} = \int_\Omega \psi_j \operatorname{div} \vec{\phi}_k \ \ d\Omega$$

and

$$f_k = \int_\Omega \vec{f} \cdot \vec{\phi}_k \ \ d\Omega.$$

In each Picard iteration, $\nu(\cdot)$ is fixed. Then every iteration represents a system where we can treat it as a standard linear Stokes problem.

7.4 Iterative solution of the discrete linearised Stokes equations

As we described our method in Chapter 6, we are interested in solving the linear system (7.6) by a preconditioned GMRES iterative method with the following block triangular preconditioners

$$P := \begin{bmatrix} \mathbf{A}(\vec{u}) & \mathbf{B}^T \\ 0 & M_p \end{bmatrix},$$

$$P_\nu := \begin{bmatrix} \mathbf{A}(\vec{u}) & \mathbf{B}^T \\ 0 & M_\nu \end{bmatrix},$$

where M_ν is the weighted mass matrix defined in (6.10). Olishanki and Simonchini [52] studied the spectral properties of preconditioned saddle point systems. Their work involved use of a deflated MINRES method coupled with a block diagonal preconditioning strategy in mitigate the effects of outlying eigenvalues. In [78], theoretical results for a GMRES approach coupled with a deflated preconditioner based on an exactly A-invariant subspace is presented.

7.5 Numerical examples

Here we present some numerical experiments for the Stokes equations. We start with the case of piecewise constant viscosity then the general case of variable viscosity.

7.5.1 Piecewise constant viscosity

In this section, we will present numerical results in the case of piecewise constant viscosity before turning our attention to the case of variable viscosity. Here, our findings will illustrate the effectiveness of the deflated preconditioner presented in (6.12). We will consider two examples, firstly involving one jump, and secondly involving two jumps in the viscosity.

Example 3 ***(1 Jump)***
Let $\Omega = [0,1]^2$ and consider the Stokes problem

$$\begin{cases} -\nu\Delta\vec{u} + \nabla p &= \vec{f} \qquad \textit{in } \Omega, \\ \nabla\cdot\vec{u} &= 0 \qquad \textit{in } \Omega, \\ \vec{u} &= \vec{u}_D \quad \textit{on } \partial\Omega, \end{cases} \tag{7.7}$$

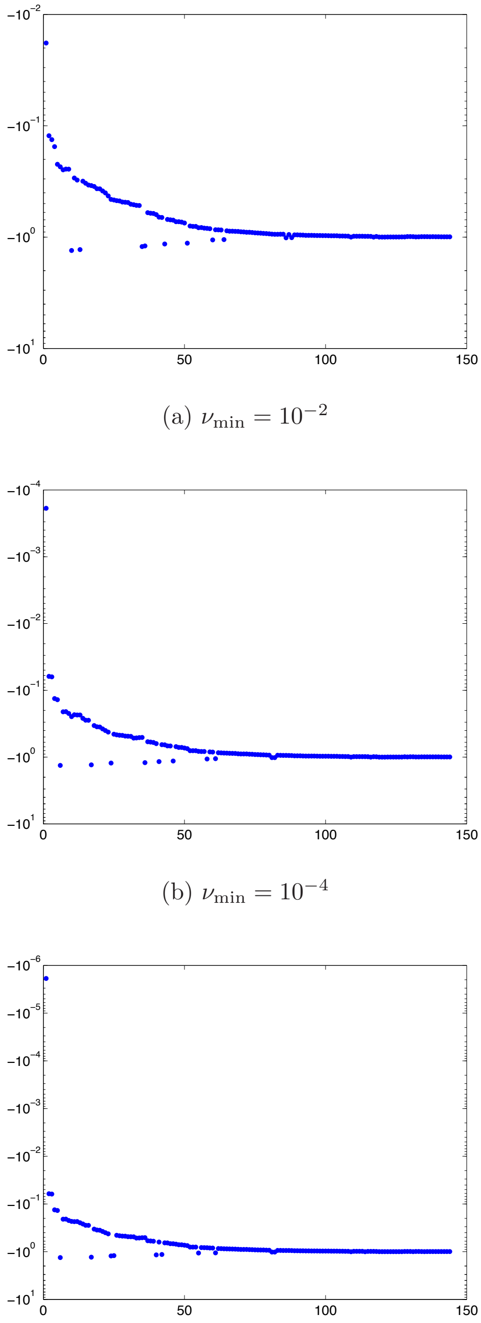
where

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 16(x-x^2)^2 \\ 0 \end{pmatrix} & \textit{on } y=1 \\ \mathbf{0} & \textit{otherwise.} \end{cases}$$

We divide the domain into two equal parts, and set the viscosity to be the following function

$$\nu = \begin{cases} 1, & 0 \leqslant x \leqslant \dfrac{1}{2}, \\ \nu_{min}, & \dfrac{1}{2} < x \leqslant 1. \end{cases}$$

Table 7.1 shows the three minimal nonzero and maximal eigenvalues of S, $M_p^{-1}S$ and $M_\nu^{-1}S$ for various different values of $\nu_{\min}$. Figure 7.1 shows the spectrum of $M_\nu^{-1}S$ for different values of $\nu_{\min}$. As can be seen in Figure 7.1, the spectrum of $M_\nu^{-1}S$ is essentially clustered in terms of its order, with few eigenvalues of small order. From Table 7.1, it is readily seen that there is a single eigenvalue of order $\nu_{\min}$ separated away from the rest of the cluster, with this gap increasing as $\nu_{\min}$ is decreased further. The eigenvalues with the smallest magnitude will not only influence the conditioning of the matrix, but also the performance of iterative solution methods. If we want to reduce the condition number and increase the performance of GMRES, we need to cancel the effect of the smallest eigenvalues. In this example, we have to remove one eigenvalue. Table 7.2 shows the number of GMRES iterations for different preconditioners for a variety of different values of $\nu_{\min}$. The table shows that there is GMRES iterations improvement with each single preconditioning that we applied.



(a) $\nu_{\min} = 10^{-2}$

(b) $\nu_{\min} = 10^{-4}$

(c) $\nu_{\min} = 10^{-6}$

Figure 7.1: The spectrum of $M_\nu^{-1}S$ for different values of $\nu_{\min}$ for one jump viscosity.

	Minimum Eigenvalues			Maximum Eigenvalue		
$\nu_{\min}$	S	$M_p^{-1}S$	$M_\nu^{-1}S$	S	$M_p^{-1}S$	$M_\nu^{-1}S$
10^{-1}	7.08×10^{-4}	2.21×10^{-1}	1.30×10^{-1}	1.72×10^{-1}	9.99	1.25
	7.11×10^{-4}	2.29×10^{-1}	1.85×10^{-1}			
	1.31×10^{-3}	3.49×10^{-1}	2.21×10^{-1}			
10^{-2}	7.08×10^{-4}	2.22×10^{-1}	1.81×10^{-2}	1.23	99.87	1.32
	7.11×10^{-4}	2.29×10^{-1}	1.23×10^{-1}			
	1.31×10^{-3}	3.51×10^{-1}	1.33×10^{-1}			
10^{-3}	7.08×10^{-4}	2.22×10^{-1}	1.87×10^{-3}	12.27	9.99×10^{2}	1.33
	7.11×10^{-4}	2.29×10^{-1}	6.90×10^{-2}			
	1.31×10^{-3}	3.51×10^{-1}	7.03×10^{-2}			
10^{-4}	7.08×10^{-4}	2.22×10^{-1}	1.88×10^{-4}	1.23×10^{2}	9.99×10^{3}	1.33
	7.11×10^{-4}	2.29×10^{-1}	6.18×10^{-2}			
	1.31×10^{-3}	3.51×10^{-1}	6.28×10^{-2}			
10^{-5}	7.08×10^{-4}	2.22×10^{-1}	1.88×10^{-5}	1.23×10^{3}	9.99×10^{4}	1.33
	7.11×10^{-4}	2.29×10^{-1}	6.11×10^{-1}			
	1.31×10^{-3}	3.51×10^{-1}	6.21×10^{-2}			
10^{-6}	7.08×10^{-4}	2.22×10^{-1}	1.88×10^{-6}	1.23×10^{4}	9.99×10^{5}	1.33
	7.11×10^{-4}	2.29×10^{-1}	6.10×10^{-2}			
	1.31×10^{-3}	3.51×10^{-1}	6.20×10^{-2}			

Table 7.1: Minimum and (three) maximum eigenvalues for Example 3 involving a single jump in the viscosity. Results are displayed for varying $\nu_{\min}$ values with different choices of preconditioner.

$\nu_{\min}$	P	P_ν	$M_{\text{def}}P_\nu$
10^{-1}	30	15	15
10^{-2}	68	22	18
10^{-3}	82	30	21
10^{-4}	87	33	21
10^{-5}	89	36	21
10^{-6}	89	38	23

Table 7.2: GMRES iterations in the case of one jump for different preconditioners.

Example 4 ***(2 Jumps)***

Let $\Omega = [0,1]^2$ *and consider the problem*

$$\begin{cases} -\nu\Delta\vec{u} + \nabla p &= \vec{f} \qquad \textit{in } \Omega, \\ \nabla \cdot \vec{u} &= 0 \qquad \textit{in } \Omega, \\ \vec{u} &= \vec{u}_D \quad \textit{on } \partial\Omega, \end{cases} \tag{7.9}$$

where

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 16(x-x^2)^2 \\ 0 \end{pmatrix} & \textit{on } y = 1, \\ \mathbf{0} & \textit{otherwise.} \end{cases}$$

In this example, the domain is divided into three equal parts with two jumps in the viscosity as follows

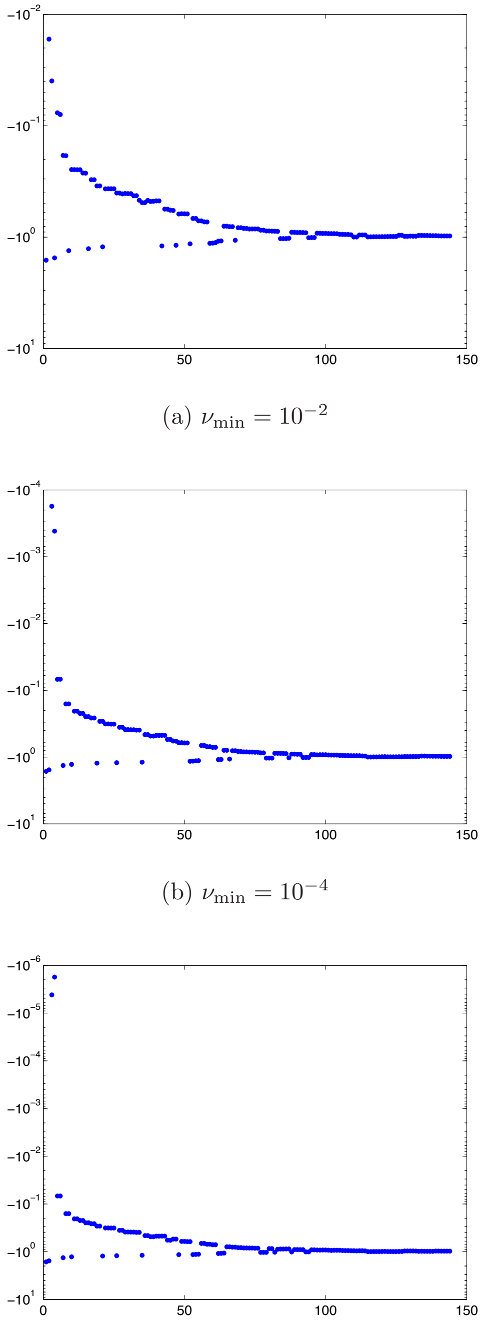
$$\nu = \begin{cases} \nu_{min}, & 0 \leqslant x \leqslant \frac{1}{3}, \\ 1, & \frac{1}{3} < x < \frac{2}{3}, \\ \nu_{min}, & \frac{2}{3} \leqslant x \leqslant 1. \end{cases}$$

Table 7.3 shows the three minimal nonzero and maximal eigenvalues of S, $M_p^{-1}S$ and $M_\nu^{-1}S$ for various different values of $\nu_{\min}$. Figure 7.2 shows the spectrum of $M_\nu^{-1}S$ for different values of $\nu_{\min}$. By direct comparison to the one jump case, we see in both examples particular eigenvalues that have order $\nu_{\min}$ lying far away from the rest of the spectrum. In this example, there are two eigenvalues that need to be deflated. Table 7.4 shows the number of GMRES iterations with different preconditioners for a variety of different values of $\nu_{\min}$. The table shows that there is improvement with each single preconditioning that we applied. Tables 7.2 and 7.4 highlight the effect of removal of just

	Minimum Eigenvalues			Max Eigenvalue		
$\nu_{\min}$	S	$M_p^{-1}S$	$M_\nu^{-1}S$	S	$M_p^{-1}S$	$M_\nu^{-1}S$
10^{-1}	1.82×10^{-3}	5.61×10^{-1}	1.13×10^{-1}	1.72×10^{-1}	9.97	1.44
	1.83×10^{-3}	5.69×10^{-1}	1.26×10^{-1}			
	1.83×10^{-3}	6.84×10^{-1}	1.39×10^{-1}			
10^{-2}	1.82×10^{-3}	5.81×10^{-1}	1.67×10^{-2}	1.27	99.65	1.61
	1.92×10^{-3}	5.99×10^{-1}	3.95×10^{-2}			
	1.92×10^{-3}	7.38×10^{-1}	7.65×10^{-2}			
10^{-3}	1.82×10^{-3}	5.85×10^{-1}	1.75×10^{-3}	11.24	9.96×10^{2}	1.63
	1.93×10^{-3}	6.05×10^{-1}	4.12×10^{-3}			
	1.93×10^{-3}	7.44×10^{-1}	6.91×10^{-2}			
10^{-4}	1.82×10^{-3}	5.85×10^{-1}	1.75×10^{-4}	1.12×10^{2}	9.96×10^{3}	1.64
	1.93×10^{-3}	6.05×10^{-1}	4.14×10^{-4}			
	1.93×10^{-3}	7.45×10^{-1}	6.83×10^{-2}			
10^{-5}	1.82×10^{-3}	5.85×10^{-1}	1.76×10^{-5}	1.12×10^{3}	9.96×10^{4}	1.64
	1.93×10^{-3}	6.05×10^{-1}	4.14×10^{-5}			
	1.94×10^{-3}	7.45×10^{-1}	6.82×10^{-2}			
10^{-6}	1.82×10^{-3}	5.85×10^{-1}	1.76×10^{-6}	1.12×10^{4}	9.96×10^{5}	1.64
	1.93×10^{-3}	6.05×10^{-1}	4.14×10^{-6}			
	1.94×10^{-3}	7.45×10^{-1}	6.82×10^{-2}			

Table 7.3: Three minimum and maximum eigenvalues for Example 4 involving a single jump in the viscosity. Results are displayed for varying $\nu_{\min}$ values with different choices of preconditioner.

one or two eigenvalues for one or two jumps in the viscosity. We will use the advantages seen here from application of the deflated preconditioner in the more general case of variable viscosity.



(a) $\nu_{\min} = 10^{-2}$

(b) $\nu_{\min} = 10^{-4}$

(c) $\nu_{\min} = 10^{-6}$

Figure 7.2: The spectrum of $M_\nu^{-1} S$ for different values of $\nu_{\min}$ for two jumps viscosity.

$\nu_{\min}$	P	P_ν	$M_{\mathrm{def}}P_\nu$
10^{-1}	19	14	12
10^{-2}	46	19	16
10^{-3}	57	24	18
10^{-4}	68	27	20
10^{-5}	89	30	20
10^{-6}	98	32	22

Table 7.4: GMRES iterations in the case of two jumps for different preconditioners.

7.5.2 Variable viscosity

In this section, we will consider the same two examples that presented in Chapter 4 but with variable viscosity model this time. In this thesis, we only consider the power law model.

Example 5 ***(Driven cavity test problem)***
Let $\Omega = [0,1]^2$. Consider the Stokes problem

$$\begin{aligned} -div\boldsymbol{\sigma} &= \vec{f} \quad in\ \Omega, \\ div\,\vec{u} &= 0 \quad in\ \Omega, \\ \vec{u} &= \vec{u}_D \quad on\ \partial\Omega, \end{aligned}$$

where the stress tensor

$$\boldsymbol{\sigma} = -pI + 2\nu_0 \left|\varepsilon(\vec{u})\right|^{\alpha-1} \varepsilon(\vec{u}).$$

The Dirichlet data on the boundary is given by

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 16(x-x^2)^2 \\ 0 \end{pmatrix} & on\ y = 1, \\ \mathbf{0} & otherwise. \end{cases}$$

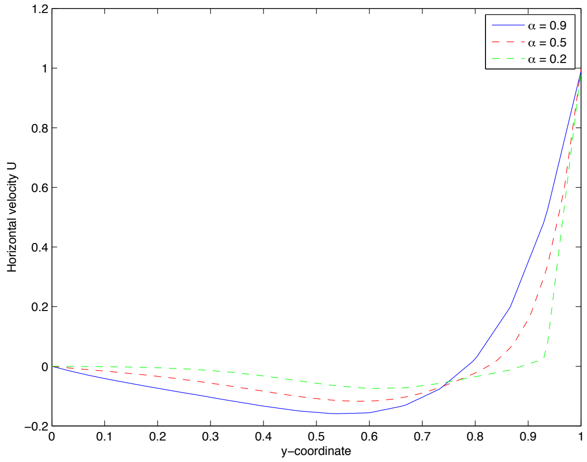


Figure 7.3: The velocity profiles of the discrete solution at $(\frac{1}{2}, y)$ for driven cavity test problem.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
15×15	7	9	12	15	21	27	48	89
31×31	7	9	12	16	20	27	41	86
63×63	7	9	12	16	21	27	40	128

Table 7.5: Total number of Picard iterations for driven cavity test problem.

The velocity profiles of the discrete solution at $(\frac{1}{2}, y)$ are shown in Figure 7.3 for varying α, with an indication of the number of Picard iterations required to achieve convergence shown in Figure 7.4. Table 7.5 shows the total number of the Picard iterations for solving Example 5 directly with different values of α and grid meshes. Our initial guess being the solution of the linear Stokes problem. The stopping tolerance for the Picard iterations is chosen to be 10^{-6}. As we can see in Table 7.5, the number of Picard iterations increase as α is close to the zero. This behaviour is relevant because as $\alpha \to 0$, the fluid viscosity becomes more nonlinear in behaviour. We use GMRES to solve the system in Example 5 with the preconditioner P_ν. Table 7.6 shows the number of Picard iterations (outer iterations) and average number of GMRES iterations (inner iterations) per outer iteration

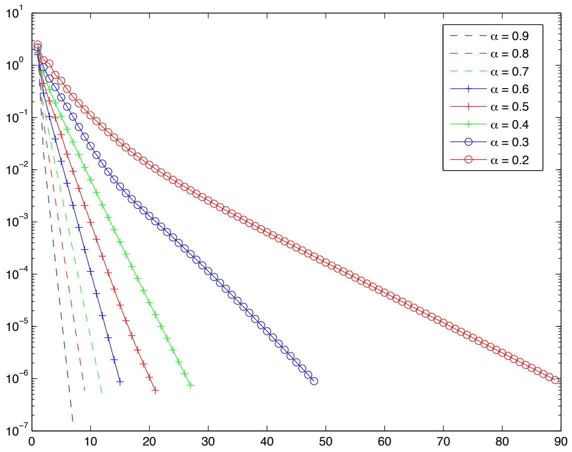


Figure 7.4: The convergence history of Picard iterations for different values of α for driven cavity test problem.

for different values of α and grid meshes. The stopping tolerance for the inner iterations depend on the residual of the system in each Picard iteration ($residual$). As $\alpha \to 0$, we need to tighten the inner tolerance, otherwise the outer iterations do not converge. In Table 7.6, it is clear that for $\alpha = 0.2$, the GMRES iterations deteriorate since the inner tolerance is tightened to $10^{-6}(residual)^{0.9}$. Therefore, we will use the deflated preconditioner (6.12) to enhance our GMRES iterations number for this particular α.

The Figure 7.5 shows the GMRES convergence profiles when solving Example 5 with preconditioner P_ν. The deflation M_{def} is started from k^* for driven cavity test problem for grid meshes 15×15 and 31×31. As the number of GMRES iterations increases, so too does the size of the Krylov subspace where a solution is sought. This means that more spectral information will be available, indicating that deflation will work better when used in later stages of GMRES as opposed to earlier on in the iterative process.

The current outer residual value can be used as an alternative indicator of when best to apply deflation to GMRES. For instance, one could start to deflate when the Picard residual is smaller than 10^{-2}, or 10^{-3}, for instance. For the control problem considered in

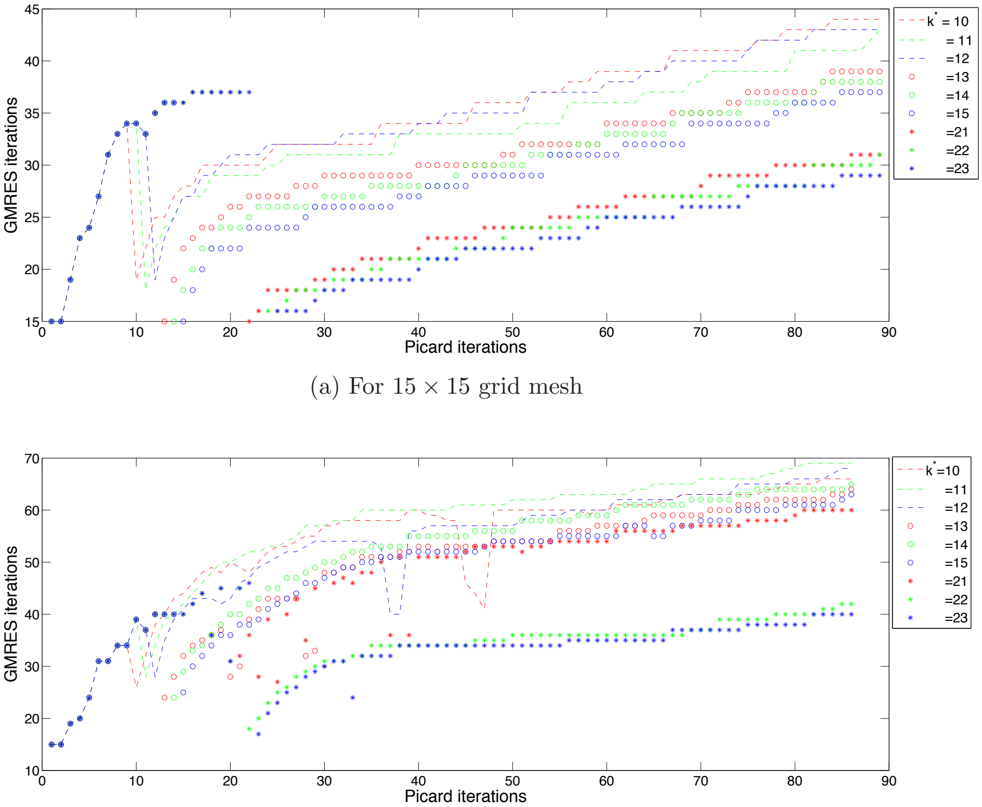
$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
15×15	7 (5)	9 (5)	12 (4)	15 (5)	21 (11)	27 (6)	47 (11)	89 (38)
31×31	7 (4)	9 (4)	12 (4)	16 (4)	20 (11)	28 (4)	40 (14)	86 (47)
63×63	7 (4)	9 (4)	12 (3)	16 (3)	21 (10)	28 (4)	43 (15)	130 (37)

Table 7.6: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 5 with preconditioner P_ν for different values of α and grid meshes.

residual=	10^{-2}	10^{-3}	10^{-4}
15×15	89 (26)	89 (26)	89 (30)
31×31	86 (35)	86 (32)	86 (33)
63×63	129 (48)	129 (39)	130 (42)

Table 7.7: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 5 with preconditioners $M_{\text{def}}P_\nu$ for $\alpha = 0.2$ and different grid meshes. The top row indicates the outer residual value at which deflation is considered.

this work, we will consider the application of deflation whenever the current outer residual is less than 10^{-3}. Table 7.7 shows the number of Picard iterations (outer iterations) and average number of GMRES iterations (inner iterations) per outer iteration for $\alpha = 0.2$ under different grid meshes after using this deflated preconditioner. Deflation is considered at the *residual* values given in the table. From the table, the minimum GMRES iterations were noted whenever the Picard residual was less than 10^{-3}. These observations will be used later on in the thesis, when deflation is considered for the Stokes control problem.



(a) For 15 × 15 grid mesh

(b) For 31 × 31 grid mesh

Figure 7.5: The GMRES convergence profiles when solving Example 5 with preconditioner P_ν. The deflation M_{def} is started from k^* for driven cavity test problem for grid meshes 15 × 15 and 31 × 31.

Example 6 ***(Pipe flow test problem)***

Let $\Omega = [0,4] \times [0,1]$*. Consider the Stokes problem*

$$\begin{aligned} -div\boldsymbol{\sigma} &= \vec{f} \quad in\ \Omega, \\ div\,\vec{u} &= 0 \quad in\ \Omega, \\ \vec{u} &= \vec{u}_D \quad on\ \partial\Omega_D \\ \mathbf{n}\cdot\boldsymbol{\sigma} &= 0 \quad on\ \partial\Omega_N, \end{aligned}$$

where

$$\boldsymbol{\sigma} = -pI + 2\nu_0 \left|\varepsilon(\vec{u})\right|^{\alpha-1} \varepsilon(\vec{u}).$$

The Dirichlet data on the boundary is given by

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 4(y-y^2) \\ 0 \end{pmatrix} & on\ x = 0 \\ \mathbf{0} & on\ y = 0,\ y = 1. \end{cases}$$

The velocity profiles of the discrete solution at $(\frac{1}{2}, y)$ are shown in Figure 7.6 for varying α, Figure 7.7 shows the convergence history of Picard iterations for different values of α. Table 7.8 shows the total number of Picard iterations for solving Example 6 directly using different values of α and grid meshes. As in the previous example, our stopping tolerance for the Picard iterations is set to 10^{-6}, with the initial guess is the solution of the linear Stokes problem for pipe flow.

Table 7.9 shows the number of Picard iterations (outer iterations) and average number of GMRES iterations (inner iterations) per outer iteration for different values of α and grid meshes. The inner stopping tolerance used for all values of α except 0.3 and 0.2 is $10^{-3}(residual)^{0.1}$. In the case of $\alpha = 0.3$ and 0.2, the inner tolerance used with

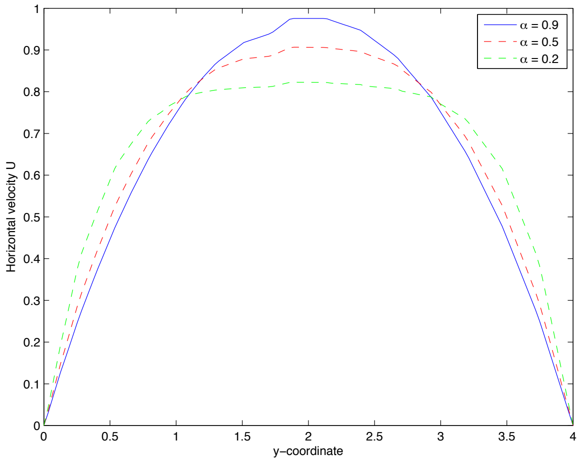


Figure 7.6: The velocity profiles of the discrete solution at $(\frac{1}{2}, y)$ for the pipe flow test problem.

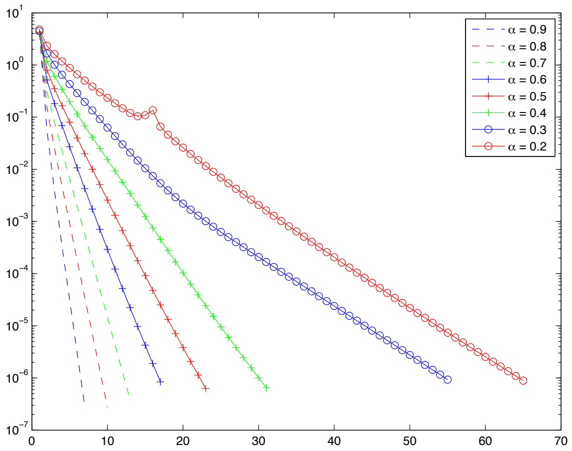


Figure 7.7: The convergence history of Picard iterations for different values of α for pipe flow test problem.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
15×15	7	10	13	17	23	31	55	65
31×31	7	10	13	17	22	36	45	79
63×63	7	10	13	16	21	31	40	63

Table 7.8: Total number of Picard iterations for pipe flow test problem.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
15×15	7 (10)	9 (9)	12 (7)	16 (7)	21 (4)	29 (3)	56 (3)	64 (7)
31×31	7 (9)	9 (9)	12 (7)	15 (6)	21 (5)	34 (3)	46 (4)	80 (5)
63×63	7 (10)	9 (9)	12 (6)	15 (6)	20 (5)	34 (4)	46 (4)	63 (6)

Table 7.9: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer,in blue) when solving Example 6 with preconditioner P_ν for different values of α and grid meshes.

$10^{-3}(residual)^{0.25}$ and $10^{-4}(residual)^{0.15}$ respectively. From the table, we have seen that the pipe flow test example is more stable than the driven cavity for α close to zero. In this example, a deflation preconditioner is not required due to the relatively small numbers of GMRES iterations required for convergence.

7.6 Summary

In this chapter, we have derived the weak formulation of the generalised Stokes equations and described some constitutive models. These models will appear in the constraints on our optimisation problem that we are going to investigate in the next chapter. The difficulty with these problems is dealing with the nonlinear behaviour that describes the complex flow.

Chapter 8

Optimal control of the generalised Stokes equations

In this chapter, we consider the problem of controlling the generalised Stokes equations. Depending on the formulation of the problem, we can consider either the case of distributed control or boundary control. Distributed control problems arise whenever the control is a domain variable, whereas boundary control problems arise whenever the control variable is formulated on the boundary. For this thesis, we only consider problems falling into the first category, namely distributed control problems, however a solution method for boundary control problems may be achieved analogously.

Both the classical and discrete formulations of the distributed control problem will be presented. We will then show that the first order optimality conditions for the latter formulation may be expressed in terms of an appropriately defined matrix-vector system. Based on the observations in Chapter 6, we consider use of GMRES coupled with an appropriate preconditioning strategy. Five different preconditioners will then be considered based on examination of the Schur complement of the system.

8.1 Distributed control problem

Let Ω be a bounded domain. Consider the distributed control problem subject to the generalised Stokes equations

$$\begin{cases} \min_{\vec{u},\vec{f}} J(\vec{u},\vec{f}) \\ \text{subject to} \\ \quad -\operatorname{div}\boldsymbol{\sigma} = \vec{f} \qquad \text{in } \Omega, \\ \qquad \operatorname{div}\vec{u} = 0 \qquad \text{in } \Omega, \\ \qquad \vec{u} = \vec{u}_D \qquad \text{on } \partial\Omega, \end{cases} \tag{8.1}$$

where

$$J(\vec{u},\vec{f}) = \frac{1}{2}\left\|\vec{u}-\vec{u}^d\right\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\left\|\vec{f}\right\|^2_{L^2(\Omega)}.$$

In the above, $\vec{u}$ denotes the velocity, p the pressure and $\vec{f}$ the control variable. Both the regularisation parameter (or the Tikhonov/penalty parameter) $\gamma > 0$ and the desired velocity field $\vec{u}^d$ are known parameters. Our aim is to determine $\vec{u}$ and $\vec{f}$ that satisfy the PDE problem (8.1) under appropriate penalisation such that $\vec{u}$ is as close to $\vec{u}^d$ with respect to the L_2 norm. The choice of regularisation parameter plays an important role in the type of solution that one can expect to achieve from (8.1). If γ is very small, the control variable $\vec{f}$ is not subject to heavy penalisation, and as such even relatively large values of $\vec{f}$ can be sought. In this situation, the range of permissible values of $\vec{f}$ with respect to the objective function J are quite varied, and thus we can expect that the velocity state variable $\vec{u}$ and the desired velocity $\vec{u}^d$ will be relatively close. However, large values of γ mean that the role played by $\vec{f}$ is much more restricted, since the associated contribution within the objective function will be dominant. As such, we can expect difficulties in determining the velocity state variable $\vec{u}$ close to $\vec{u}^d$ in the L_2 norm.

For existence and uniqueness theorems of optimal control problems for non Newtonian fluids, the interested reader is referred to [69]. As mentioned in the introduction, there are two approaches used for the solution of PDE-constrained optimisation problems. For this work, we have chosen to discretise first, then optimize. We discretise the objective function J and the PDE using the finite element method introduced in Chapter 5. Let V_h and Q_h denote an inf-sup stable mixed finite element pair of spaces. Let $\{\Phi_i\}$ and $\{\psi_k\}$ be the finite element bases of V_h and Q_h respectively. Then the discrete version of problem (8.1) is

$$\begin{cases} \min_{\mathbf{u},\mathbf{f}} J_h(\mathbf{u},\mathbf{f}) \\ \text{subject to} \\ \qquad \mathbf{A}(\mathbf{u})\mathbf{u} + \mathbf{B}^T\mathbf{p} = \mathbf{M}\mathbf{f} \\ \qquad\qquad \mathbf{B}\mathbf{u} = 0 \end{cases} \tag{8.2}$$

where

$$J_h(\mathbf{u},\mathbf{f}) = \frac{1}{2}(\mathbf{u}-\mathbf{u}^d)^T\mathbf{M}(\mathbf{u}-\mathbf{u}^d) + \frac{1}{2}\gamma\mathbf{f}^T\mathbf{M}\mathbf{f}.$$

The matrices $\mathbf{A}(\mathbf{u})$, $\mathbf{M}$ and $\mathbf{B}$ denote the vector Laplacian, vector mass and divergence matrices respectively. Each of these matrices are defined as follows:

$$\begin{aligned} \mathbf{A}_{ij} &= \int_\Omega \nu(\varepsilon(\vec{u}_h^m))\,\varepsilon(\vec{\phi}_j) : \varepsilon(\vec{\phi}_i)\; d\Omega, \\ \mathbf{B}_{jk} &= \int_\Omega \psi_j \operatorname{div}\vec{\phi}_k \;\; d\Omega, \\ \mathbf{M}_{ij} &= \int_\Omega \vec{\phi}_i \cdot \vec{\phi}_j \; d\Omega. \end{aligned}$$

The Lagrangian associated with (8.2) may be presented as follows:

$$L(\mathbf{u},\mathbf{f},\boldsymbol{\lambda}_i) = \frac{1}{2}(\mathbf{u}-\mathbf{u}^d)^T\mathbf{M}(\mathbf{u}-\mathbf{u}^d)+\frac{1}{2}\gamma\mathbf{f}^T\mathbf{M}\mathbf{f}+\boldsymbol{\lambda}_1\left[\mathbf{A}(\mathbf{u})\mathbf{u}+\mathbf{B}^T\mathbf{p}-\mathbf{M}\mathbf{f}\right]+\boldsymbol{\lambda}_2\left[\mathbf{B}\mathbf{u}\right], \quad (8.3)$$

where the vectors $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ denote Lagrange multipliers for the equality constraints.

The necessary and sufficient optimality conditions are obtained by setting the derivatives of the Lagrangian (8.3) with respect to the state variables $(\mathbf{u},\mathbf{p},\mathbf{f})$ and Lagrangian multipliers $(\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2)$ equal to zero

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{u}} &= \mathbf{M}(\mathbf{u}-\mathbf{u}^d)+\boldsymbol{J}_{\mathbf{Au}}\boldsymbol{\lambda}_1+\mathbf{B}^T\boldsymbol{\lambda}_2 = 0,\\
\frac{\partial L}{\partial \mathbf{p}} &= \mathbf{B}\boldsymbol{\lambda}_1 = 0,\\
\frac{\partial L}{\partial \mathbf{f}} &= \gamma\mathbf{M}\mathbf{f}-\mathbf{M}\boldsymbol{\lambda}_1 = 0,\\
\frac{\partial L}{\partial \boldsymbol{\lambda}_1} &= \mathbf{A}(\mathbf{u})\mathbf{u}+\mathbf{B}^T\mathbf{p}-\mathbf{M}\mathbf{f} = 0,\\
\frac{\partial L}{\partial \boldsymbol{\lambda}_2} &= \mathbf{B}\mathbf{u} = 0,
\end{aligned}$$

where

$$\boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} := \mathbf{A}(\mathbf{u})+\frac{\partial \mathbf{A}(\mathbf{u})}{\partial \mathbf{u}}\mathbf{u}\ , \quad (8.4)$$

representing the derivative of $\mathbf{A}(\mathbf{u})\mathbf{u}$ with respect to $\mathbf{u}$.

Based on the equations presented above, the optimality conditions may be obtained by solving the following matrix-vector system:

$$\begin{bmatrix} \mathbf{M} & 0 & 0 & \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ 0 & 0 & 0 & \mathbf{B} & 0 \\ 0 & 0 & \gamma\mathbf{M} & -\mathbf{M} & 0 \\ \mathbf{A}(\mathbf{u}) & \mathbf{B}^T & -\mathbf{M} & 0 & 0 \\ \mathbf{B} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \\ \mathbf{f} \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{M}\mathbf{u}^d \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

In order to use the derivation provided in Chapter 6 for saddle point systems, and also to identify a suitable block structure within the KKT system in order to motivate a preconditioning strategy, we eliminate the discrete control variable **f** from the third row and reorder the system in the following way:

$$K\mathbf{x} := \left[\begin{array}{cc|cc} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T & \frac{-1}{\gamma}\mathbf{M} & 0 \\ \mathbf{B} & 0 & 0 & 0 \\ \hline \mathbf{M} & 0 & \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ 0 & 0 & \mathbf{B} & 0 \end{array}\right] \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \\ \boldsymbol{\lambda}_1 \\ \boldsymbol{\lambda}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{M}\mathbf{u}^d \\ 0 \end{bmatrix}. \tag{8.5}$$

The matrix appearing on the left hand side of (8.5), denoted by K, is both large, nonsymmetric, nonlinear and indefinite. However, this matrix possesses a block sparse saddle structure (6.3), where we can define

$$\begin{aligned} \mathcal{A} &= \begin{bmatrix} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}, \quad \mathcal{B} = \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}, \\ \mathcal{C} &= \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}, \quad \mathcal{D} = \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}, \end{aligned} \tag{8.6}$$

so that the block $\mathcal{A}$ corresponds to the discrete generalised Stokes equations.

For the interested reader, some properties of saddle point matrices are reviewed in

[11]. A number of contributions that look to solve saddle point systems are presented in the literature, as described in [10, 63, 66], for instance. As discussed in Section 6.2, the performance of iterative solution methods is enhanced through an appropriate choice of preconditioner, which will be the topic of discussion in the next section.

8.2 Preconditioning the control problem

This section will involve the description of five different preconditioners. The first three of the preconditioners will involve an approximation to the Schur complement of $\mathcal{A}$ using the block structures described in (8.6). The last two preconditioners will not only consider an approximation to the Schur complement of K, but also for the matrix $\mathcal{A}$, corresponding to the $(1,1)$ block of K. All our preconditioners will have the following block triangular form

$$\begin{bmatrix} \widehat{\mathcal{A}} & \mathcal{B} \\ 0 & \widehat{S} \end{bmatrix}, \tag{8.7}$$

with both $\widehat{\mathcal{A}}$ and $\widehat{S}$ denoting approximations to $\mathcal{A}$ in (8.6) and the Schur complement $S := \mathcal{D} - \mathcal{C}\mathcal{A}^{-1}\mathcal{B}$, respectively.

In order to derive an appropriate approximation to the Schur complement, we consider the following:

$$S = \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \tag{8.8}$$

$$= \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A}(\mathbf{u})^{-1} + \mathbf{A}(\mathbf{u})^{-1}\mathbf{B}^T S_{loc1}^{-1}\mathbf{B}\mathbf{A}(\mathbf{u})^{-1} & * \\ * & * \end{bmatrix} \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} - \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M}(\mathbf{A}(\mathbf{u})^{-1} + \mathbf{A}(\mathbf{u})^{-1}\mathbf{B}^T S_{loc1}^{-1}\mathbf{B}\mathbf{A}(\mathbf{u})^{-1})\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}(\mathbf{A}(\mathbf{u})^{-1} + \mathbf{A}(\mathbf{u})^{-1}\mathbf{B}^T S_{loc1}^{-1}\mathbf{B}\mathbf{A}(\mathbf{u})^{-1})\mathbf{M} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} \tag{8.9}$$

where $S_{loc1} := -\mathbf{B}^T\mathbf{A}(\mathbf{u})^{-1}\mathbf{B}$ represents the Schur complement of $\mathcal{A}$.

8.2.1 First preconditioner P_0

The first preconditioner we will consider corresponds to retaining the (1,1) block as per the original system (8.5). However for the (2,2) block, it may be approximated by $\mathcal{D}_\gamma$ which takes into account the $\boldsymbol{J}_{\mathbf{A(u)u}}$ and γ. Our first preconditioner may then be presented as:

$$P_0 := \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ 0 & \mathcal{D}_\gamma \end{bmatrix}, \tag{8.10}$$

where

$$\mathcal{D}_\gamma := \begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}. \tag{8.11}$$

8.2.2 Second preconditioner $\widehat{P_0}$

If we reduce the preconditioner (8.10) in which not depend on γ any more to decrease the sparsity pattern in the term $\boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M}$ by replacing it by just $\boldsymbol{J}_{\mathbf{A(u)u}}$. Therefore, in this preconditioner we will consider corresponds to retaining the (1,1) and (2,2) blocks as per the original system (8.5), namely $\widehat{\mathcal{A}} := \mathcal{A}$ and $\widehat{\mathcal{S}} := \mathcal{D}$. The second preconditioner may written as:

$$\widehat{P_0} = \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ 0 & \mathcal{D} \end{bmatrix} = \left[\begin{array}{cc|cc} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T & \frac{-1}{\gamma}\mathbf{M} & 0 \\ \mathbf{B} & 0 & 0 & 0 \\ \hline & & \boldsymbol{J}_{\mathbf{A(u)u}} & \mathbf{B}^T \\ & & \mathbf{B} & 0 \end{array}\right]. \tag{8.12}$$

8.2.3 Third preconditioner $\widehat{\widehat{P_0}}$

Our third preconditioner for the system (8.5) aims to exploit the similarity between $\mathcal{A}$ and $\mathcal{D}$. We suggest to replace $\boldsymbol{J}_{\mathbf{A(u)u}}$ in (8.12) with $\mathbf{A(u)}$, meaning that a preconditioner of the following form is considered:

$$\widehat{\widehat{P_0}} := \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ 0 & \mathcal{A} \end{bmatrix} = \left[\begin{array}{cc|cc} \mathbf{A(u)} & \mathbf{B}^T & \frac{-1}{\gamma}\mathbf{M} & 0 \\ \mathbf{B} & 0 & 0 & 0 \\ \hline & & \mathbf{A(u)} & \mathbf{B}^T \\ & & \mathbf{B} & 0 \end{array}\right]. \tag{8.13}$$

8.2.4 Fourth preconditioner P_1

In addition to the approximation of the Schur complement, the aim with this preconditioner is to approximate the matrix $\mathcal{A}$ in (8.6). The saddle point structure of $\mathcal{A}$ displayed in (8.6) suggests the following block triangular approximation:

$$\widehat{\mathcal{A}} := \begin{bmatrix} \mathbf{A(u)} & \mathbf{B}^T \\ & M_\nu \end{bmatrix},$$

where M_ν is described in (6.10). Using this approximation, an approximation to the Schur complement S (8.8) may be considered in the following way:

$$\begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A(u)} & \mathbf{B}^T \\ & M_\nu \end{bmatrix}^{-1} \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix} - \begin{bmatrix} \mathbf{M} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{A(u)}^{-1} & * \\ & * \end{bmatrix} \begin{bmatrix} \frac{-1}{\gamma}\mathbf{M} & 0 \\ 0 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ \mathbf{B} & 0 \end{bmatrix}$$

$$\approx \begin{bmatrix} \boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ & M_\nu \end{bmatrix} := \widehat{S}.$$

Therefore, our preconditioner in this case can be described as follows:

$$P_1 = \left[\begin{array}{cc|cc} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T & \frac{-1}{\gamma}\mathbf{M} & 0 \\ & M_\nu & 0 & 0 \\ \hline & & \boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ & & & M_\nu \end{array}\right]. \tag{8.14}$$

The main computational cost associated with P_1 involves the inversion of $\mathbf{A}(\mathbf{u})^{-1}$. However, under a direct comparison, the preconditioner P_1 can be both stored and applied at a cheaper cost when compared to all previous preconditioners.

8.2.5 Fifth preconditioner $\widehat{P_1}$

Our final preconditioner uses (8.14) as well as an approximation for $\boldsymbol{J}_{\mathbf{A(u)u}}$ discussed in Section 8.2.3 to take on the following form:

$$\widehat{P_1} = \left[\begin{array}{cc|cc} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T & \frac{-1}{\gamma}\mathbf{M} & 0 \\ & M_\nu & 0 & 0 \\ \hline & & \mathbf{A}(\mathbf{u}) + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ & & & M_\nu \end{array}\right]. \tag{8.15}$$

The (1,1) and (2,2) blocks of preconditioners P_0 in (8.10), $\widehat{P_0}$ in (8.12) and $\widehat{\widehat{P_0}}$ in (8.13) have saddle point structures, which can be approximated by sub-preconditioners with same structure of (8.7). This will be the main focus of discussion in the next section.

8.3 Inner-outer GMRES approach

In order to solve our system with either P_0, $\widehat{P_0}$ or $\widehat{\widehat{P_0}}$ as described in (8.10), (8.12) and (8.13) respectively, we require the inversion of both the (1,1) and (2,2) blocks. Direct inversion and application of either of these blocks will be computationally expensive and even impractical for particularly large problems. However, we consider the application of GMRES (referred to as inner-GMRES) inside existing outer-GMRES iterations as an iterative alternative. Based on the structure of the preconditioning matrices, approximations are required for both the $(1,1)$ and $(2,2)$ blocks. We therefore require two inner GMRES solves for each outer GMRES iteration. For our work, the inner solves for the $(1,1)$ and $(2,2)$ blocks of our preconditioners will be referred to as inner GMRES1 and inner GMRES2 respectively. There are several choices that may be considered based on the five preconditioners presented in the previous chapter. We will describe some of them within this section.

8.3.1 $P_{0,1}$-solver

Based on our description of P_0 in (8.10), we consider the following upper triangular sub-preconditioners for both $\mathcal{A}$ and $\mathcal{D}_\gamma$ respectively

$$\mathcal{A} \approx \begin{bmatrix} \mathbf{A}(\mathbf{u}) & \mathbf{B}^T \\ & M_\nu \end{bmatrix} =: P_{\mathcal{A}}, \tag{8.16}$$

$$\mathcal{D}_\gamma \approx \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M} & \mathbf{B}^T \\ & M_\nu \end{bmatrix} =: P_{\mathcal{D}_\gamma}. \tag{8.17}$$

In $P_{0,1}$-solver, we solve our original system (8.5) using inner-outer GMRES. Each outer GMRES step will be preconditioned using P_0, with each inner solve preconditioned using both $P_{\mathcal{A}}$ and $P_{\mathcal{D}_\gamma}$ as sup-preconditioners for $\mathcal{A}$ and $\mathcal{D}_\gamma$, respectively.

8.3.2 $\widehat{P}_{0,1}$-solver

In this solver, we aim to solve our system (8.5) using outer-GMRES with $\widehat{P_0}$ as well as approximating $\mathcal{A}$ and $\mathcal{D}$ respectively by $P_{\mathcal{A}}$ and

$$P_{\mathcal{D}} := \begin{bmatrix} \boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} & \mathbf{B}^T \\ & M_\nu \end{bmatrix}. \tag{8.18}$$

8.3.3 $\widehat{\widehat{P}}_{0,1}$-solver

The preconditioner $\widehat{\widehat{P_0}}$ defined in (8.13) has identical $(1,1)$ and $(2,2)$ blocks. Therefore, both blocks may be approximated in a similar manner. Here, we consider $P_{\mathcal{A}}$ as per (8.16) as a sub-preconditioner for $\mathcal{A}$. We refer to this approach using the term $\widehat{\widehat{P}}_{0,1}$-solver.

Chapter 9 will provide numerical experimentation for the solution to (8.5), coupled with each of the preconditioners presented within this chapter. The deflated preconditioner presented in Section 6.4 will also be considered in order to improve results. Additionally, use of inner-outer GMRES will be considered in certain cases.

8.4 Comments

The following observations were used in order to enhance convergence and also to avoid complicated operations.

- An initial guess for our iterative method is determined based on a continuation strategy, whereby a solution is sought to a problem with a slightly higher value of α. For instance, when solving the generalised Stokes optimal control problem in the case $\alpha = 0.6$, an initial guess will be considered based on the solution to the equivalent problem in the case where $\alpha = 0.7$. In a similar manner, the initial guess in the case where the value of $\alpha = 0.9$ will correspond to the solution from the Newtonian problem (*i.e.* $\alpha = 1$). Working with such a strategy leads to a reduction in the total number of Picard iterations.

- The second term of the expression for $\boldsymbol{J}_{\mathbf{A(u)u}}$ given in (8.4) represents a tensor-vector product. However, approximations to the terms may be considered by using the standard definition of the derivative, namely

$$\frac{\partial \mathbf{A}(\mathbf{u})}{\partial \mathbf{u}}\mathbf{u} \approx \frac{\mathbf{A}\left(\mathbf{u} + \varepsilon \mathbf{u}\right) - \mathbf{A}\left(\mathbf{u}\right)}{\varepsilon},$$

  with ε denoting a small perturbation. Such an approximation allows for the avoidance of the matrix-vector product, since it is already involved.

- Section 8.3 suggests a means by which iterative approximations may be considered as an alternative to direct matrix inversion in application of our described preconditioners. In particular, the preconditioning approaches detailed in Section 8.2 required the inversion of at least one of $\mathbf{A}(\mathbf{u})$, $\boldsymbol{J}_{\mathbf{A(u)u}}$ or $\boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M}$. Iterative alternatives are also beneficial in that they are able to exploit sparsity patterns present within the involved matrices. As a result, we are able to solve the control problem through consideration of nested solution methods.

- The actual cost of the inner-outer GMRES algorithm as described is dependent on the quality of approximation for both $\mathcal{A}$ and $\mathcal{D}$. The complexity is dependent on the total number of inversions required of $\mathbf{A}(\mathbf{u})$, $\boldsymbol{J}_{\mathbf{A(u)u}}$ and/or $\boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M}$. The relevant figures can be found in A.1.

- By examining the $P_{0,1}$-solver as described in Section 8.3.1, it can be seen that this approach essentially amounts to preconditioning using P_1 inside outer preconditioned GMRES iterations (with P_0 used for the outer iterations). It is natural to question use of the inner-outer GMRES approach described for the $P_{0,1}$-solver if it is possible to instead precondition the inner iterations directly using P_1. Direct application would avoid the need for inner GMRES solves at each outer GMRES iteration. Nevertheless, such an approach would require storage and inversion of the matrix P_1, which can be computationally expensive. Use of inner GMRES iterations only require the action of P_1 applied to a vector. Therefore, if only a relatively small number of inner GMRES iterations are required for each outer GMRES iteration, significant computational savings can be made.

8.5 Summary

Within this chapter, we have presented both the classical and discrete formulation of the distributed control problem, along with the first order necessary optimality conditions in the discrete case. Based on these conditions, a matrix-vector system was formed involving a large nonsymmetric, nonlinear and indefinite system matrix. However, the block sparse saddle point structure of the matrix suggested the use of iterative solution methods combined with an appropriate preconditioning strategy.

Five block upper triangular preconditioners of the form (8.7) were presented in this section based on previous observations. Both preconditioners P_0 and $\widehat{P_0}$ considered an approximation for the Schur complement $\widehat{S}$, where preconditioners P_1 and $\widehat{P_1}$ also involved an approximation $\widehat{\mathcal{A}}$ to the matrix representation of the discrete generalised Stokes equations.

We also considered application of our preconditioning approaches through use of iterative solution methods. For this work, the structure of the involved matrices suggested further use of GMRES, leading to consideration and development of solution methods based on an inner-outer GMRES solution method, with inner GMRES solves considered at each outer GMRES iteration.

In order to compare our preconditioning approaches, we look to solve both the cavity driven flow problem and the pipe flow problem formulated as per (8.2) using P_0, $\widehat{P_0}$, $\widehat{\widehat{P_0}}$, P_1 and $\widehat{P_1}$. The application of these preconditioners will be considered both directly within GMRES and also approximately through use of inner-outer GMRES.

Chapter 9

Numerical experiments

In this chapter, numerical examples for distributed control problems will be presented. Figures for both the driven cavity flow and pipe flow problems described in Examples 5 and 6 respectively will be illustrated. The solution method used is similar to the approach in Chapter 7, involving linearisation through use of Picard iterations coupled with preconditioned GMRES for the resulting matrix-vector system. The preconditioners used are as described in Section 8.2.

For this chapter, the following test problem based on driven cavity and pipe flow will be considered. We consider the discretising the control problem using the well-studied $\mathcal{P}_2 - \mathcal{P}_0$ pair finite element basis, that discretise the velocity $\vec{u}$ using $\mathcal{P}_2$ basis functions, and the pressure p using $\mathcal{P}_0$ basis functions. We discretise the control $\vec{u}^d$ and the Lagrange multipliers λ_1 and λ_2 using $\mathcal{P}_2$ and $\mathcal{P}_0$ functions, respectively. In this chapter, we only consider the power law model. Therefore,

$$\boldsymbol{\sigma} = -pI + 2\nu_0 \left|\varepsilon(\vec{u})\right|^{\alpha-1} \varepsilon(\vec{u}).$$

A meshed grid of size 15 × 15 will be considered within this chapter.

Example 7 ***(Driven cavity flow control test problem)***

Let $\Omega = [0,1]^2$ *and consider the distributed control problem*

$$\min_{\vec{u},\vec{f}} \frac{1}{2}\left\|\vec{u}-\vec{u}^d\right\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma\left\|\vec{f}\right\|^2_{L^2(\Omega)}$$

subject to

$$\begin{aligned} -div\boldsymbol{\sigma} &= \vec{f} \quad in\ \Omega, \\ div\,\vec{u} &= 0 \quad in\ \Omega, \\ \vec{u} &= \vec{u}_D \quad on\ \partial\Omega, \end{aligned}$$

where the Dirichlet boundary condition is given by

$$\vec{u}_D = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix} \quad on\ \partial\Omega,$$

and the desired velocity

$$\vec{u}^d = \begin{pmatrix} 2(2y-1)(1-(2x-1)^2) \\ -2(2x-1)(1-(2y-1)^2) \end{pmatrix}.$$

Here, instead of applying homogeneous Dirichlet boundary conditions on the three sides of the domain as seen in Example 5, we consider Dirichlet boundary conditions corresponding to the desired velocity.

Table 9.1 illustrates results produced using the preconditioner P_0 described in (8.10) for varying values of α and γ. The numbers recorded represent the total number of Picard iterations required to achieve convergence, with the bracketed numbers representing the average number of GMRES iterations at each Picard step. The results were obtained using an adaptive stopping tolerance for GMRES, which for this problem was tightened based on γ, since the number of Picard iterations were found to blow up for particularly

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	5 (1)	6 (1)	7 (1)	9 (1)	12 (1)	15 (1)	20 (1)	30 (1)
$= 10^{-2}$	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	20 (1)	29 (1)
$= 10^{-3}$	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	22 (1)	33 (1)
$= 10^{-4}$	6 (2)	7 (2)	8 (1)	10 (1)	13 (1)	16 (1)	23 (1)	33 (1)
$= 10^{-5}$	4 (2)	5 (2)	7 (2)	8 (2)	10 (2)	14 (1)	17 (1)	23 (1)
$= 10^{-6}$	4 (3)	5 (3)	7 (3)	8 (3)	9 (3)	11 (2)	13 (2)	16 (2)

Table 9.1: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner P_0 for different values of α and γ.

small values of γ under previous criteria. For fixed α, the number of Picard iterations can be seen to remain roughly constant for varying γ. However, an increase is noted in the average number of GMRES iterations as γ is decreased due to tightness. For fixed γ, the average number of GMRES iterations per Picard step remains fairly constant for varying α.

In Table 9.2, results are provided for the driven cavity flow problem preconditioned using $\widehat{P_0}$ described in (8.12) for varying α and γ values. Here, the same adaptive inner tolerance was used from that described under preconditioning with P_0. For fixed α, we see similar characteristics to those presented in Table 7. However, for particular small values of γ, there is a gradual increase in the average number of GMRES iterations for decreasing α. The reason for this is due to the approximation of $\mathcal{D}$ by a term that does not take into consideration the changing nature of γ.

Table 9.3 illustrates results produced using the preconditioner $\widehat{\widehat{P_0}}$ described in (8.13)

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	5 (1)	6 (1)	7 (1)	9 (1)	12 (1)	15 (1)	20 (1)	30 (1)
$= 10^{-2}$	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	20 (1)	29 (1)
$= 10^{-3}$	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	21 (1)	33 (1)
$= 10^{-4}$	6 (2)	6 (2)	8 (2)	9 (2)	14 (2)	19 (1)	24 (1)	34 (1)
$= 10^{-5}$	5 (5)	6 (5)	7 (5)	8 (5)	10 (6)	12 (5)	17 (6)	23 (5)
$= 10^{-6}$	4 (10)	5 (11)	7 (12)	8 (13)	9 (14)	11 (15)	13 (15)	17 (13)

Table 9.2: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $\widehat{P_0}$ for different values of α and γ.

for varying α and γ values. We see similar characteristics to those presented in Table 9.2. The same can be said for the number of Picard iterations for fixed γ, however we now observe a logarithmic increase in the average number of GMRES iterations for all values of γ considered. The reason for this is due to the approximation of $\boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}}$ by $\mathbf{A}(\mathbf{u})$. For large values of α, the second contribution within $\boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}}$ described in (8.4) is negligible. However, the results suggest that this contribution becomes dominant for smaller values of α.

Tables 9.4 and 9.5 illustrate results for Example 7 using preconditioned GMRES based on use of P_1 and $\widehat{P}_1$ respectively. Comparison of Table 9.4 with Table 9.1 highlights similar characteristics to those displayed under preconditioning with P_0. However, the average number of GMRES iterations increases substantially for notably small values of γ, particularly for $\gamma = 10^{-5}$ and 10^{-6}. This behaviour is to be expected due to the relevant approximation. For both of these cases, a deflated preconditioner (6.12) was used and

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	5 (1)	6 (2)	7 (2)	9 (3)	12 (4)	15 (4)	20 (7)	34 (8)
$= 10^{-2}$	5 (1)	6 (2)	7 (2)	9 (3)	11 (4)	15 (4)	20 (7)	37 (7)
$= 10^{-3}$	5 (1)	6 (2)	7 (2)	9 (2)	12 (3)	15 (4)	22 (6)	37 (8)
$= 10^{-4}$	6 (2)	7 (2)	8 (2)	9 (3)	12 (4)	16 (5)	24 (6)	34 (8)
$= 10^{-5}$	5 (5)	6 (5)	7 (6)	8 (6)	10 (8)	12 (10)	16 (12)	22 (14)
$= 10^{-6}$	4 (10)	5 (11)	7 (13)	8 (14)	9 (15)	11 (18)	13 (22)	17 (26)

Table 9.3: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $\widehat{\widehat{P_0}}$ for different values of α and γ.

was able to provide some improvement, with the results displayed in Table 9.6.

Table 9.5 also displays similar characteristics to Table 9.4. However, an rise in the average number of GMRES iterations is observed for particularly small values of α. This suggests a dependence of α on the second term of $\boldsymbol{J}_{\mathbf{A(u)u}}$ in (8.4), since this second term is neglected within $\widehat{P}_1$.

For our preconditioners, there is no mesh dependence. Tables 9.7, 9.8 and 9.9 illustrate the Picard iterations and average number of GMRES iterations for solving Example 7 with preconditioners P_0, $\widehat{P_0}$ and $\widehat{\widehat{P_0}}$, respectively. For ease of presentations, we only display results for $\gamma = 10^{-3}$ based on the driven cavity flow problem.

We now consider use of the inner-outer GMRES approach for preconditioning using the solution methods described in Section 8.3.

Inner-outer GMRES approach

Using the presentation in Section 8.3, we now look to solve (8.5) based on an inner-

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	5 (5)	6 (5)	7 (5)	9 (5)	12 (5)	15 (5)	20 (4)	31 (3)
$= 10^{-2}$	5 (5)	6 (5)	7 (5)	9 (5)	11 (5)	15 (5)	20 (4)	30 (3)
$= 10^{-3}$	5 (5)	6 (5)	7 (5)	9 (5)	11 (4)	16 (5)	22 (5)	33 (3)
$= 10^{-4}$	5 (7)	6 (6)	7 (6)	9 (6)	12 (6)	16 (5)	23 (5)	34 (5)
$= 10^{-5}$	4 (13)	5 (14)	7 (14)	8 (14)	10 (15)	12 (13)	16 (12)	22 (7)
$= 10^{-6}$	4 (31)	6 (34)	7 (34)	8 (38)	9 (36)	11 (37)	13 (32)	17 (26)

Table 9.4: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner P_1 for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	5 (5)	6 (5)	7 (6)	9 (6)	12 (7)	15 (9)	20 (15)	30 (15)
$= 10^{-2}$	5 (5)	6 (5)	7 (6)	9 (6)	11 (7)	15 (9)	20 (15)	30 (15)
$= 10^{-3}$	5 (5)	6 (5)	7 (6)	9 (6)	12 (7)	16 (9)	22 (13)	34 (13)
$= 10^{-4}$	6 (5)	6 (6)	7 (5)	9 (6)	12 (6)	16 (6)	23 (8)	34 (12)
$= 10^{-5}$	5 (12)	6 (11)	7 (13)	8 (13)	10 (13)	15 (12)	19 (14)	23 (13)
$= 10^{-6}$	5 (27)	5 (33)	8 (34)	8 (35)	9 (35)	11 (28)	14 (24)	20 (31)

Table 9.5: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $\widehat{P_1}$ for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-5}$	5 (7)	6 (9)	7 (8)	8 (8)	10 (8)	13 (10)	18 (7)	27 (5)
$= 10^{-6}$	5 (16)	5 (17)	6 (17)	8 (17)	9 (15)	11 (14)	13 (20)	20 (13)

Table 9.6: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $M_{def}P_1$ for different values of α and for $\gamma = 10^{-5}$ and 10^{-6}.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
7×7	5 (1)	6 (1)	8 (1)	9 (1)	12 (1)	16 (1)	22 (1)	31 (1)
15×15	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	22 (1)	33 (1)
31×31	4 (1)	6 (1)	7 (1)	9 (1)	11 (1)	16 (1)	23 (1)	36 (1)

Table 9.7: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner P_0 for $\gamma = 10^{-3}$ and different values of α and mesh grids.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
7×7	5 (1)	6 (1)	8 (1)	9 (1)	12 (1)	16 (1)	22 (1)	31 (1)
15×15	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	15 (1)	21 (1)	33 (1)
31×31	5 (1)	6 (1)	7 (1)	9 (1)	11 (1)	16 (1)	23 (1)	36 (1)

Table 9.8: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $\widehat{P_0}$ for $\gamma = 10^{-3}$ and different values of α and mesh grids.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
7×7	5 (1)	6 (2)	7 (2)	9 (2)	12 (2)	17 (3)	24 (3)	36 (4)
15×15	5 (1)	6 (2)	7 (2)	9 (2)	12 (3)	16 (4)	22 (6)	37 (8)
31×31	5 (2)	6 (2)	7 (2)	9 (3)	11 (4)	16 (5)	23 (6)	36 (8)

Table 9.9: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 7 with preconditioner $\widehat{\widehat{P}}_0$ for $\gamma = 10^{-3}$ and different values of α and mesh grids.

outer GMRES approach under preconditioning with either P_0, $\widehat{P}_0$ or $\widehat{\widehat{P}}_0$.

We first consider the $P_{0,1}$-solver described in Section 8.3.1. The results are displayed in Table 9.10 for differing values of α and γ, with the average number of inner GMRES iterations for the application of both $P_{\mathcal{A}}$ and $P_{\mathcal{D}_\gamma}$ displayed in red and green respectively. All figures presented have been rounded to the nearest integer for ease of presentation. The tolerances for the outer GMRES solve and also for both of the inner GMRES solves were chosen by experimentation in order to achieve the same number of Picard iterations recorded in Table 9.1.

For relatively large γ, the average number of inner GMRES iterations appears to remain relatively constant for each value of α. However, a logarithmic increase is seen for particularly smaller values of γ. Nevertheless, this behaviour is to be expected based on the observations in Table 9.1.

Table 9.11 displays results for use of the $\widehat{P}_{0,1}$-solver described in Section 8.3.2. The main difference here with the aforementioned approach is the fact that the matrix $\mathcal{D}_\gamma$ is approximated by D and preconditioned by $P_{\mathcal{D}}$, with the same colour scheme used within this table as per Table 9.10. In terms of a direct comparison between both Tables 9.10 and 9.11, we see that for γ no less than 10^{-3}, the average number of inner GMRES iterations remains relatively small for all values of α. Nevertheless, a logarithmic increase is noted for particularly small values of γ. The effects of approximating $\boldsymbol{J}_{\mathbf{A(u)u}} + \frac{1}{\gamma}\mathbf{MA(u)}^{-1}\mathbf{M}$ by $\boldsymbol{J}_{\mathbf{A(u)u}}$ are also evident in the figures displayed for the average number of inner GMRES2 iterations. Here, a direct comparison of both tables shows a notable increase in the average number of iterations, particularly for small values of γ.

Whilst the results in Table 9.10 appear to be generally better than those in Table 9.11, it is important to factor in the associated computational costs in the application of either preconditioner in order to decide which of the two is not only effective but also computationally efficient to apply. Figures A.1, A.2 and A.3 illustrate the complexity for each of the preconditioning approaches $P_{0,1}$, $\widehat{P}_{0,1}$, $\widehat{\widehat{P}}_{0,1}$-solvers, respectively. A direct comparison between the associated figures from use of both $P_{\mathcal{D}}$ and $P_{\mathcal{A}}$ suggests that the overall complexity is generally better under preconditioning with $P_{\mathcal{D}}$. In fact, for notably small values of α, the complexity associated with the results in Table 9.11 is particularly substantial, largely due to the behaviour of the two terms within $\boldsymbol{J}_{\mathbf{A(u)u}}$ for differing values of α.

Table 9.12 displays results for Example 7 solved using the $\widehat{\widehat{P}}_{0,1}$-solver described in Section 8.3.3. From the figures, we see a roughly constant average number of inner GMRES1 and inner GMRES2 iterations. These results were again achieved through appropriate tightening of tolerances to deliver similar figures to Table 9.10. However, no matter how far the tolerances were tightened, the solver was seen to blow up for all values of γ in the case of $\alpha = 0.2$, indicating that this solution method is unsuitable for noticeably small values of α.

Figure 9.1 shows the velocity components and the stream lines for the desired velocity $\vec{u}^d$ for the driven cavity flow example 7. The figures in Table 9.13 show the horizontal component of computed state velocity u_1 for different values of α and γ. The figures in Tables 9.14, 9.15 and 9.16 show the equivalent plots for the vertical component of

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	3 2	3 2	3 2	3 2	3 1	3 1	2 1	2 1
$= 10^{-2}$	3 2	3 2	3 2	3 2	3 1	3 1	3 1	2 1
$= 10^{-3}$	3 2	3 2	3 2	2 1	2 1	2 1	1 1	1 1
$= 10^{-4}$	4 4	4 3	4 3	3 3	3 2	3 1	3 1	2 2
$= 10^{-5}$	5 7	5 7	5 8	5 8	6 8	4 4	4 4	3 3
$= 10^{-6}$	8 15	8 16	8 17	8 19	8 21	8 22	8 22	7 14

Table 9.10: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 7 with $P_{0,1}$-solver for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	3 2	3 2	2 2	2 1	2 1	2 1	1 1	3 1
$= 10^{-2}$	3 2	3 2	2 2	2 1	2 1	2 1	1 1	3 1
$= 10^{-3}$	2 1	3 2	2 2	2 1	2 1	2 1	1 1	3 2
$= 10^{-4}$	6 5	5 4	5 4	6 4	6 4	6 4	7 5	7 6
$= 10^{-5}$	7 7	8 7	8 8	9 8	9 8	10 9	12 9	14 11
$= 10^{-6}$	9 8	9 9	10 9	10 10	10 11	12 11	14 13	17 15

Table 9.11: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 7 with $\widehat{P}_{0,1}$-solver for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3
$\gamma = 10^{-1}$	7 6	7 6	7 7	7 7	8 7	9 8	10 7
$= 10^{-2}$	7 7	7 6	7 7	8 7	8 7	9 8	11 8
$= 10^{-3}$	7 6	7 6	7 7	8 7	8 7	9 7	11 8
$= 10^{-4}$	6 6	6 6	6 7	7 6	7 5	8 6	9 7
$= 10^{-5}$	7 7	8 8	8 8	9 8	10 9	10 9	11 10
$= 10^{-6}$	7 7	8 8	8 8	9 8	10 9	10 9	11 10

Table 9.12: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 7 with $\widehat{\widehat{P}}_{0,1}$-solver for different values of α and γ.

computed state velocity u_2, the pressure p and the stream lines for the computed state velocity, respectively.

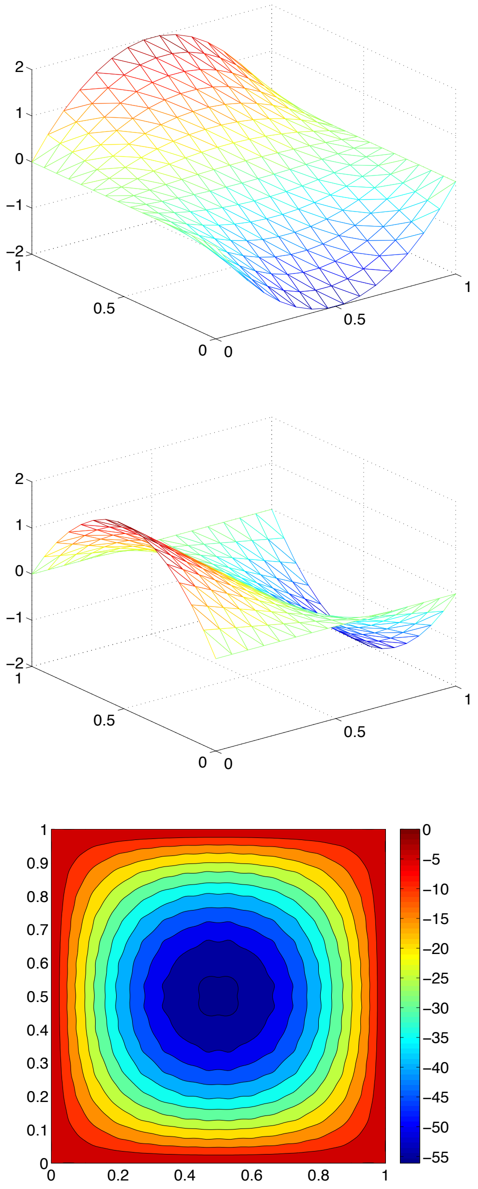


Figure 9.1: The velocity components and the stream lines for the desired velocity $\vec{u}^d$ for Example 7.

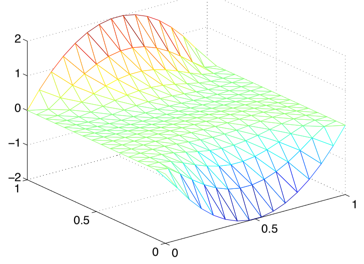
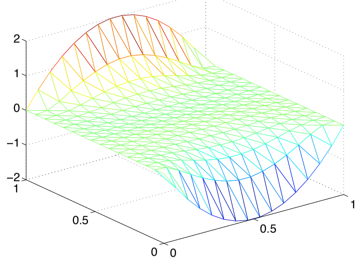
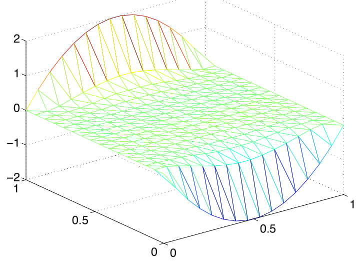
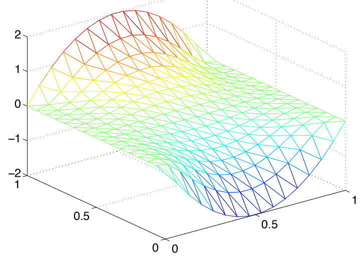
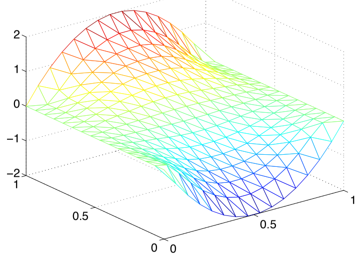
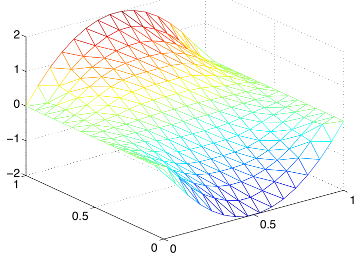
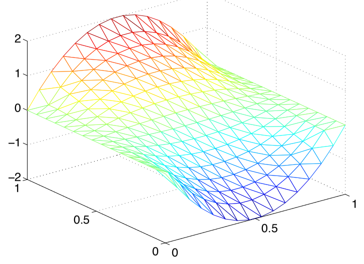
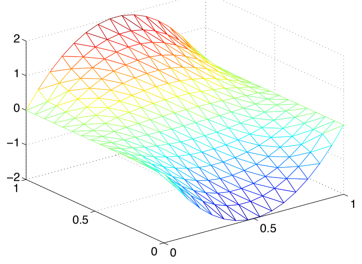
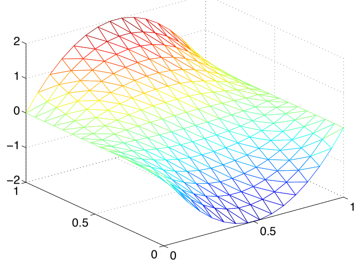
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.13: The horizontal component of computed state velocity u_1 for different values of α and γ for the driven cavity flow described in Example 7.

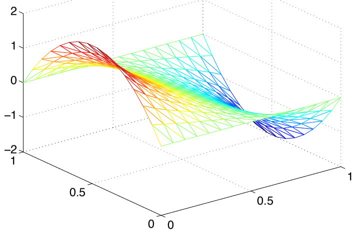
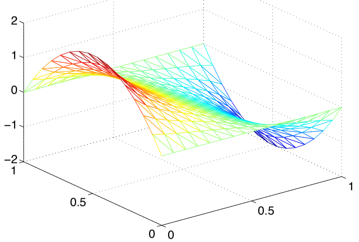
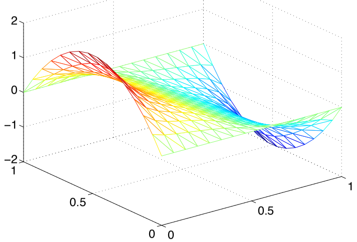
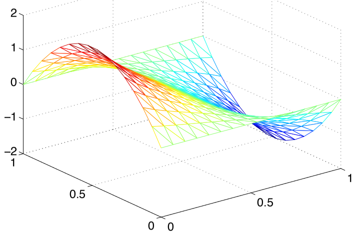
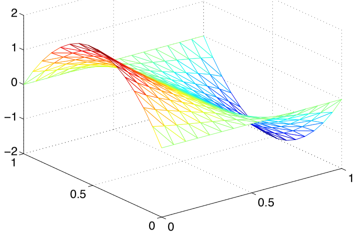
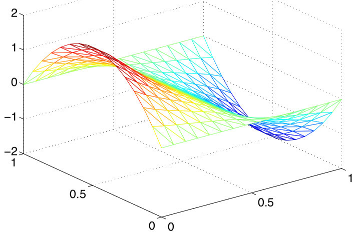
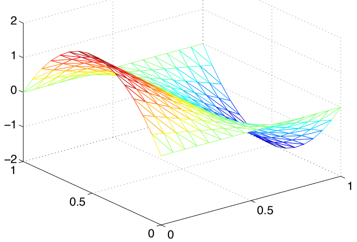
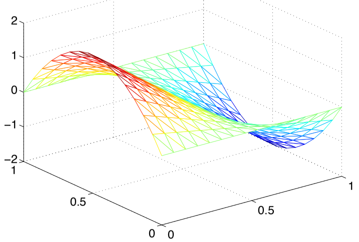
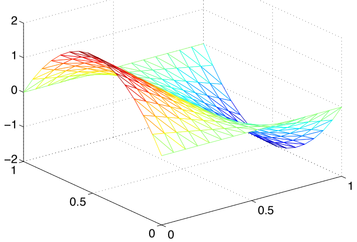
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.14: The vertical component of computed state velocity u_2 for different values of α and γ for the driven cavity flow described in Example 7.

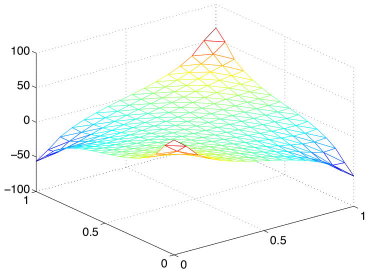
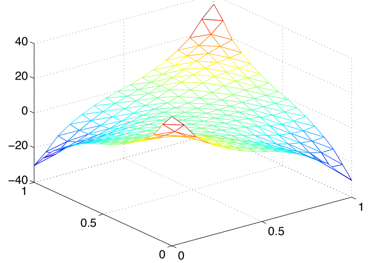
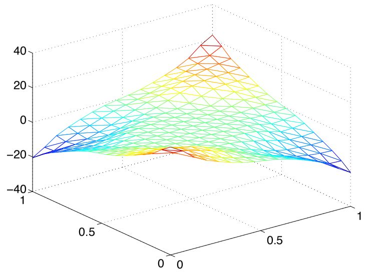
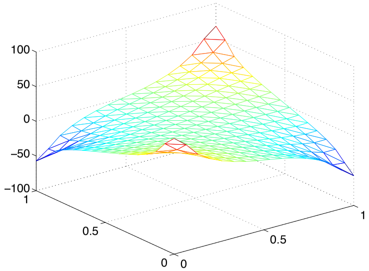
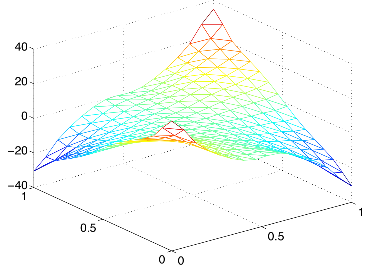
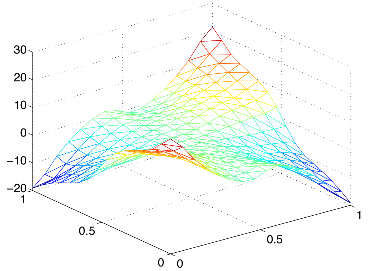
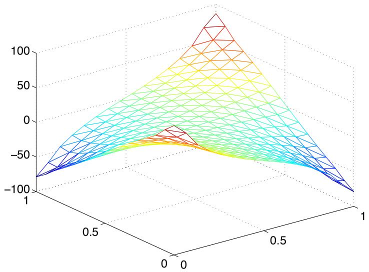
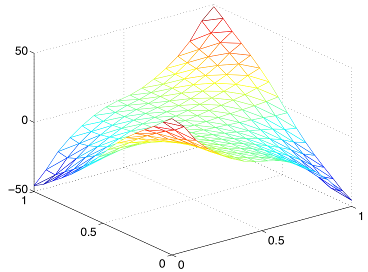
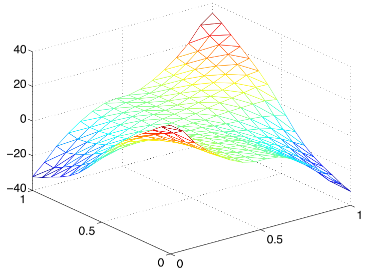
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.15: The pressure p for different values of α and γ for the driven cavity flow described in Example 7.

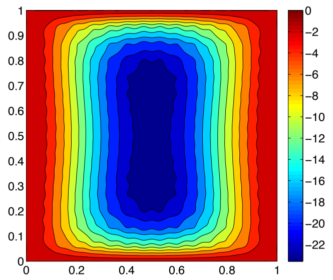
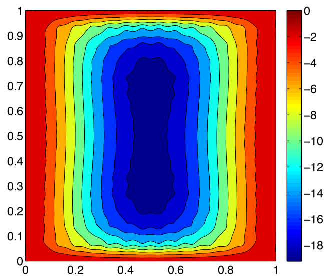
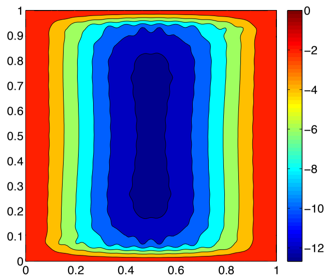
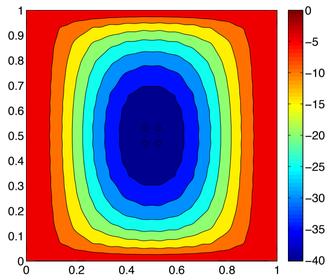
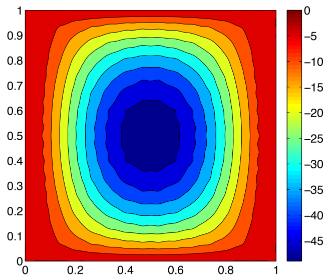
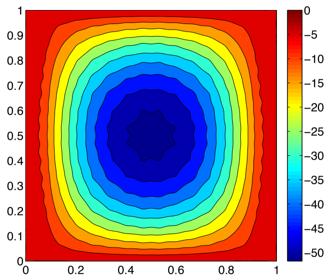
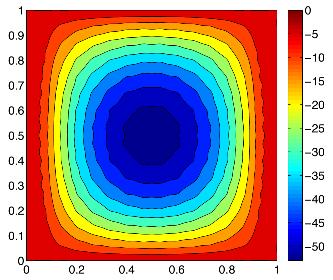
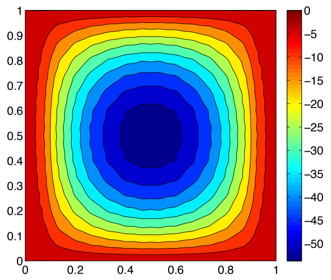
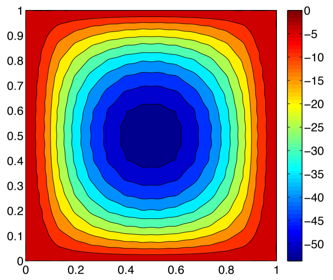
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.16: The stream lines for the computed state velocity for different values of α and γ for the driven cavity flow described in Example 7.

Example 8 ***(Pipe flow control test problem)***

Let $\Omega = [0,4] \times [0,1]$ *and consider the distributed control problem*

$$\min_{\vec{u},\vec{f}} \frac{1}{2} \left\| \vec{u} - \vec{u}^d \right\|^2_{L^2(\Omega)} + \frac{1}{2}\gamma \left\| \vec{f} \right\|^2_{L^2(\Omega)}$$

subject to

$$\begin{aligned} -div\,\boldsymbol{\sigma} &= \vec{f} \quad in\ \Omega, \\ div\,\vec{u} &= 0 \quad in\ \Omega, \\ \vec{u} &= \vec{u}_D \quad on\ \partial\Omega_D \\ \mathbf{n}\cdot\boldsymbol{\sigma} &= 0 \quad on\ \partial\Omega_N. \end{aligned}$$

The Dirichlet data on the boundary is given by

$$\vec{u}_D = \begin{cases} \begin{pmatrix} 4(y-y^2) \\ 0 \end{pmatrix} & on\ x = 0 \\ \mathbf{0} & on\ y = 0,\ y = 1. \end{cases}$$

and the desired velocity

$$\vec{u}^d = 4(y - y^2).$$

Tables 9.17 to 9.21 illustrate results for Example 8 under preconditioning with P_0, $\widehat{P}_0$, $\widehat{\widehat{P}}_0$, P_1 and $\widehat{P}_1$ respectively. For each of the preconditioning approaches, a like-for-like comparison of the numerical results shows that the recorded number of Picard iterations for the pipe flow problem are generally greater than those displayed for the driven cavity flow problem. This behaviour is particularly apparent for notably small values of α. The reasoning behind this is that as α tends to zero, the flow becomes akin to plug flow in the middle of the pipe. Consequently, more Picard iterations are needed in order to fully capture the parabolic behaviour of $\vec{u}^{\mathrm{d}}$. A direct comparison between Tables 9.4 and 9.20 for both examples under preconditioning with P_1 shows a significant increase in

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (1)	8 (1)	10 (1)	12 (1)	16 (1)	20 (1)	27 (1)	39 (1)
$= 10^{-2}$	6 (1)	8 (1)	10 (1)	12 (1)	15 (1)	20 (1)	27 (1)	39 (1)
$= 10^{-3}$	6 (1)	8 (1)	10 (1)	12 (1)	15 (1)	19 (1)	26 (1)	39 (1)
$= 10^{-4}$	6 (2)	8 (2)	10 (1)	12 (1)	13 (1)	19 (1)	26 (1)	41 (1)
$= 10^{-5}$	6 (4)	8 (4)	11 (3)	12 (3)	17 (2)	20 (2)	28 (2)	42 (2)
$= 10^{-6}$	6 (6)	7 (5)	9 (4)	12 (3)	19 (3)	20 (3)	30 (2)	54 (1)

Table 9.17: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner P_0 for different values of α and γ.

the average number of GMRES iterations for particularly small γ values. We therefore consider a deflated preconditioner M_{def} in order to improve on the recorded figures for the average number of GMRES iterations, with the results displayed in Table 9.22. A direct comparison between Tables 9.20 and Table 9.22 highlights the benefits of using a deflated preconditioner, with a reduction in the total number of GMRES iterations of up to a third noted in certain cases.

We now consider use of the inner-outer GMRES approach for preconditioning using the solution methods described in Section 8.3.

Inner-outer GMRES approach

Using the presentation in Section 8.3, we now look to solve (8.5) based on an inner-outer GMRES approach under preconditioning with either P_0, $\widehat{P_0}$ or $\widehat{\widehat{P_0}}$.

We first consider the $P_{0,1}$-solver described in Section 8.3.1. The results are displayed in Table 9.23 for differing values of α and γ, with the average number of inner GMRES

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (1)	8 (1)	11 (1)	12 (1)	16 (1)	20 (1)	27 (1)	39 (1)
$= 10^{-2}$	6 (1)	8 (1)	10 (1)	12 (1)	15 (1)	20 (1)	27 (1)	39 (1)
$= 10^{-3}$	6 (1)	8 (1)	10 (1)	12 (1)	15 (1)	19 (1)	26 (1)	39 (1)
$= 10^{-4}$	6 (3)	9 (3)	10 (3)	12 (2)	14 (2)	19 (2)	27 (2)	42 (2)
$= 10^{-5}$	6 (6)	8 (6)	11 (6)	12 (5)	16 (5)	20 (4)	28 (4)	46 (4)
$= 10^{-6}$	6 (15)	8 (14)	9 (15)	12 (13)	18 (13)	20 (12)	28 (12)	55 (10)

Table 9.18: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner $\widehat{P_0}$ for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (2)	8 (1)	10 (1)	12 (1)	16 (1)	20 (1)	28 (2)	39 (3)
$= 10^{-2}$	6 (2)	8 (1)	10 (1)	12 (1)	15 (1)	19 (1)	27 (2)	39 (3)
$= 10^{-3}$	6 (2)	8 (1)	10 (1)	11 (1)	15 (1)	18 (1)	25 (2)	40 (3)
$= 10^{-4}$	6 (3)	8 (2)	10 (2)	11 (2)	15 (2)	19 (2)	27 (3)	43 (4)
$= 10^{-5}$	6 (8)	7 (7)	10 (7)	12 (7)	16 (7)	19 (7)	27 (7)	44 (8)
$= 10^{-6}$	7 (15)	8 (14)	8 (14)	10 (14)	18 (14)	18 (14)	31 (13)	55 (12)

Table 9.19: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner $\widehat{\widehat{P_0}}$ for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (15)	8 (12)	10 (10)	12 (8)	16 (9)	20 (7)	26 (5)	39 (5)
$= 10^{-2}$	6 (19)	8 (18)	10 (15)	12 (12)	16 (10)	20 (9)	28 (8)	40 (6)
$= 10^{-3}$	6 (29)	9 (33)	10 (30)	13 (24)	15 (27)	20 (23)	28 (14)	43 (17)
$= 10^{-4}$	6 (78)	9 (67)	10 (67)	12 (73)	14 (65)	19 (68)	27 (63)	39 (57)
$= 10^{-5}$	6 (157)	8 (168)	11 (161)	12 (169)	17 (173)	20 (166)	25 (163)	44 (146)
$= 10^{-6}$	7 (330)	8 (278)	11 (290)	14 (313)	19 (289)	20 (270)	27 (262)	51 (233)

Table 9.20: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner P_1 for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (24)	8 (20)	10 (18)	12 (16)	16 (14)	20 (14)	26 (15)	39 (22)
$= 10^{-2}$	6 (29)	8 (26)	10 (20)	12 (19)	15 (18)	20 (20)	27 (18)	39 (23)
$= 10^{-3}$	7 (39)	8 (40)	10 (39)	13 (32)	14 (38)	20 (38)	24 (35)	42 (44)
$= 10^{-4}$	6 (87)	9 (80)	10 (74)	12 (89)	14 (88)	19 (87)	27 (100)	42 (96)
$= 10^{-5}$	6 (182)	8 (188)	11 (189)	12 (182)	17 (200)	20 (218)	27 (227)	44 (224)
$= 10^{-6}$	7 (348)	8 (325)	11 (335)	15 (365)	20 (330)	20 (354)	30 (347)	52 (336)

Table 9.21: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner $\widehat{P}_1$ for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	6 (13)	8 (10)	10 (7)	13 (8)	16 (9)	20 (6)	27 (4)	39 (4)
$= 10^{-2}$	6 (10)	8 (9)	10 (14)	13 (7)	16 (10)	20 (9)	27 (5)	40 (6)
$= 10^{-3}$	7 (23)	8 (16)	10 (14)	13 (23)	15 (9)	18 (9)	29 (12)	38 (12)
$= 10^{-4}$	6 (30)	9 (31)	10 (30)	12 (31)	15 (25)	20 (27)	27 (20)	43 (23)
$= 10^{-5}$	6 (70)	8 (83)	11 (84)	12 (111)	17 (71)	20 (68)	27 (56)	45 (70)
$= 10^{-6}$	6 (128)	8 (131)	10 (131)	13 (153)	19 (129)	19 (178)	32 (141)	56 (154)

Table 9.22: Number of Picard iterations and average number of GMRES iterations per outer iteration (rounded to the nearest integer, in blue) when solving Example 8 with preconditioner $M_{def}P_1$for different values of α and γ.

iterations for the application of both $P_{\mathcal{A}}$ and $P_{\mathcal{D}_\gamma}$ displayed in red and green respectively. All figures presented have been rounded to the nearest integer for ease of presentation. As was the case in Example 7, the tolerances for the outer GMRES solve and also for both of the inner GMRES solves were chosen by experimentation in order to achieve the same number of Picard iterations recorded in Table 9.17.

Here, similar results are observed as displayed for the driven cavity flow test problem. For relatively large γ, the average number of inner GMRES iterations appears to remain relatively constant for each value of α. However, a logarithmic increase is seen for particularly smaller values of γ. Nevertheless, this behaviour is to be expected based on the observations in Table 9.17.

Table 9.24 displays results for use of the $\widehat{P}_{0,1}$-solver described in Section 8.3.2 for the pipe flow problem. As mentioned in the numerical results for Example 7, the matrix $\mathcal{D}_\gamma$ is approximated by D and preconditioned by $P_{\mathcal{D}}$, with the same colour scheme used within this table as per Table 9.23. In terms of a direct comparison between both Tables 9.23 and 9.24, we see that for γ no less than 10^{-3}, the average number of inner GMRES iterations remains relatively small for all values of α. Nevertheless, a logarithmic increase is noted for particularly small values of γ. The effects of approximating $\boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}} + \frac{1}{\gamma}\mathbf{M}\mathbf{A}(\mathbf{u})^{-1}\mathbf{M}$ by $\boldsymbol{J}_{\mathbf{A}(\mathbf{u})\mathbf{u}}$ are also evident in the figures displayed for the average number of inner GMRES2 iterations. Here, a direct comparison of both tables shows a notable increase in the average number of iterations, particularly for small values of γ.

Table 9.25 displays results for Example 8 solved using the $\widehat{\widehat{P}}_{0,1}$-solver described in Section 8.3.3. As per the figures from the driven cavity flow example, we see a roughly constant average number of inner GMRES1 and inner GMRES2 iterations. These results were again achieved through appropriate tightening of tolerances to deliver similar figures to Table 9.10. However, no matter how far the tolerances were tightened, the solver was seen to blow up for all values of γ in the case of $\alpha = 0.2$. Failure to effectively solve for both examples suggests that this solution method is unsuitable for noticeably small values of α. Figures A.4, A.5 and A.6 illustrate the complexity for each of the preconditioning approaches $P_{0,1}$, $\widehat{P}_{0,1}$, $\widehat{\widehat{P}}_{0,1}$-solvers, respectively.

The figures in Table 9.26 show the horizontal component of computed state velocity u_1 for different values of α and γ. The figures in Tables 9.27 and 9.28 show the equivalent plots for the vertical component of computed state velocity u_2 and the pressure p, respectively.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	7 8	6 7	5 7	4 8	3 8	3 9	2 7	2 6
$= 10^{-2}$	8 7	6 8	5 8	4 9	3 9	3 9	2 8	2 5
$= 10^{-3}$	9 9	6 10	5 11	4 11	3 10	3 9	2 7	2 5
$= 10^{-4}$	10 15	9 16	9 18	8 18	6 16	4 13	3 10	3 9
$= 10^{-5}$	13 35	12 38	12 35	11 38	11 37	11 38	10 29	9 23
$= 10^{-6}$	15 58	15 56	15 59	16 61	16 59	17 56	19 60	20 59

Table 9.23: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 8 with $P_{0,1}$-solver for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2
$\gamma = 10^{-1}$	7 7	6 8	5 7	4 8	3 8	3 9	2 7	2 5
$= 10^{-2}$	7 7	6 7	5 8	4 9	3 8	3 9	2 8	2 6
$= 10^{-3}$	7 8	5 8	4 8	4 8	3 8	3 8	2 8	2 8
$= 10^{-4}$	12 10	11 11	11 11	11 11	11 11	10 11	11 13	12 17
$= 10^{-5}$	12 11	12 11	12 11	12 11	13 12	14 12	16 14	20 18
$= 10^{-6}$	13 11	13 11	13 11	13 11	14 12	16 13	18 15	21 19

Table 9.24: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 8 with $\widehat{P}_{0,1}$-solver for different values of α and γ.

$\alpha =$	0.9	0.8	0.7	0.6	0.5	0.4	0.3
$\gamma = 10^{-1}$	10 11	10 11	9 11	7 10	8 11	7 10	9 8
$= 10^{-2}$	10 11	10 11	9 11	8 11	8 11	8 10	9 7
$= 10^{-3}$	11 11	11 11	10 10	10 10	9 9	9 8	10 7
$= 10^{-4}$	12 12	11 12	12 12	11 12	11 12	12 13	13 15
$= 10^{-5}$	12 13	13 13	13 13	13 13	13 14	14 15	16 18
$= 10^{-6}$	16 16	16 16	16 16	17 16	17 17	19 19	21 22

Table 9.25: The average number of inner GMRES1 (in red) and inner GMRES2 (in green) iterations per outer GMRES iteration when solving Example 8 with $\widehat{\widehat{P}}_{0,1}$-solver for different values of α and γ.

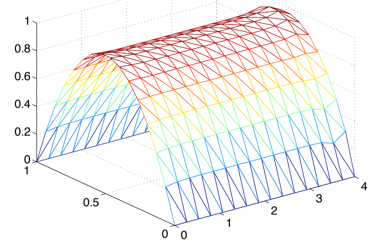
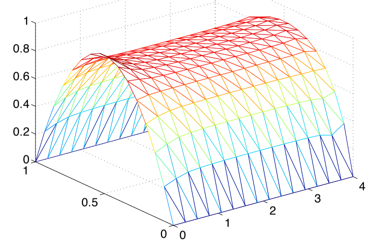
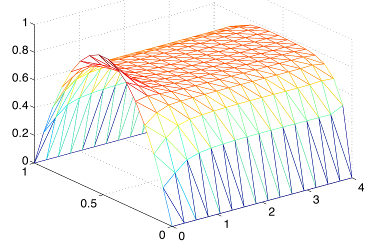
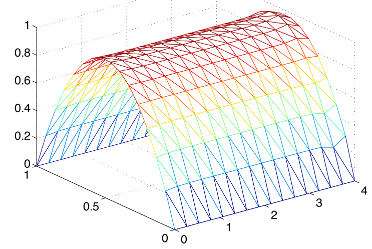
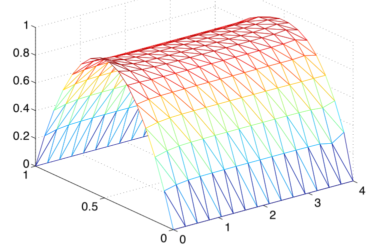
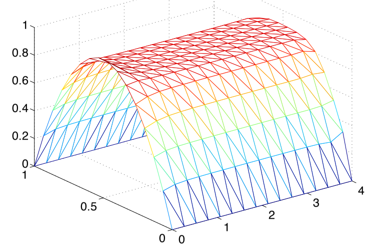
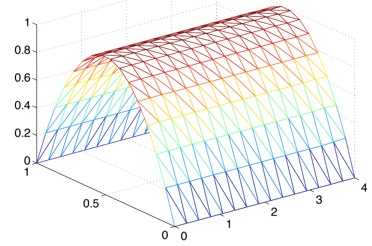
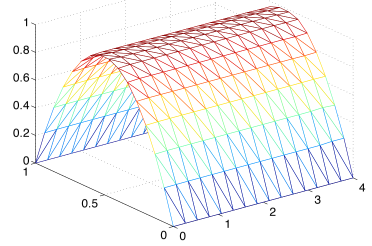
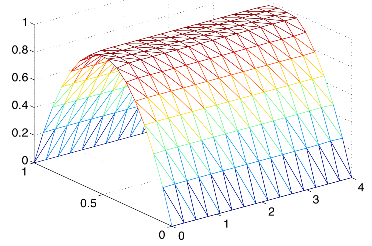
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.26: The horizontal component of computed state velocity u_1 for different values of α and γ for the pipe flow example 8.

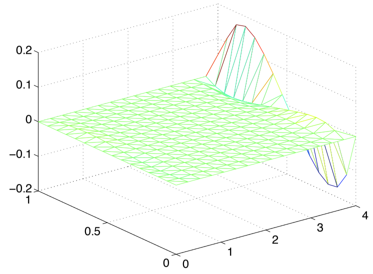
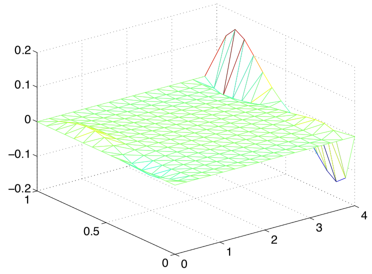
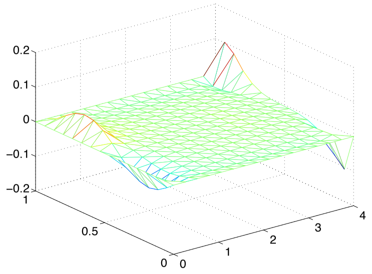
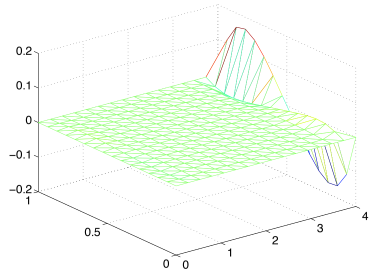
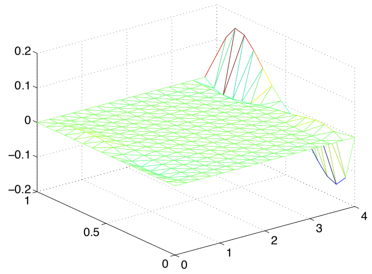
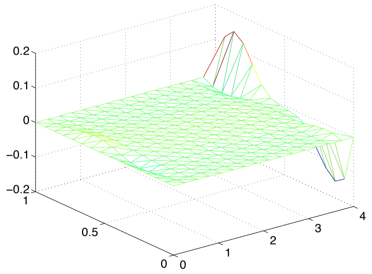
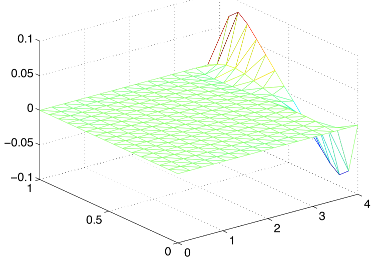
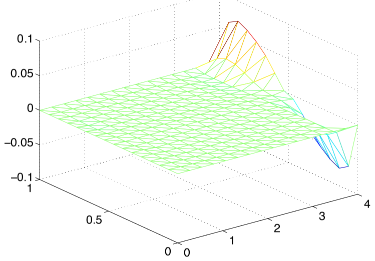
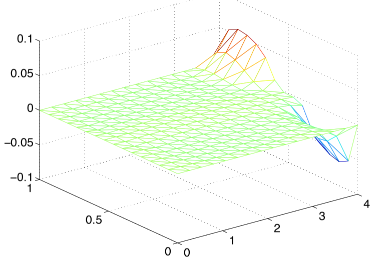
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.27: The vertical component of computed state velocity u_2 for different values of α and γ for the pipe flow example 8.

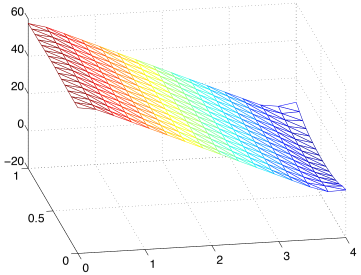
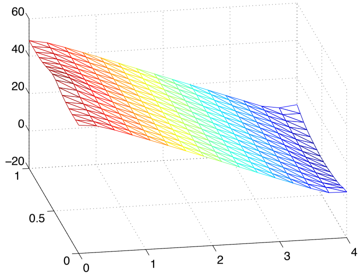
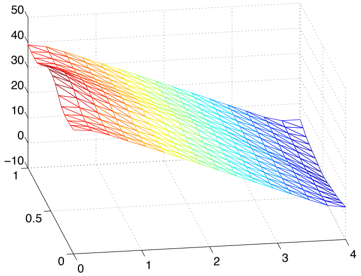
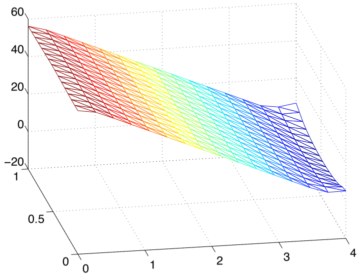
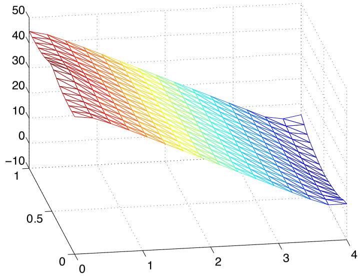
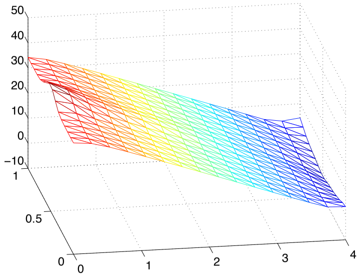
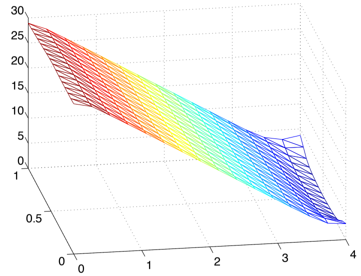
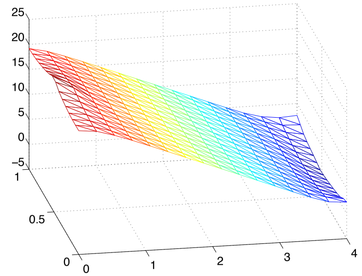
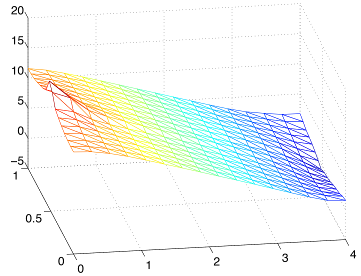
	$\alpha = 0.9$	$\alpha = 0.5$	$\alpha = 0.2$
$\gamma = 10^{-2}$			
$\gamma = 10^{-4}$			
$\gamma = 10^{-6}$			

Table 9.28: The pressure p for different values of α and γ for the pipe flow example 8.

Chapter 10

Conclusion and future work

Within this thesis, an in-depth study into solution techniques for PDE-constrained optimisation has been undertaken. These particular problems arise in a wide range of scientific and engineering applications, notably in problems of design, and so formulating effective solution methods for such problems is desired. We have described the underlying aim, namely to minimize an objective function $J(u, d)$ subject to constrains described through a system of PDE. We have described the steady-state Stokes equations for incompressible fluids, along with the associated weak formulation. Discretisation through use of the Galerkin finite element method is well understood for the Stokes problem, and was used within this thesis for both the Stokes equations and also the optimisation problem. The generalised version of the Stokes problem was also presented, along with specific models for viscosity.

Numerical experimentation was considered for both piecewise constant and also variable viscosity, including investigation (in the former case) into the spectrum of both the full and preconditioned Schur complement. These observations were then used in the application of an appropriate preconditioner based on deflation in the case of variable viscosity, with numerical results produced for both driven cavity and pipe flow problems.

Based on this presentation, attention was then focussed on controlling the generalised

Stokes equations. The notion of a distributed control problem was presented, along with the associated control problem subject to the generalised Stokes equations, both in continuous and discrete form. By writing down the Lagrangian for the discretised problem, a linear system was formed based on the first order optimality conditions.

By considering different solution methods for such systems, it was decided to solve the system using GMRES coupled with an appropriate preconditioning strategy. Under a suitable block ordering, the system matrix could then be viewed in terms of a block 2×2 representation, with the $(1, 1)$ block representing the system matrix for the discrete generalised Stokes equations. Attention was then focussed on the task of preconditioning the resulting system, involving consideration of a right preconditioning strategy. The preconditioner presented had a block triangular structure, where the task was to determine suitable approximations for both the $(1, 1)$ block and the Schur complement of the system matrix.

This led to the presentation of five different preconditioners based on the structure of the actual Schur complement. Results were shown suggesting mesh independent performance for each of the preconditioners considered. Nevertheless, each of the five preconditioners required inversion of both the $(1, 1)$ and $(2, 2)$ blocks within GMRES. These blocks possess a saddle point structure and have the potential to be substantial in size. Therefore, depending on the problem at hand storage and application of these matrices can present computational issues. An iterative alternative through use of inner-outer GMRES was considered, involving use of inner GMRES solves at each outer GMRES iteration. For our work, two inner GMRES solves were considered, leading to the development of three different solution methods.

Numerical results were presented for both driven cavity and pipe flow problems. Initially, figures were displayed based on direct application of each of the five aforementioned preconditioning approaches. Results for two of the preconditioning approaches showed an increase in the average number of GMRES iterations for particularly small values of γ. In order to remedy this issue, a deflated preconditioner was considered and was seen to provide effective results for both examples. Figures for the three inner-outer GMRES solution methods were also presented, along with associated complexity calculations in the case of driven cavity and pipe flow problems. Overall, results were obtained matching those recorded from direct application of the preconditioner under suitable adjustment of inner tolerances. Nevertheless, one of the solution methods was seen to struggle for notably small values of α, regardless of how tightly the inner tolerances were set.

Future work

- The focus of this thesis has involved distributed optimal control problems. Future work would see the presentation given here extended to boundary optimal control problems for the generalised Stokes problem.
- We would also like to investigate the inclusion of a convection term within the generalised Stokes equations, leading to consideration of the Oseen problem.
- Furthermore, deeper investigations into more appropriate treatment of the nonlinearity present within the optimisation problem should be considered. For instance, use of Newton's method in place of Picard iterations would suggest that the nonlinearities within the system would be handled more appropriately due to the use of first order information. Nevertheless, this would incur additional costs in the computation of the necessary derivatives, and so would need to be used under certain practical considerations. Ultimately, the main task would involve the derivation of an appropriate preconditioning strategy for the resulting formulation, with associated numerical results comparable (or showing improvements) to those provided within this thesis.

APPENDIX

A.1 Complexity of control problem

Figures in this appendix show the total inner-GMRES iterations per outer-GMRES iteration for both Examples 7 and 8 using $P_{0,1}$, $\widehat{P}_{0,1}$, $\widehat{\widehat{P}}_{0,1}$-solvers in Sections 8.3.1,8.3.2 and 8.3.3, respectively.

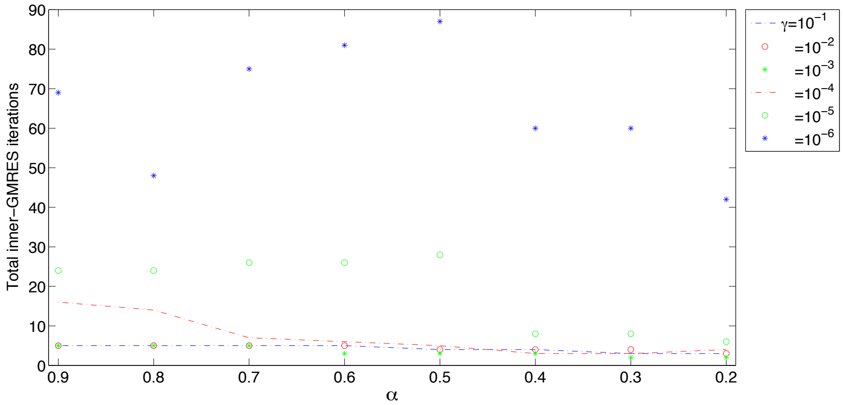


Figure A.1: The total inner-GMRES iterations per outer-GMRES iteration when solving driven cavity test problem in Example 7 using $P_{0,1}$-solver.

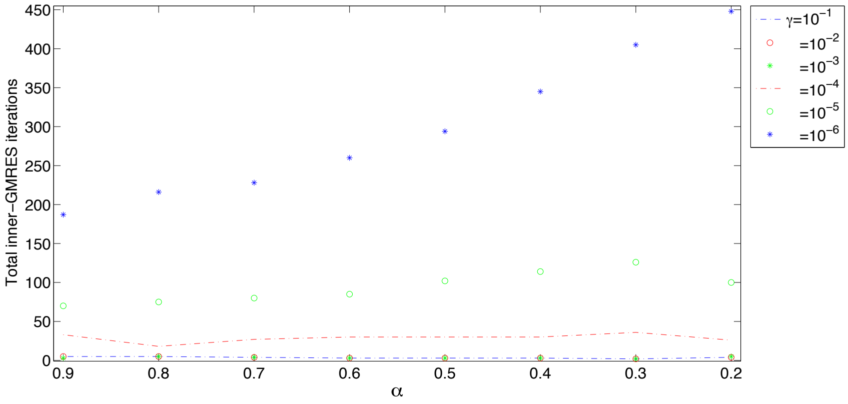


Figure A.2: The total inner-GMRES iterations per outer-GMRES iteration when solving driven cavity test problem in Example 7 using $\widehat{P}_{0,1}$-solvers.

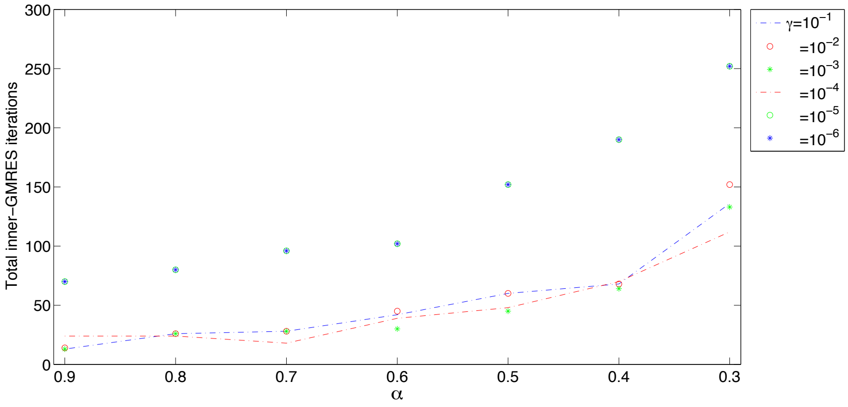


Figure A.3: The total inner-GMRES iterations per outer-GMRES iteration when solving driven cavity test problem in Example 7 using $\widehat{\widehat{P}}_{0,1}$-solvers.

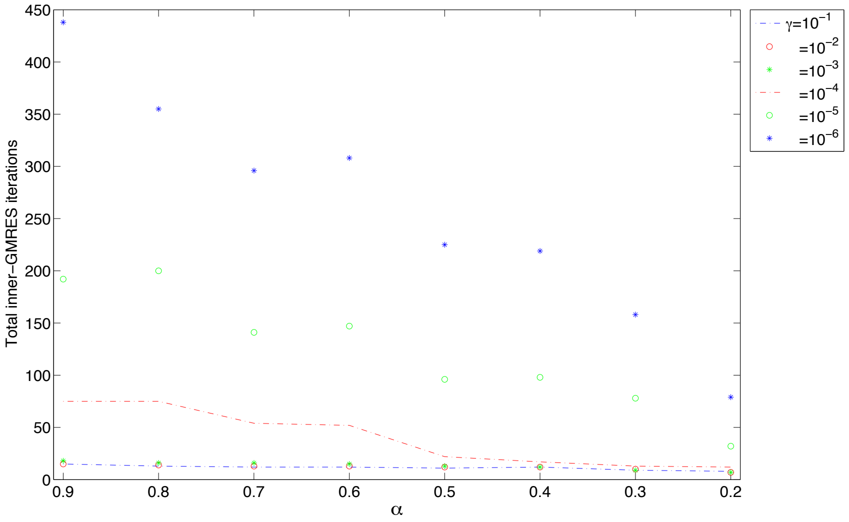


Figure A.4: The total inner-GMRES iterations per outer-GMRES iteration when solving pipe flow test problem in Example 8 using $P_{0,1}$-solvers.

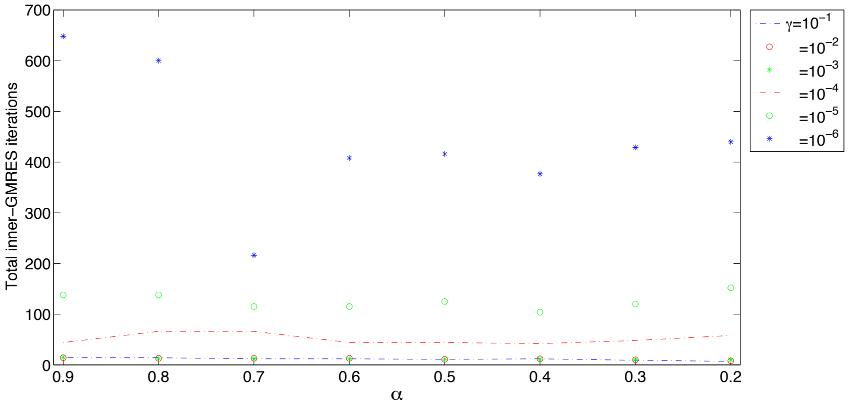


Figure A.5: The total inner-GMRES iterations per outer-GMRES iteration when solving pipe flow test problem in Example 8 using $\widehat{P}_{0,1}$-solvers.

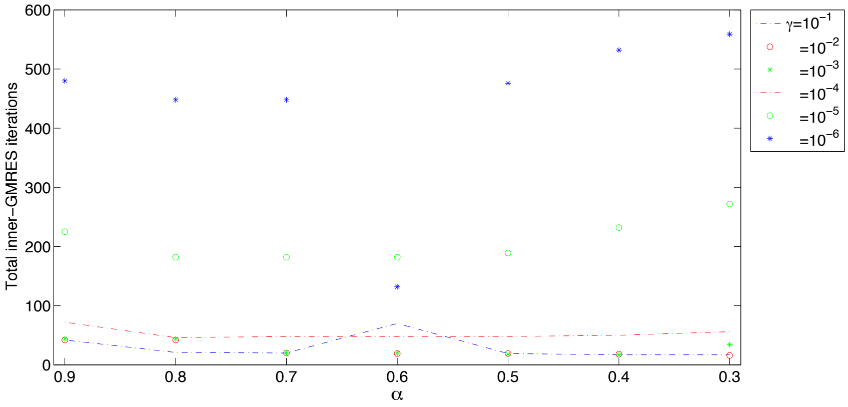


Figure A.6: The total inner-GMRES iterations per outer-GMRES iteration when solving pipe flow test problem in Example 8 using $\widehat{\widehat{P}}_{0,1}$-solvers.

References

[1] F. Abraham, M. Behr, and M. Heinkenschloss. The effect of stabilization in finite element methods for the optimal boundary control of the Oseen equations. *Finite Elements in Analysis and Design*, 41(3):229–251, 2004.

[2] R. A. Adams and J. J. Fournier. *Sobolev spaces*, volume 140. Academic press, 2003.

[3] M. Ainsworth and J. T. Oden. A posteriori error estimators for the Stokes and Oseen equations. *SIAM Journal on Numerical Analysis*, 34(1):228–245, 1997.

[4] D. N Arnold, F. Brezzi, and M. Fortin. A stable finite element for the Stokes equations. *Calcolo*, 21(4):337–344, 1984.

[5] J. Baglama, D. Calvetti, G. H. Golub, and L. Reichel. Adaptively preconditioned GMRES algorithms. *SIAM J. Sci. Comput.*, 20(1):243–269, 1998.

[6] J. Baranger and K. Najib. Analyse numerique des ecoulements quasi-Newtoniens dont la viscosite obeit a la loi puissance ou la loi de carreau. *Numerische Mathematik*, 58(1):35–49, 1990.

[7] J. W. Barrett and W. B. Liu. Finite element approximation of the p-Laplacian. *Mathematics of Computation*, 61:523–537, 1993.

[8] J. W. Barrett and W. B. Liu. Finite element error analysis of a quasi-Newtonian flow obeying the carreau or power law. *Numerische Mathematik*, 64(1):433–453, 1993.

[9] R. Barrett, M. Berry, T. F. Chan, and et al. *Templates for the solution of linear systems: building blocks for iterative methods*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1994.

[10] M. Benzi and G. H. Golub. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 2004.

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

[12] M. Benzi, E. Haber, and L. Taralli. A preconditioning technique for a class of PDE-constrained optimization problems. *Advances in Computational Mathematics*, 35(2-4):149–173, 2011.

[13] J. T. Betts. *Practical methods for optimal control using nonlinear programming*, volume 3 of *Advances in Design and Control*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001.

[14] R. B. Bird. Useful non-Newtonian models. *Annual Review of Fluid Mechanics*, 8(1):13–34, 1976.

[15] F. Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *RAIRO Anal. Numer*, 8(2):129–151, 1974.

[16] F. Brezzi and K. J. Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computer Methods in Applied Mechanics and Engineering*, 82(1):27–57, 1990.

[17] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods.* Springer-Verlag, 1991.

[18] F. Brezzi and J. Pitkäranta. *On the stabilization of finite element approximations of the Stokes equations.* Springer, 1984.

[19] P. J. Carreau. Rheological equations from molecular network theories. *Transactions of the Society of Rheology*, 16(1):99–127, 1972.

[20] P. J. Carreau, D. De Kee, and R. P. Chhabra. *Rheology of polymeric systems: principles and applications.* Hanser Publishers Munich, 1997.

[21] E. Casas and L. A. Fernández. Distributed control of systems governed by a general class of quasilinear elliptic equations. *Journal of differential equations*, 104(1):20–47, 1993.

[22] E. Casas, L. A. Fernández, and J. Yong. Optimal control of quasilinear parabolic equations. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 125(03):545–565, 1995.

[23] Z. Chen. *Finite element methods and their applications*, volume 5. Springer, 2005.

[24] R. P. Chhabra and J. F. Richardson. *Non-Newtonian flow and applied rheology: engineering applications*. Butterworth-Heinemann, 2008.

[25] S. S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. *CAAM TR02-01*, 2002.

[26] M. M. Cross. Rheology of non-Newtonian fluids: a new flow equation for pseudo-plastic systems. *Journal of Colloid Science*, 20(5):417–437, 1965.

[27] G. Dhatt, E. Lefran**c**cois, and G. Touzot. *Finite element method*. John Wiley & Sons, 2012.

[28] J. Donea and A. Huerta. *Finite element methods for flow problems*. John Wiley&Sons, Ltd, 2003.

[29] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct Methods for Sparse Matrices*. Monographs on Numerical Analysis. The Clarendon Press, Oxford University Press, New York, second edition, 1989. Oxford Science Publications.

[30] H. C. Elman. Preconditioners for saddle point problems arising in computational fluid dynamics. *Appl. Numer. Math.*, 43(1-2):75–89, 2002. 19th Dundee Biennial Conference on Numerical Analysis (2001).

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

[32] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, USA, 2005.

[33] J. Erhel, K. Burrage, and B. Pohl. Restarted GMRES preconditioned by deflation. *J. Comput. Appl. Math.*, 69(2):303–318, 1996.

[34] L. C. Evans. *Partial differential equations*, volume 19. American Mathematical Society, 1998.

[35] V. Girault and P. A. Raviart. Finite element methods for Navier-Stokes equations: theory and algorithms. *NASA STI/Recon Technical Report A*, 87:52227, 1986.

[36] N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM J. Sci. Comput.*, 23(4):1376–1395, 2001.

[37] P. M. Gresho, R. L. Sani, and M. S. Engelman. *Incompressible Flow and the Finite Element Method*. John Wiley & Sons, 1998.

[38] P. Grinevich. *Numerical solver for the variable viscosity Stokes type problem and applications*. PhD thesis, Moscow State University, 2010.

[39] P. P. Grinevich and M. A. Olshanskii. An iterative method for the Stokes-type problem with variable viscosity. *SIAM J. Sci. Comput.*, 31(5):3959–3978, 2009.

[40] J. Harris. *Rheology and non-Newtonian flow*. Longman New York, 1977.

[41] X. He and M. Neytcheva. Preconditioning the incompressible Navier-Stokes equations with variable viscosity. *Journal of Computational Mathematics*, 30(5):461–482, 2012.

[42] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Research Nat. Bur. Standards*, 49:409–436 (1953), 1952.

[43] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE constraints*, volume 23. Springer Verlag, 2009.

[44] Ilse CF Ipsen. A note on preconditioning nonsymmetric matrices. *SIAM Journal on Scientific Computing*, 23(3):1050–1051, 2001.

[45] D. Kay, D. Loghin, and A. J. Wathen. A preconditioner for the steady-state Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 24(1):237–256, 2002.

[46] C. Keller, N. I. M. Gould, and A. J. Wathen. Constraint preconditioning for indefinite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1300–1317, 2000.

[47] A. Klawonn. Block-triangular preconditioners for saddle point problems with a penalty term. *SIAM J. Sci. Comput.*, 19(1):172–184, 1998. Special issue on iterative methods (Copper Mountain, CO, 1996).

[48] M. Kollmann and W. Zulehner. A robust preconditioner for distributed optimal control for Stokes flow with control constraints. In *Numerical Mathematics and Advanced Applications 2011*, pages 771–779. Springer, 2013.

[49] J. L. Lions. *Optimal control of systems governed by partial differential equations.* 170 of Grundlehren Math. Wiss. Springer, 1971.

[50] D. Loghin and A. J. Wathen. Analysis of preconditioners for saddle-point problems. *SIAM Journal on Scientific Computing*, 25(6):2029–2049, 2004.

[51] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM J. Sci. Comput.*, 21(6):1969–1972 (electronic), 2000.

[52] M. A. Olshanskii and V. Simoncini. Acquired clustering properties and solution of certain saddle point systems. *SIAM Journal on Matrix Analysis and Applications*, 31(5):2754–2768, 2010.

[53] R. G. Owens and T. N. Phillips. *Computational rheology*, volume 2. World Scientific, 2002.

[54] J. W. Pearson. *Fast Iterative Solver for PDE-constrained optimization Problems.* PhD thesis, Oxford University, 2013.

[55] J. W. Pearson. On the role of commutator arguments in the development of parameter-robust preconditioners for Stokes control problems. *submitted to Electronic Transactions on Numerical Analysis*, 2013.

[56] J. W. Pearson. Preconditioned iterative methods for Navier-Stokes control problems. *submitted to SIAM Journal on Scientific Computing*, 2013.

[57] J. W. Pearson and M. Stoll. Fast iterative solution of reaction-diffusion control problems arising from chemical processes. *SIAM Journal on Scientific Computing*, 35(5):B987–B1009, 2013.

[58] J. W. Pearson and A. J. Wathen. Fast iterative solvers for convection-diffusion control problems. *Electronic Transactions on Numerical Analysis*, 40:294–310, 2013.

[59] T. Rees. *Preconditioning iterative methods for PDE constrained optimization*. PhD thesis, Oxford University, 2010.

[60] T. Rees, H. S. Dollar, and A. J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010.

[61] T. Rees and M. Stoll. Block-triangular preconditioners for PDE-constrained optimization. *Numerical Linear Algebra with Applications*, 17(6):977–996, 2010.

[62] T. Rees, M. Stoll, and A. J. Wathen. All-at-once preconditioning in PDE-constrained optimization. *Kybernetika*, 46(2):341–360, 2010.

[63] T. Rees and A. J. Wathen. Preconditioning iterative methods for the optimal control of the Stokes equations. *SIAM Journal on Scientific Computing*, 33(5):2903–2926, 2011.

[64] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[65] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM J. Sci. Statist. Comput.*, 7(3):856–869, 1986.

[66] J. Schoberl and W. Zulehner. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis And Applications*, 29(3):752–773, 2008.

[67] D. Silvester and A. J. Wathen. Fast iterative solution of stabilised stokes systems part i: using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis*, 30:630–649, 1993.

[68] D. Silvester and A. J. Wathen. Fast iterative solution of stabilised stokes systems part ii: using general block preconditioners. *SIAM Journal on Numerical Analysis*, 31(5):1352–1367, 1994.

[69] T. Slawig. Distributed control for a class of non-Newtonian fluids. *Journal of Differential Equations*, 219(1):116–143, 2005.

[70] M. Stoll and A. J. Wathen. All-at-once solution of time-dependent PDE-constrained optimization problems. *Kybernetika*, 46:341–360, 2010.

[71] M. Stoll and A. J. Wathen. All-at-once solution of time-dependent Stokes control. *Journal of Computational Physics*, 232(1):498–515, 2013.

[72] R. Temam. *Navier-Stokes Equation: Theory and Numerical Analysis*, volume 2. North-Holland, 1977.

[73] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

[74] F. Tröltzsch. *Optimal control of partial differential equations: theory, methods, and applications*, volume 112. Amer Mathematical Society, 2010.

[75] R. Verfürth. A posteriori error estimators for the Stokes equations. *Numerische Mathematik*, 55(3):309–325, 1989.

[76] A. J. Wathen. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA J. Numer. Anal.*, 7(4):449–457, 1987.

[77] L. W. White. Control of power-law fluids. *Nonlinear Analysis: Theory, Methods & Applications*, 9(3):289–298, 1985.

[78] M. Yeung, J. Tang, and C. Vuik. *On the convergence of GMRES with invariant-subspace deflation.* Delft University of Technology, Department of Applied Mathematical Analysis, 2010.

[79] W. Zulehner. Analysis of iterative methods for saddle point problems: a unified approach. *Mathematics of computation*, 71(238):479–505, 2002.

[80] W. Zulehner. Nonstandard norms and robust estimates for saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 32(2):536–560, 2011.