Iterative Solution of a Mixed Finite Element Discretisation of an Incompressible Magnetohydrodynamics Problem

by

Michael Wathen

M.Sci., The University of Birmingham, 2012

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

August 2014

Abstract

The aim of this thesis is to develop and numerically test a large scale preconditioned finite element implementation of an incompressible magnetohydrodynamics (MHD) model. To accomplish this, a broad-scope code has been generated using the finite element software package `FEniCS` and the linear algebra software `PETSc`. The code is modular, extremely flexible, and allows for implementing and testing different discretisations and linear algebra solvers with relatively modest effort. It can handle two- and three-dimensional problems in excess of 20 million degrees of freedom.

Incompressible MHD describes the interaction between an incompressible electrically charged fluid governed by the incompressible Navier-Stokes equations coupled with electromagnetic effects from Maxwell's equations in mixed form. We introduce a model problem and a mixed finite element discretisation based on using Taylor-Hood elements for the fluid variables and on a mixed Nédélec pair for the magnetic unknowns. We introduce three iteration strategies to handle the non-linearities present in the model, ranging from Picard iterations to completely decoupled schemes.

Adapting and extending ideas introduced in [Dan Li, *Numerical Solution of the Time-Harmonic Maxwell Equations and Incompressible Magnetohydrodynamics Problems*, Ph.D. Dissertation, The University of British

Columbia, 2010], we implement a preconditioning approach motivated by the block structure of the underlying linear systems in conjunction with state of the art preconditioners for the mixed Maxwell and Navier-Stokes subproblems. For the Picard iteration scheme we implement an inner-outer preconditioner.

The numerical results presented in this thesis demonstrate the efficient performance of our preconditioned solution techniques and show good scalability with respect to the discretisation parameters.

Preface

This dissertation is original, unpublished, independent work by the author, M. Wathen.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First, and foremost, I would like to thank my supervisors Chen Greif and Dominik Schötzau for their help and guidance. I greatly appreciate their understanding and insights throughout my graduate study and writing of this thesis. I thank Uri Ascher for reading this thesis and providing me with excellent feedback.

I would like to thank UBC Computer Science for their financial support over the last few year.

Finally, I would like to thank my friends and family for there love and support throughout my studies.

Chapter 1

Introduction

The primary topic of this thesis is to develop and numerically test a large scale implementation of an incompressible magnetohydrodynamics model. In this introductory chapter, we will first present a description of the model problem studied. We then give a brief overview of finite element methods and Krylov subspace methods for this problem. Finally, we outline the objectives, contributions and structure of the thesis.

1.1 A model problem in incompressible magnetohydrodynamics

The area of incompressible magnetohydrodynamics (MHD) describes the behaviour of electrically conductive incompressible fluids (liquid metals, plasma, salt water, etc) in an electromagnetic field [14, 23, 45]. MHD models couple electromagnetism and fluid dynamics. The coupling effects are due to two fundamental physical properties. Firstly, through the movement of the conductive material that induces a magnetic field which then modifies any existing electromagnetic field. Secondly, the magnetic and electric fields generate a mechanical force on the fluid known as the Lorentz force. The

Lorentz force accelerates the fluid particles in the direction normal to both the electric and magnetic fields.

Incompressible MHD has a number of important applications within technology and industry as well as Geophysical and Astrophysical applications. Some such applications are: electromagnetic pumping, aluminium electrolysis, the Earth's molten core and solar flares. For more applications see [45].

In this thesis, we are principally interested in an incompressible MHD model. This means that the electrically conductive fluid is incompressible, i.e., the mass of the fluid is conserved, and the electric resistivity of the fluid cannot be ignored. The MHD model we consider consist of two coupled fundamental equations: the incompressible Navier-Stokes equations and Maxwell's equations. We will outline the derivation of a formulation of an incompressible MHD model for a homogeneous and isotropic medium to ensure that all material parameters are constant; for full details see [8].

The transient incompressible Navier-Stokes equations that govern incompressible fluid flow are given by:

$$\rho_f\left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u}\right) - \mu\,\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} + \boldsymbol{f}_L \qquad \text{in } \Omega\times(0,T), \tag{1.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega\times(0,T). \tag{1.1b}$$

Here $\boldsymbol{u}$ and p are the velocity and pressure of the fluid, $\boldsymbol{f}$ denotes the body forces acting on the fluid and $\boldsymbol{f}_L$ is the Lorentz force, which will be specified later. The parameters $\mu > 0$ and $\rho_f > 0$ denote the dynamic viscosity and density of the fluid, respectively. The spatial domain is given by Ω and the

end time is denoted by T. Mass conservation is given by (1.1b), see [18, Chapter 0] for the derivation of the incompressible Navier-Stokes equations.

Maxwell's equations that govern electromagnetic effects are given by:

$$\text{Faraday's law:} \qquad \frac{\partial \boldsymbol{b}}{\partial t} + \nabla \times \boldsymbol{e} = \boldsymbol{0}, \tag{1.2a}$$

$$\text{Coulomb's law:} \qquad \nabla \cdot \boldsymbol{d} = \hat{\rho_e}, \tag{1.2b}$$

$$\text{Ampère's law:} \qquad -\frac{\partial \boldsymbol{d}}{\partial t} + \nabla \times \boldsymbol{h} = \boldsymbol{j}, \tag{1.2c}$$

$$\text{Gauss's law:} \qquad \nabla \cdot \boldsymbol{b} = 0. \tag{1.2d}$$

The fields in (1.2) are given by: $\boldsymbol{h}$ the magnetic field, $\boldsymbol{e}$ the electric field, $\boldsymbol{d}$ the electric displacement, $\boldsymbol{b}$ the magnetic induction and $\boldsymbol{j}$ the electric current density. The parameter $\hat{\rho_e}$ is the electric charge density. In a homogeneous and isotropic medium, the following linear relations hold:

$$\boldsymbol{d} = \delta \boldsymbol{e}, \quad \boldsymbol{b} = \mu \boldsymbol{h}, \tag{1.3}$$

where the constant $\delta > 0$ denotes the electric permittivity and the constant $\mu > 0$ the magnetic permeability. Using (1.2) together with (1.3) yields the form of Maxwell's equations considered in this thesis:

$$\text{Faraday's law:} \qquad \frac{\partial \boldsymbol{b}}{\partial t} + \nabla \times \boldsymbol{e} = \boldsymbol{0}, \tag{1.4a}$$

$$\text{Coulomb's law:} \qquad \nabla \cdot \boldsymbol{e} = \rho_e, \tag{1.4b}$$

$$\text{Ampère's law:} \qquad -\frac{\partial (\delta \boldsymbol{e})}{\partial t} + \nabla \times (\frac{1}{\mu}\boldsymbol{b}) = \boldsymbol{j}, \tag{1.4c}$$

$$\text{Gauss's law:} \qquad \nabla \cdot \boldsymbol{b} = 0, \tag{1.4d}$$

where $\rho_e = \frac{\hat{\rho_e}}{\delta}$; see [44, Chapter 1] for more details on Maxwell's equations.

The physical assumptions we consider to form a MHD model are the same as in [8, Section 2.1]. More precisely we assume:

- Non-relativistic motion: The characteristic fluid velocity is assumed to be orders of magnitude smaller than the speed of light.

- Low-frequency approximation: Phenomena involving high frequencies are omitted. That is, the term $\frac{\partial(\delta \boldsymbol{e})}{\partial t}$ involving the displacement current is neglected in Maxwell's equations. Therefore, Ampère's law (1.4c) simplifies to

$$\nabla \times \left(\frac{1}{\mu}\boldsymbol{b}\right) = \boldsymbol{j}. \tag{1.5}$$

- Quasi-neutrality assumption: Positive and negative charges are equal in any given region. The convection current is omitted and Ohm's law now reads

$$\boldsymbol{j} = \theta(\boldsymbol{e} + \boldsymbol{u} \times \boldsymbol{b}), \tag{1.6}$$

  where positive parameter θ defines the electric conductivity of the fluid and $\boldsymbol{u} \times \boldsymbol{b}$ corresponds to the charge density induced by the fluid motion. The electrical resistivity is given by $1/\theta$ and causes dissipative effects in Maxwell's equations but is not to be neglected in this model.

- Non-magnetisation and non-polarisation: The assumptions of homogeneity and isotropy also imply that the medium is non-magnetisable and non-polarisable.

The Lorentz force $\boldsymbol{f}_L$ in (1.1a) can now be expressed as:

$$\boldsymbol{f}_L = \frac{1}{\rho_f \, \mu}(\nabla \times \boldsymbol{b}) \times \boldsymbol{b}. \tag{1.7}$$

The electromotive term in (1.6) now enters Faraday's law (1.4a) by combining (1.6) and (1.5) into the following expression for $\boldsymbol{e}$:

$$\boldsymbol{e} = \frac{1}{\theta}\left(\nabla \times \left(\frac{1}{\mu}\boldsymbol{b}\right) - \boldsymbol{u} \times \boldsymbol{b}\right).$$

Using the assumptions above, elimination of the electric field $\boldsymbol{e}$ and non-dimensionalisation, the systems in (1.1) and (1.4) are coupled into the following set of partial differential equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} - \nu \, \Delta \boldsymbol{u} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p - \kappa \, (\nabla \times \boldsymbol{b}) \times \boldsymbol{b} = \boldsymbol{f}, \tag{1.8a}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1.8b}$$

$$\frac{\partial \boldsymbol{b}}{\partial t} + \nu_m \, \nabla \times (\nabla \times \boldsymbol{b}) - \nabla \times (\boldsymbol{u} \times \boldsymbol{b}) = \boldsymbol{0}, \tag{1.8c}$$

$$\nabla \cdot \boldsymbol{b} = 0, \tag{1.8d}$$

with suitable initial conditions and boundary conditions; see [8, Section 2]. The unknowns $\boldsymbol{u}$, p and $\boldsymbol{b}$ are the fluid velocity, the fluid pressure and the magnetic field, respectively.

The solution to (1.8) depends on three non-dimensional parameters $\nu = {}^1\!/_{\mathrm{Re}}$, $\nu_m = {}^1\!/_{\mathrm{Rm}}$ and κ. The first parameter Re is the hydrodynamic Reynolds number, which indicates the balance between the inertial forces and the viscous forces. The parameter Rm is the magnetic Reynolds number,

which measures the effect by which the magnetic field induces flow motion. The final parameter, the coupling number, κ, represents the influence of the electromagnetic field on the flow. It is sometimes defined in terms of the Hartmann number denoted by Ha, as

$$\mathrm{Ha} = \sqrt{\kappa \,\mathrm{Re}\,\mathrm{Rm}}.$$

To find typical physical values for these parameters, we refer to [8, 23, 51]. We refer to ν as the viscosity for the rest of this thesis.

In this thesis, we are interested in the steady-state ($\frac{\partial}{\partial t} = 0$) version of (1.8):

$$-\nu\,\Delta \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \kappa\,(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \boldsymbol{f}, \tag{1.9a}$$

$$\nabla\cdot\boldsymbol{u} = 0, \tag{1.9b}$$

$$\kappa\nu_m\,\nabla\times(\nabla\times\boldsymbol{b}) - \kappa\,\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) = \boldsymbol{0}, \tag{1.9c}$$

$$\nabla\cdot\boldsymbol{b} = 0. \tag{1.9d}$$

Note we have multiplied (1.9c) by κ to enforce skew-symmetry of the coupling terms. We will consider both two- and three-dimensional solutions to (1.9). See Appendix A for curl definitions.

1.2 Numerical solution

The partial differential equation (PDE) system given in (1.9) requires a numerical approximation as in general an analytical solution is not possible.

There are two main components in computing a numerical solution of a PDE:

1. Discretisation: take a continuous model and transfer it into a discrete model;

2. Solve: take the discretised model and solve for the unknowns.

In this thesis, we use mixed finite element methods for discretising an MHD model problem, and solve it using preconditioned Krylov subspace methods. In the sequel, we briefly describe these components in the context of the approach we take.

1.2.1 Finite element methods for incompressible MHD problems

There are several finite element methods for discretising MHD problems as in (1.9). A common approach in the literature is to approximate the magnetic field using standard nodal H^1-conforming elements [8, 22, 28, 50]. Such a formulation enables one to use the following vector calculus identity

$$-\Delta \boldsymbol{b} = \nabla \times (\nabla \times \boldsymbol{b}) - \nabla(\nabla \cdot \boldsymbol{b}). \tag{1.10}$$

Since $\boldsymbol{b}$ is divergences-free it is then possible to apply an augmentation technique to replace the curl-curl operator with a vector Laplacian. This then reduces one of the principal computational difficulties, namely the large null-space of the curl-curl operator in (1.9c). However, one of the main problems using H^1-conforming elements for the magnetic field is that for

non-convex domains (such as the 2D L-shaped domain with a reentrant corner, or the 3D Fichera corner domain with reentrant edges and corners) the magnetic field will converge to a solution that is not correct around the singular point, see [12, 13]. Therefore, we consider a mixed discretisation that captures singular solutions correctly. One such family of elements are $H(\mathrm{curl})$ conforming Nédélec elements [47].

To enable the use of Nédélec elements for the magnetic field we use the mixed formulation in [25, 55]. This leads to the following governing equations in a domain Ω:

$$-\nu\,\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \kappa\,(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1.11a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{1.11b}$$

$$\kappa\nu_m\,\nabla\times(\nabla\times\boldsymbol{b}) + \nabla r - \kappa\,\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) = \boldsymbol{g} \qquad \text{in } \Omega, \tag{1.11c}$$

$$\nabla\cdot\boldsymbol{b} = 0 \qquad \text{in } \Omega, \tag{1.11d}$$

where we have introduced the Lagrange multiplier, r, in the form of ∇r in (1.11c). Again, $\boldsymbol{u}$ and p are the velocity and pressure of the fluids and $\boldsymbol{b}$ is the magnetic field. The introduction of the Lagrange multiplier, r, corresponds to the divergence-free constraint (1.11d) of the magnetic field. With the addition of the Lagrange multiplier, r, we may also introduce a generic forcing term $\boldsymbol{g}$ associated with the Maxwell part of (1.11).

The numerical tests that we will consider will have inhomogeneous Dirichlet boundary conditions for the fluid and magnetic fields and homo-

geneous Dirichlet boundary condition for the multiplier, r, of the form:

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega, \tag{1.12a}$$

$$\boldsymbol{n} \times \boldsymbol{b} = \boldsymbol{n} \times \boldsymbol{b}_D \qquad \text{on } \partial\Omega, \tag{1.12b}$$

$$r = 0 \qquad \text{on } \partial\Omega, \tag{1.12c}$$

where $\boldsymbol{u}_D$ and $\boldsymbol{b}_D$ are given functions, and $\boldsymbol{n}$ is the unit outward normal to the boundary $\partial\Omega$. Notice, by taking the divergence of equation (1.11c) we obtain Poisson's equation (as $\nabla \cdot \nabla \times \boldsymbol{b} = 0$):

$$\Delta r = \nabla \cdot \boldsymbol{g} \quad \text{in } \Omega, \quad r = 0 \quad \text{on } \partial\Omega.$$

In many physical applications $\boldsymbol{g}$ is divergence-free, which implies that the multiplier, r, is zero. In general, the main purpose of the magnetic multiplier is to provide stability, see [17].

1.2.2 Preconditioning incompressible MHD problems

Incompressible MHD problems have been extensively studied in the context of various discretisation and formulations. However, the development of preconditioned iterative solutions to MHD problems is limited. We refer the reader to the Appendix for a review of Krylov subspace solvers for linear systems. In this thesis our focus will be on the implementation of preconditioners which are tailored to MHD model problem.

In the literature there have not been too many approaches to precodition-ing the MHD equations given in either the non-multiplier form (1.9) or with

the multiplier form (1.11). In the very recent work [50], an operator-based preconditioner for the non-multiplier MHD equations (1.9) has been proposed in the context of H^1-conforming elements for the magnetic field. To form their preconditioner the authors use the identity (1.10) and a discrete commutator idea to form approximations to the Schur complements. This is based on an approach for a preconditioner to the Navier-Stokes system as in [18, Chapter 8]. The preconditioners we employ are based on similar Navier-Stokes preconditioners but rely on the Maxwell preconditioner in [27] for $H(\text{curl})$ elements.

1.3 Objectives and contributions

The aim of this thesis is to develop and test fully scalable iterative solution methods for the incompressible MHD model (1.11), (1.12) using natural Taylor-Hood elements [59] for the fluid variables and Nédélec mixed element [47] pair for the magnetic variables. Our numerical results show good scalability with respect to the mesh size. We provide several tests to study the performance of the preconditioners with respect to the relevant non-dimensional parameters. We also present two- and three-dimensional results.

To enable large scale preconditioned tests of the MHD model we use the finite element software `FEniCS` [40] together with the linear algebra software from `PETSc` [9, 10]. Using these two principal software packages, experiments were run in excess of 20 million degrees of freedom. As well, it provides an example of how significant physical problems described by partial differential

equations can be solved by combining state of the art numerical software packages. The aim is to release the code for public use.

As stated above, little has been done in terms of preconditioned iterations for the MHD model other than the works of [50] for the non-multiplier form (1.9). Our approach is based on $H(\text{curl})$ elements for the magnetic field [55], and motivated by the preliminary results of [35] in the context of exactly divergence-free elements for the velocity field. It combines preconditioners for the incompressible Navier-Stokes and Maxwell's equations in [18, 27, 36].

The availability of our large-scale solvers and code will hopefully allow more development and research into such MHD models.

1.4 Outline

This thesis is made up of five chapters and is structured as follows. In Chapter 2, we introduce a mixed finite element approximation to the MHD system (1.11)-(1.12). The mixed approximation is based on a standard nodal Taylor-Hood finite element approximation for the velocity field and the pressure, together with a mixed Nédélec element approximation for the magnetic field and the multiplier. Using this approximation, we introduce three possible non-linear iteration schemes.

In Chapter 3, we present an overview of the preconditioning approaches for the individual subproblems separately, namely the incompressible Navier-Stokes and Maxwell's equations. We then apply these preconditioning techniques to propose preconditoning strategies for the linearisations which arise from the three non-linear iteration schemes from Chapter 2. In

particular, for the coupled linearised Picard scheme we propose an inner-outer preconditioning approach.

In Chapter 4, to perform numerical experiments in both two and three spatial dimensions we use the following two main software packages `FEniCS` [40] and `PETSc` [9, 10]. We show convergence results for the linearised MHD system along with the incompressible Navier-Stokes and Maxwell subproblems in isolation. Along with the convergence results we numerically test the preconditioning approaches for the three non-linear iteration schemes, providing heuristic tests with respect to the dimensionless parameters (ν, ν_m and κ; see (1.11)) and mesh size. These tests examine the robustness of both the preconditioners and the iteration schemes.

Chapter 5 provides conclusions and outlines possible extensions for future work.

Chapter 2

Finite element discretisation

In this chapter we introduce a mixed finite element discretisation for the steady-state incompressible MHD problem (1.11), (1.12) that models electrically conductive fluids under the influence of a magnetic field. Following the setting in [55], we use curl-conforming elements for the magnetic field and conforming continuous elements for the velocity field. The resulting discretisation is verified though a series of numerical experiments which appear later in Chapter 4. For simplicity, we initially only discuss in detail homogeneous Dirichlet boundary conditions, that is

$$\boldsymbol{u} = \mathbf{0} \quad \text{and} \quad \boldsymbol{n} \times \boldsymbol{b} = \mathbf{0}. \tag{2.1}$$

Inhomogeneous conditions as in (1.12) are discussed in Section 2.5.

2.1 Variational formulation

Suppose that the domain Ω is a Lipschitz domain of $\mathbb{R}^d$ for $d = 2, 3$. To express the problem (1.11), (1.12) in weak form we follow [55] and denote the L^2-inner product on $L^2(\Omega)^d$ by $(\cdot, \cdot)_\Omega$, for $d = 2, 3$. We introduce the

standard Sobolev spaces

$$
\begin{aligned}
&\boldsymbol{V} = H_0^1(\Omega)^d = \left\{ \boldsymbol{u} \in H^1(\Omega)^d \,:\, \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega \right\}, \\
&Q = L_0^2(\Omega) = \{ p \in L^2(\Omega) \,:\, (p\,, 1)_\Omega = 0 \}, \\
&\boldsymbol{C} = H_0(\mathrm{curl}; \Omega) = \left\{ \boldsymbol{b} \in L^2(\Omega)^d \,:\, \nabla \times \boldsymbol{b} \in L^2(\Omega)^{\bar{d}},\ \boldsymbol{n} \times \boldsymbol{b} = \boldsymbol{0} \text{ on } \partial\Omega \right\}, \\
&S = H_0^1(\Omega) = \{ r \in H^1(\Omega) \,:\, r = 0 \text{ on } \partial\Omega \}.
\end{aligned}
\tag{2.2}
$$

where $\bar{d} = 2d - 3$ is used to cover the 2D and 3D cases. We write $\|\cdot\|_{L^2(\Omega)}$, $\|\cdot\|_{H^1(\Omega)}$ and $\|\cdot\|_{H(\mathrm{curl};\Omega)}$ for the associated natural norms. More precisely, for vector fields $\boldsymbol{u}, \boldsymbol{b}$ and a scalar function r the norms are defined as follows:

$$
\begin{aligned}
\|\boldsymbol{u}\|_{L^2(\Omega)} &= \left(\int_\Omega \boldsymbol{u} \cdot \boldsymbol{u} \, dx \right)^{\frac{1}{2}}, \\
\|\boldsymbol{u}\|_{H^1(\Omega)} &= \left(\|\boldsymbol{u}\|_{L^2(\Omega)}^2 + \|\nabla \boldsymbol{u}\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}, \\
\|\boldsymbol{b}\|_{H(\mathrm{curl},\Omega)} &= \left(\|\boldsymbol{b}\|_{L^2(\Omega)}^2 + \|\nabla \times \boldsymbol{b}\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}}, \\
\|r\|_{L^2(\Omega)} &= \left(\int_\Omega r^2 \, dx \right)^{\frac{1}{2}}, \\
\|r\|_{H^1(\Omega)} &= \left(\|r\|_{L^2(\Omega)}^2 + \|\nabla r\|_{L^2(\Omega)}^2 \right)^{\frac{1}{2}},
\end{aligned}
$$

where $\|\nabla \boldsymbol{u}\|_{L^2(\Omega)}^2$ is given by:

$$
\|\nabla \boldsymbol{u}\|_{L^2(\Omega)}^2 = \left(\int_\Omega \sum_{i,j=1}^{d} (\nabla \boldsymbol{u})_{ij} (\nabla \boldsymbol{u})_{ij} \, dx \right)^{\frac{1}{2}}.
$$

The weak formulation of the incompressible MHD system (1.11), (1.12) consists in finding $(\boldsymbol{u}, p, \boldsymbol{b}, r) \in \boldsymbol{V} \times Q \times \boldsymbol{C} \times S$ such that

$$\begin{aligned} A(\boldsymbol{u},\boldsymbol{v}) + O(\boldsymbol{u};\boldsymbol{u},\boldsymbol{v}) + C(\boldsymbol{b};\boldsymbol{v},\boldsymbol{b}) + B(\boldsymbol{v},p) &= (\boldsymbol{f},\boldsymbol{v})_\Omega, && \text{(2.3a)} \\ B(\boldsymbol{u},q) &= 0, && \text{(2.3b)} \\ M(\boldsymbol{b},\boldsymbol{c}) - C(\boldsymbol{b};\boldsymbol{u},\boldsymbol{c}) + D(\boldsymbol{c},r) &= (\boldsymbol{g},\boldsymbol{c})_\Omega, && \text{(2.3c)} \\ D(\boldsymbol{b},s) &= 0, && \text{(2.3d)} \end{aligned}$$

for all $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V} \times Q \times \boldsymbol{C} \times S$. The individual variational forms are given by

$$\begin{aligned} A(\boldsymbol{u},\boldsymbol{v}) &= \int_\Omega \nu\, \nabla\boldsymbol{u} : \nabla\boldsymbol{v}\, d\boldsymbol{x}, \\ O(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v}) &= \int_\Omega (\boldsymbol{w}\cdot\nabla)\boldsymbol{u}\cdot\boldsymbol{v}\, d\boldsymbol{x}, \\ B(\boldsymbol{u},q) &= -\int_\Omega (\nabla\cdot\boldsymbol{u})\, q\, d\boldsymbol{x}, \\ M(\boldsymbol{b},\boldsymbol{c}) &= \int_\Omega \kappa\nu_m (\nabla\times\boldsymbol{b})\cdot(\nabla\times\boldsymbol{c})\, d\boldsymbol{x}, \\ D(\boldsymbol{b},s) &= \int_\Omega \boldsymbol{b}\cdot\nabla s\, d\boldsymbol{x}, \\ C(\boldsymbol{d};\boldsymbol{v},\boldsymbol{b}) &= \int_\Omega \kappa\,(\boldsymbol{v}\times\boldsymbol{d})\cdot(\nabla\times\boldsymbol{b})\, d\boldsymbol{x}, \end{aligned} \tag{2.4}$$

where $\nabla\boldsymbol{u} : \nabla\boldsymbol{v}$ is defined as

$$\nabla\boldsymbol{u} : \nabla\boldsymbol{v} = \sum_{i,j=1}^{d} (\nabla\boldsymbol{u})_{ij}(\nabla\boldsymbol{v})_{ij}.$$

In [55] it has been shown that this formulation of the problem is discretely energy-stable and has a unique solution for small data (i.e. for small ν, ν_m,

κ and forcing terms $\boldsymbol{f}$ and $\boldsymbol{g}$ with small L^2-norms).

2.2 Mixed finite element discretisation

Consider the domain Ω to be divided up into a regular and quasi-uniform mesh $\mathcal{T}_h = \{K\}$ consisting of triangles ($d = 2$) or tetrahedra ($d = 3$) with mesh size h. Based on the function spaces defined in (2.2), our finite element approximation will be sought in the finite dimensional spaces given by:

$$\begin{aligned}
\boldsymbol{V}_h &= \{\, \boldsymbol{u} \in H^1(\Omega) \,:\, \boldsymbol{u}|_K \in \mathcal{P}_k(K)^d,\, K \in \mathcal{T}_h \,\}, \\
Q_h &= \{\, p \in L^2(\Omega) \cap H^1(\Omega) \,:\, p|_K \in \mathcal{P}_{k-1}(K),\, K \in \mathcal{T}_h \,\}, \\
\boldsymbol{C}_h &= \{\, \boldsymbol{b} \in H_0(\mathrm{curl};\Omega) \,:\, \boldsymbol{b}|_K \in \mathcal{P}_{k-1}(K)^d \oplus \boldsymbol{R}_k(K),\, K \in \mathcal{T}_h \,\}, \\
S_h &= \{\, r \in H_0^1(\Omega) \,:\, r|_K \in \mathcal{P}_k(K),\, K \in \mathcal{T}_h \,\},
\end{aligned} \tag{2.5}$$

for $k \geq 2$. We define $\mathcal{P}_k(K)$ as the space of polynomials of total degree at most k on K and $\boldsymbol{R}_k(K)$ as the space of homogeneous vector polynomials of total degree k on K that are orthogonal to the position vector $\boldsymbol{x}$. Here we note that we are using $\mathcal{P}_k/\mathcal{P}_{k-1}$ Taylor-Hood elements for the fluid unknowns $(\boldsymbol{u}, p)$ [59]. For the magnetic variables $(\boldsymbol{b}, r)$ we use the curl-conforming Nédélec element pair of the first kind [47]. These choices of finite elements spaces $\boldsymbol{V}_h$, $\boldsymbol{C}_h$, Q_h and S_h imply that we have conforming subspaces to our Sobolev spaces $\boldsymbol{V}$, $\boldsymbol{C}$, Q and S, respectively. Then the finite element solution to (2.3) consists in finding $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$

such that

$$
\begin{aligned}
A(\boldsymbol{u}_h, \boldsymbol{v}) + \tilde{O}(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}) + C(\boldsymbol{b}_h; \boldsymbol{v}, \boldsymbol{b}_h) + B(\boldsymbol{v}, p_h) &= (\boldsymbol{f}, \boldsymbol{v}), && \text{(2.6a)} \\
B(\boldsymbol{u}_h, q) &= 0, && \text{(2.6b)} \\
M(\boldsymbol{b}_h, \boldsymbol{c}) - C(\boldsymbol{b}_h; \boldsymbol{u}_h, \boldsymbol{c}) + D(\boldsymbol{c}, r_h) &= (\boldsymbol{g}, \boldsymbol{c}), && \text{(2.6c)} \\
D(\boldsymbol{b}_h, s) &= 0, && \text{(2.6d)}
\end{aligned}
$$

for all $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$.

The forms A, M, B, D and C stay the same as on the continuous level. However, for the convection term $\tilde{O}(\cdot;\cdot,\cdot)$ we modify the form $O(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v})$ in a standard fashion to ensure the energy-stability property

$$
\tilde{O}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{u}) = 0, \quad \forall \boldsymbol{w}, \boldsymbol{u} \in \boldsymbol{V}_h. \tag{2.7}
$$

To do so we integrate by parts the convection form $O(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{u})$ to obtain

$$
\int_\Omega (\boldsymbol{w} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{u} \, d\boldsymbol{x} = -\frac{1}{2} \int_\Omega \nabla \cdot \boldsymbol{w}\boldsymbol{u} \cdot \boldsymbol{u} \, d\boldsymbol{x} + \frac{1}{2} \int_{\partial\Omega} \boldsymbol{w} \cdot \boldsymbol{n} |\boldsymbol{u}|^2 \, ds,
$$

recalling that $\boldsymbol{n}$ is the unit outward normal on $\partial\Omega$. Therefore, we choose the modified convection form $\tilde{O}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v})$ as

$$
\tilde{O}(\boldsymbol{w}; \boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (\boldsymbol{w} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{v} \, d\boldsymbol{x} + \frac{1}{2} \int_\Omega \nabla \cdot \boldsymbol{w}\boldsymbol{u} \cdot \boldsymbol{v} \, d\boldsymbol{x} - \frac{1}{2} \int_{\partial\Omega} \boldsymbol{w} \cdot \boldsymbol{n}\boldsymbol{u} \cdot \boldsymbol{v} \, ds.
$$

By construction, property (2.7) is now satisfied. Note also that for homogeneous boundary conditions as assumed in (2.1), the boundary integral term

in $\tilde{O}$ can be omitted.

Again in [55] it has been shown that this variational formulation of a MHD problem is discretely energy-stable and has a unique solution for small data. Also, optimal order error estimates in the mesh size h have been derived for small data using the stability property (2.7). Namely, for sufficiently smooth solutions, we have the error bound

$$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}+\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(\mathrm{curl};\Omega)}+\|p-p_h\|_{L^2(\Omega)}+\|r-r_h\|_{H^1(\Omega)}\leq Ch^k,$$

for a constant $C>0$ independent of the mesh size. In addition, the L^2-norm error for the velocity field is of order $\mathcal{O}(h^{k+1})$ (as $\boldsymbol{V}_h$ consists of a full polynomial space on each element). However, we cannot expect L^2-norm errors of order $\mathcal{O}(h^{k+1})$ for the magnetic field (as $\boldsymbol{C}_h$ does not consist of a full polynomial space on each element).

2.2.1 Matrix representation

The variational formulation (2.6) now can be converted into a matrix representation. To do this, we introduce the basis function for the finite element spaces in (2.5):

$$\boldsymbol{V}_h=\mathrm{span}\langle\boldsymbol{\psi}_j\rangle_{j=1}^{n_u},\qquad Q_h=\mathrm{span}\langle\alpha_i\rangle_{i=1}^{m_u},\tag{2.8}$$

$$\boldsymbol{C}_h=\mathrm{span}\langle\boldsymbol{\phi}_j\rangle_{j=1}^{n_b},\qquad S_h=\mathrm{span}\langle\beta_i\rangle_{i=1}^{m_b}.\tag{2.9}$$

The aim now is to find the coefficient vectors $u=(u_1,\ldots,u_{n_u})\in\mathbb{R}^{n_u}$, $p=(p_1,\ldots,p_{m_u})\in\mathbb{R}^{m_u}$, $b=(b_1,\ldots,b_{n_b})\in\mathbb{R}^{n_b}$, and $r=(r_1,\ldots,r_{m_b})\in\mathbb{R}^{m_b}$

of the finite element functions $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$ in terms of the chosen bases. As usual, this is done by writing the bilinear forms in (2.6) in terms of the following stiffness matrices and load vectors:

$$\begin{aligned}
A_{i,j} &= A(\boldsymbol{\psi}_j, \boldsymbol{\psi}_i), && 1 \le i, j \le n_u, \\
B_{i,j} &= B(\boldsymbol{\psi}_j, \alpha_i), && 1 \le i \le m_u,\ 1 \le j \le n_u, \\
D_{i,j} &= D(\boldsymbol{\phi}_j, \beta_i), && 1 \le i \le m_b,\ 1 \le j \le n_b, \\
M_{i,j} &= M(\boldsymbol{\phi}_j, \boldsymbol{\phi}_i), && 1 \le i, j \le n_b, \\
f_i &= (\boldsymbol{f}, \boldsymbol{\psi}_i)_\Omega, && 1 \le i \le n_u, \\
g_i &= (\boldsymbol{g}, \boldsymbol{\phi}_i)_\Omega, && 1 \le i \le n_b.
\end{aligned}$$

For the two non-linear forms, $\tilde{O}$ and C, we define the corresponding stiffness matrices with respect to given finite element functions $\boldsymbol{w}_h \in \boldsymbol{V}_h$ and $\boldsymbol{d}_h \in \boldsymbol{C}_h$ in the first argument and their associated coefficient vectors w and d as

$$\begin{aligned}
O(w)_{i,j} &= \tilde{O}(\boldsymbol{w}_h; \boldsymbol{\psi}_j, \boldsymbol{\psi}_i), \quad 1 \le i, j \le n_u, \\
C(d)_{i,j} &= C(\boldsymbol{d}_h; \boldsymbol{\psi}_j, \boldsymbol{\phi}_i), \quad 1 \le i \le n_b,\ 1 \le j \le n_u.
\end{aligned}$$

Thus, the numerical solution to (1.11) consists in solving the non-linear

system

$$\begin{pmatrix} A+O(u) & B^T & C^T(b) & 0 \\ B & 0 & 0 & 0 \\ -C(b) & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ g \\ 0 \end{pmatrix}, \tag{2.10}$$

where the vectors $u \in \mathbb{R}^{n_u}$, $p \in \mathbb{R}^{m_u}$, $b \in \mathbb{R}^{n_b}$, and $r \in \mathbb{R}^{m_b}$ are the unknown coefficients of the finite element functions.

2.3 Picard iteration (P)

The discrete system (2.10) is non-linear, and therefore appling a non-linear solver to this problem is necessary. A common choice to deal with the non-linearity within the incompressible Navier-Stokes equations in isolation is to perform Oseen or Picard iterations [18]. This involves linearising around the current velocity and solving for updates.

For simplicity we only consider the linearly convergent Picard iterations. Since we have modified the convection form to be discretely energy-stable as in (2.7), an advantage of this approach is that the discrete convection-diffusion operator is real positive. Thus, with small data the fixed-point/Picard iteration is a contraction. A more efficient non-linear solver is Newton's method which converges quadratically near the solution. However, applying Newton's method is more involved, as it requires construction and solving linear systems associated with a Jacobian as well as finding an initial guess sufficiently close to the solution. We leave the im-

plementation of Newton's method or other non-linear solvers as an area of possible future work (Section 5.2).

We adapt the fixed-point/Picard iterations to an approach for the full MHD system, where we linearise around the current velocity and magnetic fields. Given a current iterate $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$ we solve for updates $(\delta\boldsymbol{u}_h, \delta p_h, \delta\boldsymbol{b}_h, \delta r_h)$ and introduce the next iterate by setting:

$$\begin{aligned} \boldsymbol{u}_h &\to \boldsymbol{u}_h + \delta\boldsymbol{u}_h, \quad p_h \to p_h + \delta p_h, \\ \boldsymbol{b}_h &\to \boldsymbol{b}_h + \delta\boldsymbol{b}_h, \quad r_h \to r_h + \delta r_h. \end{aligned}$$

In variational form, the updates $(\delta\boldsymbol{u}_h, \delta p_h, \delta\boldsymbol{b}_h, \delta r_h) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$ are found by solving the Picard system (P):

$$\begin{aligned} A(\delta\boldsymbol{u}_h, \boldsymbol{v}) + \tilde{O}(\boldsymbol{u}; \delta\boldsymbol{u}_h, \boldsymbol{v}) + C(\boldsymbol{b}_h; \boldsymbol{v}, \delta\boldsymbol{u}_h) + B(\boldsymbol{v}, \delta p_h) &= R_u(\boldsymbol{u}_h, \boldsymbol{b}_h, p_h; \boldsymbol{v}), \\ B(\delta\boldsymbol{u}_h, q) &= R_p(\boldsymbol{u}_h; q), \\ M(\delta\boldsymbol{b}_h, \boldsymbol{c}) + D(\boldsymbol{c}, \delta r_h) - C(\boldsymbol{b}_h; \delta\boldsymbol{u}_h, \boldsymbol{v}) &= R_b(\boldsymbol{u}_h, \boldsymbol{b}_h, r_h; \boldsymbol{c}), \\ D(\delta\boldsymbol{b}_h, s) &= R_r(\boldsymbol{b}_h; s), \end{aligned}$$

for all $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$. Note that this system is linearised around $(\boldsymbol{u}_h, \boldsymbol{b}_h)$. The right-hand side linear forms correspond to the residual

at the current iteration $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$ defined by:

$$\begin{aligned} R_u(\boldsymbol{u}_h, \boldsymbol{b}_h, p_h; \boldsymbol{v}) &= (\boldsymbol{f}, \boldsymbol{v})_\Omega - A(\boldsymbol{u}_h, \boldsymbol{v}) - \tilde{O}(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}) \\ &\quad - C(\boldsymbol{b}_h; \boldsymbol{v}, \boldsymbol{b}_h) - B(\boldsymbol{v}, p_h), \\ R_p(\boldsymbol{u}_h; q) &= -B(\boldsymbol{u}_h, q), \\ R_b(\boldsymbol{u}_h, \boldsymbol{b}_h, r_h; \boldsymbol{c}) &= (\boldsymbol{g}, c)_\Omega - M(\boldsymbol{b}_h, \boldsymbol{c}) + C(\boldsymbol{b}_h; \boldsymbol{u}_h, \boldsymbol{c}) - D(\boldsymbol{c}, r_h), \\ R_r(\boldsymbol{b}_h; s) &= -D(\boldsymbol{b}_h, s), \end{aligned}$$

for all $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$.

In [55] it is shown that for small data the Picard iteration (P) will converge to the exact solution for any initial guess.

To formulate the variational form of the Picard iteration (P) in matrix form, let (u, p, b, r) be the coefficient vectors associated with $(\boldsymbol{u}_h, p_h, \boldsymbol{b}_h, r_h)$ and $(\delta u, \delta p, \delta b, \delta r)$ be the coefficient vectors of $(\delta \boldsymbol{u}_h, \delta p_h, \delta \boldsymbol{b}_h, \delta r_h)$. Then it can readily seen that the Picard iteration (P) amounts to solving the matrix system

$$\begin{pmatrix} A + O(u) & B^T & C(b)^T & 0 \\ B & 0 & 0 & 0 \\ -C(b) & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \\ \delta b \\ \delta r \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \\ r_b \\ r_r \end{pmatrix}, \tag{2.11}$$

with

$$\begin{aligned} r_u &= f - Au - O(u)u - C(b)^T b - B^T p, \\ r_p &= -Bu, \\ r_b &= g - Mu + C(b)b - D^T r, \\ r_r &= -Db. \end{aligned} \tag{2.12}$$

At each non-linear iteration, the right hand side vectors and matrices $O(u)$ and $C(b)$ in (2.12) and (2.11) respectively must be assembled with the solution coefficient vectors (u, p, b, r) of the current iterate. Here, the matrix A is symmetric positive definite (SPD), $O(u)$ is non-symmetric and $-C(b)$, $C(b)^T$ appear in a skew symmetric fashion. We also note that M is symmetric positive semidefinite (SPSD) with nullity m_b corresponding to the dimension of the scalar space S_h giving rise to the discrete gradients, see [27].

2.4 Decoupled iterations

The full MHD system (1.11), (1.12) is a coupled system consisting of the incompressible Navier-Stokes and Maxwell's equations, coupled through the non-linear skew symmetric coupling term $C(b)$. In addition, the convection term $O(u)$ is non-linear as well. These two terms make the numerical solution challenging. Therefore, if one or both of these terms is small then it may be possible to iterate explicitly. In particular if the coupling term, $C(b)$, is small then we may completely decouple the system into an incompressible Navier-Stokes problem and a Maxwell problem. The two resulting decoupling schemes are what we call Magnetic and Complete Decoupling

and are both described below. Note that unlike the Picard iteration, there is no small data guarantee that such iterations based on these decoupling schemes will converge; although we see convergence for reasonable values of the non-dimensional parameters.

2.4.1 Magnetic Decoupling (MD)

Consider first the situation where there is weak coupling within the system, that is when $C(b)$ is small. Then it may be possible to drop these terms to completely decouple the system into the two subproblems, the incompressibleNavier-Stokes and Maxwell's equations. We will call this approach Magnetic Decoupling (MD). Then the system(2.11) simplifies to

$$\begin{pmatrix} A+O(u) & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \\ \delta b \\ \delta r \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \\ r_b \\ r_r \end{pmatrix}, \tag{2.13}$$

with

$$\begin{aligned} r_u &= f - Au - O(u)u - C(b)^T b - B^T p, \\ r_p &= -Bu, \\ r_b &= g - Mu + C(b)b - D^T r, \\ r_r &= -Db. \end{aligned}$$

We iterate in the same fashion as the Picard iteration with the simpler matrix (2.13). From (2.13) we can see that the system is now completely decoupled. This enable us to solve each individual subproblem separately and possibly in parallel.

2.4.2 Complete Decoupling (CD)

For the second decoupling scheme, we again consider there to be weak coupling of the system but we also consider that the fluid equations are diffusion-dominated and hence can exclude the convection terms. This amounts to the system

$$\begin{pmatrix} A & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \\ \delta b \\ \delta r \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \\ r_b \\ r_r \end{pmatrix}, \tag{2.14}$$

with

$$\begin{aligned} r_u &= f - Au - O(u)u - C(b)^T b - B^T p, \\ r_p &= -Bu, \\ r_b &= g - Mu + C(b)b - D^T r, \\ r_r &= -Db. \end{aligned}$$

Again, we perform iterations in the same fashion as the Picard iteration. This is the simplest technique as it removes all non-linear terms in the iteration matrix and hence leaves the linear Stokes problem in the upper

$(1,1)$ block matrix.

2.5 Inhomogeneous Dirichlet boundary conditions and initial guess

When considering inhomogeneous Dirichlet boundary conditions as in (1.12), we still solve (2.11), (2.13) and (2.14) for the solution updates with homogeneous Dirichlet boundary conditions. Therefore, in this approach we must incorporate the inhomogeneous Dirichlet boundary conditions only within the initial guess.

To form a suitable initial guess, we solve the decoupled Stokes problem with the inhomogeneous boundary condition (1.12a):

$$\begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix}, \tag{2.15}$$

and then the non-symmetric Maxwell problem the with inhomogeneous boundary conditions (1.12b), (1.12c):

$$\begin{pmatrix} M - C(u) & D^T \\ D & 0 \end{pmatrix} \begin{pmatrix} b \\ r \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix}. \tag{2.16}$$

Here the term $C(u)$ corresponds the coupling term using u (the initial guess for the velocity field). We expect the inclusion of the coupling term to increase the accuracy of the initial guess because additional information of the problem is used.

The inhomogeneous Dirichlet boundary conditions are incorporated in a standard fashion by suitably modifying the matrix system. The outcome of this procedure is that the boundary data interpolation is only performed for the initial guess. Hence, the iterative solves for the initial guess must be run to a sufficient accuracy to ensure the accuracy of the discrete boundary conditions.

2.6 Summary

In this chapter we reviewed a mixed finite element approximation to the full MHD system given in (1.11) and (1.12). We followed the mixed approach outlined in [55] and expressed the MHD system in the matrix form (2.11). Using the Picard iteration (2.11) we introduced two possible decoupling schemes ((MD) and (CD)) which may be simpler to solve for. The performance of the resulting three non-linear iteration schemes depends on the values of the parameters κ, ν and ν_m. The next chapter will discuss possible preconditioning approaches for these systems.

In the sequel, we shall omit the dependence of $O(u)$ and $C(b)$ on u and b, respectively, and simply write O and C.

Chapter 3

Preconditioning

The linear system (2.11) is typically sparse and of large dimensions, hence to efficiently solve for it we use a preconditioned iterative approach as proposed in [35]. We start by reviewing some preconditioning strategies for the incompressible Navier-Stokes and Maxwell subproblems in isolation. From these techniques we will then introduce and numerically test preconditioners for the full MHD system in linearised form as in (2.11).

Let us introduce a number of matrices in addition to the ones already defined in Chapter 2: the velocity mass matrix Q, the pressure mass matrix Q_p, the pressure convection diffusion matrix F_p, the pressure Laplacian matrix A_p, the scalar Laplacian matrix L and the mass matrix X by

$$Q_{i,j} = \int_\Omega \boldsymbol{\psi}_j \cdot \boldsymbol{\psi}_i \, d\boldsymbol{x}, \quad 1 \le i,j \le n_u, \tag{3.1a}$$

$$(Q_p)_{i,j} = \int_\Omega \alpha_j \alpha_i \, dx, \quad 1 \le i,j \le m_u, \tag{3.1b}$$

$$(F_p)_{i,j} = \nu \int_\Omega \nabla\alpha_j \cdot \nabla\alpha_i + (\boldsymbol{w} \cdot \nabla\alpha_j)\alpha_i \, dx, \quad 1 \le i,j \le m_u, \tag{3.1c}$$

$$(A_p)_{i,j} = \int_\Omega \nabla\alpha_j \cdot \nabla\alpha_i \, dx, \quad 1 \le i,j \le m_u, \tag{3.1d}$$

$$L_{i,j} = \int_\Omega \nabla\beta_j \cdot \nabla\beta_i \, d\boldsymbol{x}, \quad 1 \le i,j \le m_b \tag{3.1e}$$

$$X_{i,j} = \int_\Omega \boldsymbol{\phi}_j \cdot \boldsymbol{\phi}_i \, d\boldsymbol{x}, \quad 1 \le i,j \le n_b, \tag{3.1f}$$

where $\boldsymbol{w}$ is the finite element velocity from the current iteration. The matrices F_p and A_p are well defined as we use continuous elements for the pressure finite element space Q_h. The matrices incorporate homogeneous Neumann boundary conditions.

3.1 Preconditioning the incompressible Navier-Stokes equations

Consider the steady state incompressible Navier-Stokes equations in isolation. Let

$$\mathcal{K}_{\mathrm{NS}} = \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix}, \tag{3.2}$$

be the discretised and linearised Navier-Stokes (or Oseen) subproblem where $F = A + O$ has been defined in (2.4). Due to the convection term, O, this system is non-symmetric and we will use GMRES to iteratively solve this subproblem; [54]. A common approach for solving a saddle point system, as in (3.2), is to use a block diagonal or block triangular preconditioner of the form

$$\mathcal{M}_{\mathrm{tNS}} = \begin{pmatrix} F & B^T \\ 0 & -S \end{pmatrix} \quad \text{or} \quad \mathcal{M}_{\mathrm{dNS}} = \begin{pmatrix} F & 0 \\ 0 & S \end{pmatrix}, \tag{3.3}$$

where $S \approx BF^{-1}B^T$ is an approximation to the Schur complement. If $S = BF^{-1}B^T$ is *precisely* the Schur complement, then it has been proved in [46] that the preconditioned matrix has exactly 2 eigenvalues when using the block triangular preconditioner (i.e., $\mathcal{M}_{\mathrm{tNS}}^{-1}\mathcal{K}_{\mathrm{NS}}$) or 3 distinct eigenvalues in the block diagonal case (i.e., $\mathcal{M}_{\mathrm{dNS}}^{-1}\mathcal{K}_{\mathrm{NS}}$). Both are diagonalisable and

hence GMRES will converge in exactly 2 or 3 iterations in the absence of round-off errors.

In practice it is often too expensive to form and solve for the exact Schur complement $BF^{-1}B^T$, hence, a good approximation is needed. Two well-known preconditioners for the incompressible Navier-Stokes equations are the Least Squares Commutator (LSC) and the Pressure Convection-Diffusion (PCD) preconditioners. A description of both can be found in [18] and we will just outline the procedure how these can be applied on the discrete level. For the Navier-Stokes preconditioner we will consider block triangular preconditioners of the same form as $\mathcal{M}_{\text{tNS}}$.

3.1.1 Pressure Convection-Diffusion (PCD)

In [18, Chap. 8] the discrete commutator of the convection-diffusion operator associated with the gradient operation is introduced and given by

$$\epsilon_h = (Q^{-1}F)(Q^{-1}B^T) - (Q^{-1}B^T)(Q_p^{-1}F_p). \tag{3.4}$$

where the matrices Q, Q_p and F_p are defined in (3.1). Assuming that the commutator is small then pre- and post-multiplying (3.4) by $BF^{-1}Q$ and $F_p^{-1}Q_p$, respectively, let us separate out the Schur complement to give

$$BF^{-1}B^T \approx BQ^{-1}B^TF_p^{-1}Q_p. \tag{3.5}$$

Our discretisation is inf-sup stable which implies that there is spectral equivalence between $BQ^{-1}B^T$ and the pressure Laplacian matrix, A_p; see

[18, Section 5.5.1]. Hence, the Schur complement can be approximated by:

$$S_{\text{PCD}} = A_p F_p^{-1} Q_p.$$

Applying the PCD preconditioner (i.e., taking $S = S_{\text{PCD}}$) to the linearised Navier-Stokes system involves solving the system

$$\begin{pmatrix} F & B^T \\ 0 & -A_p F_p^{-1} Q_p \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

at each Krylov iteration. This can be solved efficiently by splitting it into the following two steps

1. Solve for y: $y = -Q_p^{-1} F_p A_p^{-1} b$;
2. Solve for x: $x = F^{-1}(a - B^T y)$.

This means that we have one pressure Poisson solve (A_p^{-1}), one mass matrix solve (Q_p^{-1}), one convection-diffusion solve (F^{-1}) and multiplications with F_p and B^T at each Krylov iteration. It is possible to solve these iteratively using multigrid and/or iterative methods. However, to test the preconditioner we will use direct solver in this thesis.

3.1.2 Least-Squares Commutator (LSC)

One disadvantage with using the PCD preconditioner is that it requires the construction of the matrices A_p, Q_p and F_p in (3.1). A second approach to approximate the Schur complement is the LSC preconditioner [18, Chap. 8] which primarily uses the available matrix coefficients in (3.2) and the

construction of Q to form the preconditioner (without explicitly forming A_p, Q_p and F_p).

As for the derivation of the PCD preconditioner we start off with the discrete commutator of the convection-diffusion operator

$$\epsilon_h = (Q^{-1}F)(Q^{-1}B^T) - (Q^{-1}B^T)(Q_p^{-1}F_p).$$

Suppose that the Q-norm is defined by $\|v\|_Q = (Qv, v)^{1/2}$. Then this time we minimise ϵ_h over the jth column of F_p (that is $[F_p]_j$) in the Q-norm to try to find an approximation for F_p. As shown in [18], the minimisation is given by

$$\min \|[Q^{-1}FQ^{-1}B^T]_j - Q^{-1}B^TQ_p^{-1}[F_p]_j)\|_Q.$$

Solving this optimisation problem, as shown in [18], is done by solving the normal equations

$$Q_p^{-1}BQ^{-1}B^TQ_p^{-1}F_p = Q_p^{-1}BQ^{-1}FQ^{-1}B^T.$$

This yields the an approximation to F_p as

$$F_p \approx Q_p(BQ^{-1}B^T)^{-1}(BQ^{-1}FQ^{-1}B^T).$$

By substituting this into expression (3.5) we obtain the LSC approximation to the Schur complement:

$$S \approx BF^{-1}B^T \approx S_{\text{LSC}} = (BQ^{-1}B^T)(BQ^{-1}FQ^{-1}B^T)^{-1}(BQ^{-1}B^T).$$

Therefore, applying the LSC preconditioner to the Oseen system $\mathcal{K}_{\text{NS}}$ in (3.2) involves solving for the matrix

$$\begin{pmatrix} F & B^T \\ 0 & -S_{\text{LSC}} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix}$$

at each Krylov iteration. Again, this can be efficiently split up into the following two steps:

1. Solve for y: $y = -(BQ^{-1}B^T)^{-1}(BQ^{-1}FQ^{-1}B^T)(BQ^{-1}B^T)^{-1}b$;
2. Solve for x: $x = F^{-1}(a - B^T y)$.

Hence, we have two pressure Poisson solves ($(BQ^{-1}B^T)^{-1}$) and one convection-diffusion solve (F^{-1}) at each Krylov iteration as well as matrix multiplications. In practice, we take the diagonal or lumped diagonal of Q to form $BQ^{-1}B^T$. These solves, as with the PCD preconditioner, will be done directly in this thesis.

3.1.3 PCD versus LSC

The main advantage of solving the commutator using the least-squares approach is that the matrices that define the preconditioner are available from the original system $\mathcal{K}_{\text{NS}}$ in (3.2) and the construction of Q . However, for PCD we require the construction of the matrices A_p, Q_p and F_p. Therefore, LSC is slightly more computationally efficient to form. On the other hand, to apply the LSC preconditioner we require two pressure Poisson solves, whereas PCD only requires one.

We will consider experiments with both preconditioners for the incompressible Navier-Stokes problem in isolation to determine which seems more effective. This preconditioner will then be applied within the solver for the linearised MHD system.

3.2 Preconditioning Maxwell's equations

Next, consider the Maxwell subproblem

$$\mathcal{K}_{\mathrm{MX}} = \begin{pmatrix} M & D^T \\ D & 0 \end{pmatrix}, \tag{3.6}$$

appearing in a linearised MHD system (2.11). As we did for the Navier-Stokes subproblem in Section 3.1, we apply a block preconditioning strategy for $\mathcal{K}_{\mathrm{MX}}$ in (3.6). Notice that, $\mathcal{K}_{\mathrm{MX}}$ in (3.6) is symmetric and hence we will focus on SPD block diagonal preconditioners.

3.2.1 An ideal preconditioner

The $(1,1)$ block of $\mathcal{K}_{\mathrm{MX}}$ is the curl-curl operator, hence, the matrix M is singular, where the null space is of dimension m_b and consists of the discrete gradients. Therefore the usual Schur complement does not exist as the matrix M cannot be inverted. To overcome this difficulty, we employ the approach in [24, 26] based on augmentation and use preconditioners of the form:

$$\begin{pmatrix} M + D^T W^{-1} D & 0 \\ 0 & W \end{pmatrix},$$

where W is a symmetric positive definite matrix.

It has been shown in [27] that an appropriate choice of W is the scalar Laplacian, L, defined in (3.1). This leads to the ideal preconditioner:

$$\mathcal{M}_{\text{iMX}} = \begin{pmatrix} M + D^T L^{-1} D & 0 \\ 0 & L \end{pmatrix}. \tag{3.7}$$

Applying (3.7) as the preconditioner yields exactly two eigenvalues, 1 and -1 of algebraic multiplicities of n_b and m_b, respectively. Therefore using this matrix as a preconditioner means that MINRES will converge in two iterations, in the absence of roundoff errors [49]. However, forming the matrix $M + D^T L^{-1} D$ is costly, hence, $\mathcal{M}_{\text{iMX}}$ is impractical for large systems.

3.2.2 A practical preconditioner

A good approximation for $M + D^T L^{-1} D$ is required to make the ideal preconditioner, $\mathcal{M}_{\text{iMX}}$, suitable in practice. It has been shown in [27] that $M + D^T L^{-1} D$ is spectrally equivalent to $M + X$ where X is the vector mass matrix on the magnetic space defined in (3.1). Using this approximation yields the practical preconditioner

$$\mathcal{M}_{\text{MX}} = \begin{pmatrix} N & 0 \\ 0 & L \end{pmatrix}, \tag{3.8}$$

where $N = M + X$ is a shifted curl-curl operator. A scalable multigrid solver for N has been developed in [31] which involves one shifted Laplacian solves on the vector space and one scalar Laplacian solve. However, the

construction of this multigrid solver is involved and hence we will consider direct solves for this preconditioner and leave the multigrid implementation as a possible area of future work.

3.3 Preconditioning the MHD equations

In Section 2.3 and 2.4 we introduced three iteration schemes, namely Picard iteration (P), Magnetic Decoupling (MD) and Complete Decoupling (CD). Using the results from Sections 3.2 and 3.1 we will discuss the preconditioning approaches that we apply for these non-linear iteration schemes.

3.3.1 Picard iteration

Using the techniques of Sections 3.1 and 3.2 for the incompressible Navier-Stokes and Maxwell problems, respectively, we will look at possible scalable preconditioners for the linearised MHD problem

$$\mathcal{K}_{\mathrm{MH}} = \begin{pmatrix} A+O & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix}. \tag{3.9}$$

We propose the following preconditioner for $\mathcal{K}_{\mathrm{MH}}$

$$\mathcal{M}_{\mathrm{MH}} = \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -S & 0 & 0 \\ -C & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}, \tag{3.10}$$

where S is now either the LSC or PCD approximation to the fluid flow Schur complement. The preconditioned matrix, $\mathcal{M}_{\mathrm{MH}}^{-1}\mathcal{K}_{\mathrm{MH}}$, has eigenvalues $\lambda = 1$ with algebraic multiplicity of at least $n_u + n_b$ and the eigenvalue $\lambda = -1$ with algebraic multiplicity of at least m_b see [35, Theorem 8]. Due to the coupling terms, C, the application of this preconditioner is computationally expensive. To overcome this, we propose to invert $\mathcal{M}_{\mathrm{MH}}$ by means of an inner preconditioner. The inner preconditioner is taken as

$$\mathcal{M}_{\mathrm{innerMH}} = \begin{pmatrix} F & B^T & 0 & 0 \\ 0 & -S & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{3.11}$$

Here the preconditioned matrix, $\mathcal{M}_{\mathrm{innerMH}}^{-1}\mathcal{M}_{\mathrm{MH}}$, has an eigenvalue $\lambda = 1$ with algebraic multiplicity of at least $m_u + n_u + 3m_b - n_b$, see [35, Theorem 10].

3.3.2 Magnetic Decoupling

From Section 2.4.1 the matrix to be preconditioned for (MD) is

$$\mathcal{K}_{\mathrm{MD}} = \left(\begin{array}{cc|cc} F & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ \hline 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{array}\right). \tag{3.12}$$

Recall that removing the coupling terms completely decouples the system. This therefore enables us to use the preconditioners for each of the subproblems separately and in parallel. Using the subproblem preconditioners (3.3) and (3.8) then we propose the following preconditioner for $\mathcal{K}_{\mathrm{MD}}$:

$$\mathcal{M}_{\mathrm{MD}} = \left(\begin{array}{cc|cc} F & B^T & 0 & 0 \\ 0 & -S & 0 & 0 \\ \hline 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{array}\right), \tag{3.13}$$

where S is the LSC or PCD approximation and N is the shifted curl-curl matrix.

3.3.3 Complete Decoupling

From Section 2.4.2 the matrix to be preconditioned for (CD) is

$$\mathcal{K}_{\mathrm{CD}} = \left(\begin{array}{cc|cc} A & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ \hline 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{array}\right). \tag{3.14}$$

Note that the matrix $\mathcal{K}_{\mathrm{CD}}$ is now symmetric. First we consider how to deal with the upper $(2,2)$ block matrix which corresponds to the discrete Stokes equations

$$\mathcal{K}_{\mathrm{S}} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}.$$

As with the incompressible Navier-Stokes subproblem the idea for the Stokes preconditioner is again to approximate the Schur complement

$$S_{\mathrm{S}} = BA^{-1}B^T.$$

Recall that the matrix A is defined with the viscosity ν in Section 2.1. It was shown in [56, 61] that the scaled pressure mass matrix, $\frac{1}{\nu}W$ defined in (3.1), is spectrally equivalent to the Schur complement S_S (which is also a consequence of the inf-sup stability from our mixed discretisation). Therefore a possible scalable Stokes preconditioner is

$$\begin{pmatrix} A & 0 \\ 0 & \frac{1}{\nu}W \end{pmatrix}. \tag{3.15}$$

Using (3.15) together with the Maxwell subproblem preconditioner (3.8) gives the preconditioner

$$\mathcal{M}_{\mathrm{CD}} = \left(\begin{array}{cc|cc} A & 0 & 0 & 0 \\ 0 & \frac{1}{\nu}W & 0 & 0 \\ \hline 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{array}\right). \tag{3.16}$$

As the matrix system $\mathcal{K}_{\mathrm{CD}}$ is symmetric, then the appropriate choice for the Krylov subspace method is MINRES for each subproblem. The main advantage of using MINRES over GMRES is that we do not need to store a new vector at each iteration. Therefore, in terms of computational memory MINRES is more efficient.

3.3.4 Summary

In summary, we outlined the three preconditioning approaches for the linearised systems arising in the non-linear iteration schemes proposed in Chapter 2. Table 3.1 references the coefficient matrices together the associated preconditioner.

Iteration scheme	Coefficient matrix	(Outer) preconditioner
(P)	$\mathcal{K}_{\mathrm{MH}}$ in (3.9)	$\mathcal{M}_{\mathrm{MH}}$ in (3.10)
(MD)	$\mathcal{K}_{\mathrm{MD}}$ in (3.12)	$\mathcal{M}_{\mathrm{MD}}$ in (3.13)
(CD)	$\mathcal{K}_{\mathrm{CD}}$ in (3.14)	$\mathcal{M}_{\mathrm{CD}}$ in (3.16)

Table 3.1: Summary of coefficient matrices and corresponding preconditioners for each iteration scheme

Note that for the Picard iteration (P), we employ the inner preconditioner $\mathcal{M}_{\text{innerMH}}$ in (3.11) to solve systems corresponding to (3.10).

Chapter 4

Numerical results

In this chapter we present a series of convergence and preconditioning experiments. The principal aim is to check the scalability performance of the preconditioned iterative methods for the MHD problem and the two decoupling schemes proposed in Chapter 3. We have chosen to work with the Python programming language, due to its flexibility and the availability of a large collection of external libraries that this environment offers. Python is an extremely portable platform, which allows for a nearly seamless interface between multiple languages.

4.1 Software

Due to the complex nature of the MHD problem, a number of different libraries have been used both to discretise and then to solve the resulting systems.

The finite element software that was used to discretise (1.11), (1.12) is `FEniCS` [40]. The core libraries used within `FEniCS` are the problem-solving interface `DOLFIN` [42, 43], the compiler for finite element variational forms `FFC` [34, 41, 48], the finite element tabulator `FIAT` [32, 33] for creating finite element function spaces, the just-in-time compiler `Instant`, the code

generator UFC [2, 4] and the form language UFL [1, 3].

Along with FEniCS we have used a number of stand-alone linear algebra libraries. The main one that has been used is PETSc4PY, which is the Python interface for PETSc [9, 10]; it has been used mainly for the iterative solvers as well as the blockwise preconditioning setup. The following packages were used to solve the preconditioned system: HYPRE [19] as a multigrid solver and the sparse direct solvers UMFPACK [15, 16, 57, 58], PASTIX [29], SuperLU [37, 38] and MUMPS [5–7].

4.2 Problem setup

When numerically solving a set of linear or non-linear equations the problem needs to be initialised: the mesh sequence, non-linear iteration stopping criteria and initial guess tolerance need to be defined. In this section, we will briefly cover these aspects. For simplicity, all our numerical experiments are based on lowest-order elements (i.e., $k = 2$).

Mesh sequences

In our tests, we consider sequences of uniformly refined simplicial grids (i.e., triangles in 2D and tetrahedra in 3D). The mesh level ℓ defines the number of edges between the nodes along a single boundary edge to be 2^ℓ. For example, consider the domain to be the unit square then the third level ($\ell = 3$) generates the uniform 8×8 triangular grid given in Figure 4.1. The first column of all convergence and iteration tables below will show the grid level.

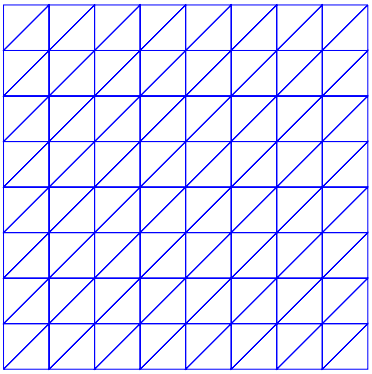

Figure 4.1: Level 3 grid on unit square domain

Stopping criteria

The incompressible Navier-Stokes equations as well as the full MHD problem are a set of non-linear equations. Section 2.3 outlines the process in which we linearise the problem and then solve for the updates. The stopping criteria that we enforce for the Navier-Stokes and MHD problems are:

- NS: $\|\delta u\|_2 + \|\delta p\|_2 < tol$,
- MHD: $\|\delta u\|_2 + \|\delta p\|_2 + \|\delta b\|_2 + \|\delta r\|_2 < tol$,

where $\|\cdot\|_2$ denotes the 2-norm of a vector. The tolerance is stated in each example table.

Note that the non-linear stopping tolerance that has been used varies from example to example. For the convergence results we set the tolerance

to be stricter to make sure the convergence rates are accurate. For the preconditioning tests the tolerance is loosened slightly for ease of computation.

Initial guess tolerance

To initialise the non-linear iterations, we follow Section 2.5. As discussed there, this involves iteratively solving Stokes equations (2.15) and a modified Maxwell problem of the form (2.16) to a sufficient accuracy. To that end, and throughout the chapter we choose the Krylov tolerance as 1e-10 for each problem. Inhomogeneous boundary conditions are enforced in a standard fashion within the corresponding finite element spaces, by interpolation at the boundary degrees of freedom.

The notation "Dofs" in the tables to follow stand for the number of degrees of freedom. We usually indicate which unknowns are meant but if not then "Dofs" refers to the total number of degrees of freedom of the system.

4.3 Numerical results: Navier-Stokes equations

Before considering the discretised MHD problem, we require that the Navier-Stokes subproblem perform as expected in terms of the error estimates and preconditioner scalability. To check the validity of the `FEniCS` code, we introduce two problems with smooth solutions, the first in 2D and the second in 3D. As mention in Section 4.2, we only provide results lowest-order ($k = 2$) Taylor-Hood elements; the discretisation in this case is stable and is sufficiently accurate for our purposes.

4.3.1 Convergence results for a 2D smooth solution

Consider the domain $\Omega = (0,1)^2$. Taking the viscosity to be $\nu = 1$, we choose the source terms $\boldsymbol{f}$ and inhomogeneous Dirichlet boundary conditions on $\partial\Omega$ from the analytical solution

$$\begin{aligned} \boldsymbol{u}(x,y) &= \begin{pmatrix} \sin(x)\exp(x+y) + \cos(y)\exp(x+y) \\ -\sin(y)\exp(x+y) \end{pmatrix}, \\ p(x,y) &= x^3\sin(y) + \exp(x+y). \end{aligned}$$

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}$	order
3	578/81	2.2100e-04	-	1.5645e-02	-
4	2,178/289	2.7598e-05	3.00	3.9133e-03	2.00
5	8,450/1,089	3.4465e-06	3.00	9.7822e-04	2.00
6	33,282/4,225	4.3072e-07	3.00	2.4455e-04	2.00
7	132,098/16,641	5.3836e-08	3.00	6.1137e-05	2.00
8	526,338/66,049	6.7291e-09	3.00	1.5284e-05	2.00

Table 4.1: Convergence of the velocity field for 2D Navier-Stokes - tol = 1e-10

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|p-p_h\|_{L^2(\Omega)}$	order
3	578/81	7.2758e-03	-
4	2,178/289	1.8190e-03	2.00
5	8,450/1,089	4.5025e-04	2.01
6	33,282/4,225	1.1341e-04	1.99
7	132,098/16,641	2.8639e-05	1.99
8	526,338/66,049	7.0146e-06	2.03

Table 4.2: Convergence of the pressure variable for 2D Navier-Stokes - tol = 1e-10

For the lowest-order Taylor-Hood elements the expected convergence

rates for the velocity field in the L^2- and H^1-norm errors are third and second order, respectively, and for the pressure the L^2-norm error is second order. This is in agreement with observed orders from Tables 4.1 and 4.2.

4.3.2 Convergence results for a 3D smooth solution

The three-dimensional set up is very similar as with the 2D case. The domain is the unit cube $\Omega = (0,1)^3$. As before, the viscosity is $\nu = 1$, then the source term $\boldsymbol{f}$ and inhomogeneous Dirichlet boundary conditions on $\partial\Omega$ are calculated from the analytical solution

$$\begin{aligned} \boldsymbol{u}(x,y,z) &= \begin{pmatrix} -\exp(x+y+z)\sin(y) + \exp(x+y+z)\sin(z) \\ \exp(x+y+z)\sin(x) - \exp(x+y+z)\sin(z) \\ -\exp(x+y+z)\sin(x) + \exp(x+y+z)\sin(y) \end{pmatrix}, \\ p(x,y,z) &= \exp(x+y+z) + \sin(y). \end{aligned}$$

Running the 3D code produces the results in Tables 4.3 and 4.4, which shows the computed errors and convergence rates.

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}$	order
1	375/27	2.6211e-02	-	4.5804e-01	-
2	2,187/125	3.2997e-03	2.99	1.1547e-01	1.99
3	14,739/729	4.1267e-04	3.00	2.8944e-02	2.00
4	107,811/4,913	5.1565e-05	3.00	7.2416e-03	2.00
5	823,875/35,937	6.4443e-06	3.00	1.8108e-03	2.00

Table 4.3: Convergence of the velocity field for 3D Navier-Stokes - tol = 1e105

We see that the convergence rates are the same as for the 2D test solution

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|p - p_h\|_{L^2(\Omega)}$	order
1	375/27	1.7725e+00	-
2	2,187/125	2.8602e-01	2.63
3	14,739/729	4.0587e-02	2.82
4	107,811/4,913	6.4794e-03	2.65
5	823,875/35,937	1.2724e-03	2.35

Table 4.4: Convergence of the pressure variable for 3D Navier-Stokes - tol = 1e-10

in the previous section for the velocity field. However, we observe that for the pressure we get a slightly higher than expected order, namely about 2.5.

4.3.3 Preconditioning with LSC and PCD

In this section, we check the performance of both the PCD and LSC preconditioners for the Navier-Stokes subproblem, see Sections 3.1.1 and 3.1.2. The problem setup is the 2D one from Section 4.3.1. We use GMRES as the Krylov subspace solver (with a convergence tolerance of 1e-5) and the application of the preconditioners will be done directly. Note that, we do not restart GMRES. The results tables show GMRES iterations averaged over all non-linear iterations.

We first consider the performance of the LSC preconditioner. In Table 4.5 we show the average number of GMRES iterations for various values of ν and grid levels. We observe that as the grid level increases the average number of GMRES iterations increases. For the smaller mesh levels ($\ell \leq 4$) the iteration increase is fairly small. However, for the higher levels the average GMRES iterations seems to roughly double as the level doubles. This is behaviour is similar to applying an incomplete factorisation preconditioner

ℓ	Dofs	Average iterations			
	$\boldsymbol{u}_h/p_h$	$\nu = 10$	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$
1	50/9	5	5	6	8
2	162/25	10	10	10	21
3	578/81	16	16	15	30
4	2,178/289	25	24	24	31
5	8,450/1,089	48	47	46	39
6	33,282/4,225	103	86	77	63
7	132,098/16,641	190	190	141	131

Table 4.5: Iteration table for LSC preconditioner for 2D example for various values of ν and tol = 1e-5

to the problem. Hence, LSC does not seem to yield a scalable preconditioner.

We next look at how PCD performs as a preconditioner for the Navier-Stokes subproblem. Applying this preconditioner to the incompressible Navier-Stokes problem in isolation gives the results in Table 4.6. We observe that as the mesh level increases, the average number of GMRES iteration numbers stays roughly constant with respect to the mesh level ℓ. This scalability is what we expect and require for the Navier-Stokes subproblem preconditioner. Also, note for fixed values of ν the number of average number of GMRES iterations remains about constant. We see that for $\ell = 5, 6$ and $\nu = 0.01$ the iterations seem to be slightly higher than expected. This behaviour is likely due to the mesh size being too big.

4.4 Numerical results: Maxwell's equations

In this section we consider 2D and 3D test problems to validate the `FEniCS` code for the Maxwell subproblem in mixed form.

ℓ	Dofs	Average iterations			
	$\boldsymbol{u}_h/p_h$	$\nu = 10$	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$
5	8,450/1,089	17	17	21	58
6	33,282/4,225	17	17	22	30
7	132,098/16,641	18	17	22	21
8	526,338/66,049	18	18	22	20
9	2,101,250/263,169	18	19	22	21

Table 4.6: Iteration table for a PCD preconditioned 2D example for various values of ν and tol = 1e-5

4.4.1 Convergence results for a 2D smooth solution

Again, consider the domain $\Omega = (0,1)^2$ with $\kappa = \nu_m = 1$. We choose the source term $\boldsymbol{g}$ and inhomogeneous Dirichlet boundary conditions on $\partial\Omega$ from the analytical solution

$$\begin{aligned} \boldsymbol{b}(x,y) &= \begin{pmatrix} \exp(x+y)\cos(x) \\ \exp(x+y)\sin(x) - \exp(x+y)\cos(x) \end{pmatrix}, \\ r(x,y) &= \sin(2\pi x)\sin(2\pi y). \end{aligned}$$

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$	order
1	48/25	9.3833e-02	-	1.0696e-01	-
2	176/81	2.3350e-02	2.01	2.7088e-02	1.98
3	672/289	5.8372e-03	2.00	6.4220e-03	2.08
4	2,624/1,089	1.4597e-03	2.00	1.4586e-03	2.14
5	10,368/4,225	3.6497e-04	2.00	3.5066e-04	2.06
6	41,216/16,641	9.1246e-05	2.00	8.6688e-05	2.02
7	164,352/66,049	2.2812e-05	2.00	2.1609e-05	2.00

Table 4.7: Convergence for 2D Maxwell smooth solution - magnetic field

For Maxwell's equations in mixed form we use second order ($k = 2$)

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|r - r_h\|_{L^2(\Omega)}$	order	$\|r - r_h\|_{H^1(\Omega)}$	order
1	48/25	2.7761e-01	-	2.8932e+00	-
2	176/81	4.3540e-02	2.67	9.3299e-01	1.63
3	672/289	4.8633e-03	3.16	2.5904e-01	1.85
4	2,624/1,089	5.6724e-04	3.10	6.6810e-02	1.96
5	10,368/4,225	6.9363e-05	3.03	1.6841e-02	1.99
6	41,216/16,641	8.6203e-06	3.01	4.2193e-03	2.00
7	164,352/66,049	1.0760e-06	3.00	1.0554e-03	2.00

Table 4.8: Convergence for 2D Maxwell smooth solution - multiplier variable

Nédélec elements of the first kind [47] and quadratic nodal elements for the magnetic field and multiplier, respectively. We expect second order convergence in the errors $\|\boldsymbol{b} - \boldsymbol{b}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{b} - \boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$ and $\|r - r_h\|_{H^1(\Omega)}$, and third order in $\|r - r_h\|_{L^2(\Omega)}$, see [44]. Tables 4.7 and 4.8 confirm that these rates are indeed obtained for both the magnetic field and the multiplier.

4.4.2 Convergence results for a 3D smooth solution

As we did for the 3D Navier-Stokes problem the domain is the unit cube, i.e. $\Omega = (0,1)^3$, $\kappa = \nu_m = 1$ and using the analytical solution

$$\begin{aligned} \boldsymbol{b}(x,y,z) &= \begin{pmatrix} -\exp(x+y+z)\sin(y) + \exp(x+y+z)\sin(z) \\ \exp(x+y+z)\sin(x) - \exp(x+y+z)\sin(z) \\ -\exp(x+y+z)\sin(x) + \exp(x+y+z)\sin(y) \end{pmatrix}, \\ r(x,y,z) &= \sin(2\pi x)\sin(2\pi y)\sin(2\pi z), \end{aligned}$$

the source term $\boldsymbol{g}$ and inhomogeneous Dirichlet boundary conditions on $\partial\Omega$ are defined.

Tables 4.9 and 4.10 show that the orders of convergence do asymptoti-

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$	order
1	436/125	5.3312e-02	-	3.5258e-01	-
2	2,936/729	1.4192e-02	1.91	8.9944e-02	1.97
3	21,424/4,913	3.5801e-03	1.99	2.2690e-02	1.99
4	163,424/35,937	8.9697e-04	2.00	5.7585e-03	1.98

Table 4.9: Convergence for 3D Maxwell smooth solution - magnetic field

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|r-r_h\|_{L^2(\Omega)}$	order	$\|r-r_h\|_{H^1(\Omega)}$	order
1	436/125	2.2689e-01	-	2.8923e+00	-
2	2,936/729	5.9452e-02	1.93	1.1782e+00	1.30
3	21,424/4,913	6.9068e-03	3.11	3.3901e-01	1.80
4	163,424/35,937	7.5913e-04	3.19	9.0082e-02	1.91

Table 4.10: Convergence for 3D Maxwell smooth solution - multiplier variable

cally approach the expected orders as in the 2D case from Section 4.4.1.

4.4.3 Preconditioning

Section 3.2 introduces the preconditioning strategy for Maxwell's equations that will be employed in this thesis. Two- and three-dimensional results are presented in this section.

2D example

Table 4.11 shows the performance of using the preconditioner $\mathcal{M}_{\mathrm{MX}}$ defined in (3.8) with a MINRES tolerance of 1e-6 and a direct application of the preconditioner for the 2D problem in Section 4.4.1.

From Table 4.11 we observe that the number of MINRES iterations stays about constant as the mesh level, ℓ, increases. Note that for a fixed magnetic viscosity (ν_m), the number of iterations remains roughly the same, and the

ℓ	Dofs $\boldsymbol{b}_h/r_h$	Number of iterations			
		$\nu_m = 10$	$\nu_m = 100$	$\nu_m = 1000$	$\nu_m = 10000$
5	10,368/4,225	5	4	6	6
6	41,216/16,641	5	6	6	6
7	164,352/66,049	5	6	6	6
8	656,384/263,169	5	6	6	8
9	2,623,488/1,050,625	4	6	6	8
10	10,489,856/4,198,401	4	6	8	10

Table 4.11: Iteration count for Maxwell preconditioner for 2D example - direct application of preconditioner

performance is fairly insensitive with respect to the values of ν_m.

3D example

Running the 3D code for the problem in Section 4.4.2 produces the results in Table 4.12. Again, the number of MINRES iterations stay roughly constant as the mesh level, ℓ, increases. We also note that the number of iterations seems again to be insensitive to the value of the magnetic viscosity, ν_m.

ℓ	Dofs $\boldsymbol{b}_h/r_h$	Number of iterations			
		$\nu_m = 10$	$\nu_m = 100$	$\nu_m = 1000$	$\nu_m = 10000$
2	2,936/729	4	4	6	5
3	21,424/49,13	4	4	6	5
4	163,424/35,937	4	6	6	5
5	1,276,096/274,625	4	6	6	5

Table 4.12: Iteration table for Maxwell preconditioner for 3D example - direct application of preconditioner

4.5 Numerical results: MHD problem

Sections 4.3 and 4.4 show results for the Navier-Stokes and Maxwell's equations in isolation. The next step is to incorporate these two subproblems into the full MHD system.

4.5.1 Convergence results for a 2D smooth solution

To validate the code, we consider the following 2D test problem: let the domain be the unit square $\Omega = (0,1)^2$. Take $\nu = \kappa = 1$, $\nu_m = 10$; then the source terms $\boldsymbol{f}$, $\boldsymbol{g}$ and inhomogeneous Dirichlet boundary conditions on $\partial\Omega$ are defined from the analytical solution:

$$\boldsymbol{u}(x,y) = \begin{pmatrix} xy\exp(x+y) + x\exp(x+y) \\ -xy\exp(x+y) - y\exp(x+y) \end{pmatrix},$$

$$p(x,y) = \exp(y)\sin(x),$$

$$\boldsymbol{b}(x,y) = \begin{pmatrix} \exp(x+y)\cos(x) \\ \exp(x+y)\sin(x) - \exp(x+y)\cos(x) \end{pmatrix},$$

$$r(x,y) = x\sin(2\pi x)\sin(2\pi y).$$

The convergence results computed in Tables 4.13 to 4.16 agree with the optimal rates for the errors $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{u} - \boldsymbol{u}_h\|_{H^1(\Omega)}$, $\|\boldsymbol{b} - \boldsymbol{b}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{b} - \boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$, $\|r - r_h\|_{L^2(\Omega)}$ and $\|r - r_h\|_{H^1(\Omega)}$. The pressure convergence rate exhibits a little higher than expected rate of convergence for $\ell < 5$ but eventually settles down to second order for the higher levels.

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}$	order
3	578/81	1.3357e-03	-	8.3244e-02	-
4	2,178/289	1.6752e-04	3.00	2.0840e-02	2.00
5	8,450/1,089	2.1029e-05	2.99	5.2110e-03	2.00
6	33,282/4,225	2.6675e-06	2.98	1.3028e-03	2.00
7	132,098/16,641	3.5200e-07	2.92	3.2570e-04	2.00
8	526,338/66,049	5.2267e-08	2.75	8.1425e-05	2.00

Table 4.13: Convergence for 2D MHD smooth solution (velocity field) - tol = 1e-8

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|p-p_h\|_{L^2(\Omega)}$	order
3	578/81	4.9329e-03	-
4	2,178/289	7.4197e-04	2.73
5	8,450/1,089	1.5604e-04	2.25
6	33,282/4,225	1.5604e-04	2.25
7	132,098/16,641	3.7738e-05	2.05
8	526,338/66,049	2.3415e-06	2.00

Table 4.14: Convergence for 2D MHD smooth solution (pressure variable) - tol = 1e-8

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$	order
3	672/289	7.1735e-03	-	1.1521e-02	-
4	2,624/1,089	1.7944e-03	2.00	2.8833e-03	2.00
5	10,368/4,225	4.4868e-04	2.00	7.2102e-04	2.00
6	41,216/16,641	1.1217e-04	2.00	1.8027e-04	2.00
7	164,352/66,049	2.8044e-05	2.00	4.5067e-05	2.00
8	656,384/263,169	7.0110e-06	2.00	1.1267e-05	2.00

Table 4.15: Convergence for 2D MHD smooth solution (magnetic field) - tol = 1e-8

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|r - r_h\|_{L^2(\Omega)}$	order	$\|r - r_h\|_{H^1(\Omega)}$	order
3	672/289	3.1956e-03	-	1.6711e-01	-
4	2,624/1,089	3.6326e-04	3.14	4.2565e-02	1.97
5	10,368/4,225	4.3989e-05	3.05	1.0692e-02	1.99
6	41,216/16,641	5.4517e-06	3.01	2.6763e-03	2.00
7	164,352/66,049	6.7998e-07	3.00	6.6928e-04	2.00
8	656,384/263,169	8.4951e-08	3.00	1.6733e-04	2.00

Table 4.16: Convergence for 2D MHD smooth solution (multiplier variable) - tol = 1e-8

4.5.2 Convergence results for a 3D smooth solution

Newt, we consider a 3D test problem discretised on the unit cube $\Omega = (0,1)^3$ with Dirichlet boundary conditions. Let $\nu = \kappa = 1$, $\nu_m = 10$ and let the analytical solution be given by:

$$\boldsymbol{u}(x,y,z) = \begin{pmatrix} -xy\exp(x+y+z) + xz\exp(x+y+z)) \\ xy\exp(x+y+z) - yz\exp(x+y+z) \\ -xz\exp(x+y+z) + yz\exp(x+y+z) \end{pmatrix},$$

$$p(x,y,z) = \exp(x+y+z)\sin(y),$$

$$\boldsymbol{b}(x,y,z) = \begin{pmatrix} -\exp(x+y+z)\sin(y) + \exp(x+y+z)\sin(z) \\ xy\exp(x+y+z) - yz\exp(x+y+z) \\ -\exp(x+y+z)\sin(x) + \exp(x+y+z)\sin(y) \end{pmatrix},$$

$$r(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z).$$

Then the source terms $\boldsymbol{f}$ and $\boldsymbol{g}$ and inhomogeneous boundary conditions are defined from the analytical solution.

From Tables 4.17 to 4.20 it can be seen that we obtain roughly the same

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}$	order
1	375/27	4.2196e-03	-	7.8282e-02	-
2	2,187/125	5.2733e-04	3.00	1.9495e-02	2.01
3	14,739/729	6.5749e-05	3.00	4.8664e-03	2.00
4	107,811/4,913	8.2092e-06	3.00	1.2161e-03	2.00

Table 4.17: Convergence for 3D MHD smooth solution (velocity field) - tol = 1e-8

ℓ	Dofs $\boldsymbol{u}_h/p_h$	$\|p-p_h\|_{L^2(\Omega)}$	order
1	375/27	5.0614e-02	-
2	2,187/125	9.0306e-03	2.49
3	14,739/729	1.9035e-03	2.25
4	107,811/4,913	4.5311e-04	2.07

Table 4.18: Convergence for 3D MHD smooth solution (pressure variable) - tol = 1e-8

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(\mathrm{curl},\Omega)}$	order
1	436/125	3.9931e-03	-	2.4817e-02	-
2	2,936/729	1.1211e-03	1.83	7.4101e-03	1.74
3	21,424/4,913	2.9642e-04	1.92	1.9439e-03	1.93
4	163,424/35,937	7.5494e-05	1.97	4.9180e-04	1.98

Table 4.19: Convergence for 3D MHD smooth solution (magnetic field) - tol = 1e-8

ℓ	Dofs $\boldsymbol{b}_h/r_h$	$\|r-r_h\|_{L^2(\Omega)}$	order	$\|r-r_h\|_{H^1(\Omega)}$	order
1	436/125	7.4750e-04	-	1.0113e-02	-
2	2,936/729	9.2457e-05	3.02	2.9368e-03	1.78
3	21,424/4,913	1.1182e-05	3.05	7.7197e-04	1.93
4	163,424/35,937	1.3829e-06	3.02	1.9597e-04	1.98

Table 4.20: Convergence for 3D MDH smooth solution (multiplier variable) - tol = 1e-8

convergence orders as with the $2D$ case. The computed orders are not quite as clean as with the 2D results and in some cases the calculated order is only slightly lower than the expected rate. However, for mesh levels 3 and 4 the computed'' orders seem to converge to the expected rates.

4.5.3 Parameter tests

There are two different decoupling schemes considered in thesis ((MD) and (CD)). The convergence of the non-linear iterations (within some tolerance tol) is likely to be limited by the parameter setup of the problem, i.e., by the values of the fluid viscosity (ν), the magnetic viscosity (ν_m) and the coupling number (κ). In this section we will look at how these two schemes perform with respect to the Picard iteration (P) when fixing two parameters and varying the third. We will only look at varying κ and ν since ν_m appears within all three schemes. Also, if the non-linear iterations do not converge then this is denoted by "-" in the tables.

Viscosity test

As a first test we consider varying the fluid viscosity, ν, for tol = 1e-5, $\kappa = 1$ and $\nu_m = 10$. The results are shown in Table 4.21.

As the fluid viscosity (ν) decreases, the fluid flow equations (1.11a) and (1.11b) become more convection-dominated. Thus we see that the (CD) scheme breaks down for small ν, as in this decoupling scheme the convection term is taken explicitly. On the other hand, the Picard and (MD) schemes perform similarly. For $\nu = 0.0001$ all schemes break down and do not converge. This may be due to the stability of the convection discretisation

		$\nu = 1$			$\nu = 0.1$			$\nu = 0.01$			$\nu = 0.001$		
ℓ	Dofs	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)
3	1,620	5	7	15	8	10	-	12	14	-	40	40	-
4	6,180	5	7	15	8	10	-	12	14	-	40	40	-
5	24,132	5	7	15	8	10	-	12	14	-	19	19	-
6	95,364	5	7	15	8	10	-	12	14	-	17	19	-
7	379,140	5	7	15	8	10	-	12	14	-	17	19	-
8	1,511,940	5	7	15	8	10	-	12	14	-	17	19	-

Table 4.21: Number of non-linear iterations for various values of ν with tol = 1e-5, $\kappa = 1$ and $\nu_m = 10$.

or it may be that the solution is unstable for these Reynolds numbers.

Coupling number test

		$\kappa = 1$			$\kappa = 10$			$\kappa = 100$			$\kappa = 1000$		
ℓ	Dofs	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)
3	1,620	5	5	13	6	7	17	7	13	26	7	93	-
4	6,180	5	5	15	6	7	18	7	13	26	7	93	-
5	24,132	5	5	15	6	7	18	7	13	26	7	93	-
6	95,364	5	5	15	6	7	18	7	13	26	7	93	-
7	379,140	5	5	15	6	7	18	7	13	26	7	93	-
8	1,511,940	5	5	15	6	7	18	7	13	26	7	93	-

Table 4.22: Number of non-linear iterations for various values of κ with tol = 1e-5, $\nu = 1$ and $\nu_m = 10$.

The next parameter test examines the effects of the coupling terms in the three non-linear iteration schemes. We expect the Picard scheme to outperform the other schemes for large values of κ. The results in Table 4.22 show that this is indeed the case. For the 2D example in Section 4.5.1 with $\nu = 1$, $\nu_m = 10$, the (CD) scheme completely breaks down for $\kappa > 100$, whereas the number of non-linear iterations for (MD) stays roughly constant for each mesh level but is larger than for the Picard iteration. When $\kappa =$

1000, (MD) also converges more slowly. This is the point at which the Picard iteration (P) becomes the most viable option.

4.6 Preconditioned MHD problem

Section 3.3 introduced the three preconditioning techniques that we apply to each of the decoupling schemes and the Picard iteration. In this section we investigate in detail the performance of these three strategies.

In the subsequent tables we use the following notation:

- Its_{NL}: the number of non-linear iterations (used in (P), (MD) and (CD));
- Its_{O}: the average number of outer GMRES iterations (used in (P));
- Its_{I}: the average number of inner iterations (used in (P));
- Its_{NS}: the average number of iterations to solve the Navier-Stokes problem (used in (MD));
- Its_{S}: the average number of iterations to solve the Stokes problem (used in (CD));
- Its_{M}: the average number of iterations to solve the Maxwell problem (used in (MD) and (CD));
- Dofs: the total degrees of freedom of the system.
- Av solve time: the solve time (with out matrix assembly) averaged over all non-linear iterations

- Total time: the total solve and assembly time until the non-linear iterations converge

The problem setups are the same as Sections 4.5.1 and 4.5.2. The numerical tests presented in this section are two-dimensional unless otherwise stated. As mentioned in Chapter 3, the application of the preconditioners is be done through direct solves.

4.6.1 Picard iteration

In this section, we present preconditioning results for the Picard iteration (P) given in Section 3.3.1. Recall, this is the inner-outer preconditioning technique. The only difference between the outer preconditioner (3.10) and inner preconditioner (3.11) is the inclusion of the coupling terms C and $-C^T$. Therefore, it may be possible to relax the inner solve tolerance and still produce scalable results with a smaller number of inner solves. In tougher cases (e.g., large κ) we expect to have a trade off, that is, an inner tolerance that is too loose may degrade convergence. We start off with parameter tests similar to those in Section 4.5.3.

Preconditioned parameter tests

The first preconditioning parameter test we consider is again to see how the viscosity, ν, effects the preconditioners performance for $\kappa = 1$ and $\nu_m = 10$. For the preconditioner for the Picard iteration we apply the inner-outer preconditioning approach from Section 3.3.1. For the outer Krylov solver we use flexible GMRES (FGMRES), see [52], with a convergence tolerance

of 1e-6 and for the inner Krylov solver we use GMRES with a tolerance of 1e-6. The results are presented in Table 4.23.

		$\nu = 0.1$			$\nu = 1$			$\nu = 10$		
ℓ	Dofs	Its_{NL}	Its_{O}	Its_{I}	Its_{NL}	Its_{O}	Its_{I}	Its_{NL}	Its_{O}	Its_{I}
4	6,180	-	-	-	6	18.8	14.7	5	11.2	12.6
5	24,132	-	-	-	6	27.3	15.6	6	9.8	12.7
6	95,364	13	91.5	63.4	6	20.2	15.3	6	9.7	11.3
7	379,140	10	74.6	31.4	7	16.3	14.3	7	14.6	15.9
8	1,511,940	12	82.3	37.8	8	24.3	14.9	7	15.4	15.7

Table 4.23: Number of non-linear and average number of preconditioning iterations for various values of ν with tol = 1e-5, $\kappa = 1$ and $\nu_m = 10$.

We observe that for smaller values of ν (i.e., in the convection-dominated regime) the number of inner and outer iterations increases. The table shows that as the mesh level increases the number of inner and outer iterations stay roughly constant. We also note that for mesh levels 4 and 5 the iterations do not seem to converge for $\nu = 0.1$. This may be because we are not considering grid levels of sufficient size. The PCD approximation to the Schur complement becomes more accurate for when h decreases.

The second test we consider is to again vary the coupling number. As before, we use the outer convergence tolerance of 1e-6 and inner tolerance 1e-6 with the same Krylov solvers.

Table 4.24 shows that as the mesh level increases then the number of outer and inner GMRES iterations stays roughly constant. We note that for larger values of κ the number of outer iterations increases. We also see this trend with the inner iterations for $\kappa = 10$ and 100.

		$\kappa = 0.1$			$\kappa = 1$			$\kappa = 10$			$\kappa = 100$		
ℓ	Dofs	Its_{NL}	Its_{O}	Its_{I}	Its_{NL}	Its_{O}	Its_{I}	Its_{NL}	Its_{O}	Its_{I}	Its_{NL}	Its_{O}	Its_{I}
4	6,180	5	22.2	15.4	6	18.8	14.7	7	30.7	14.1	9	61.4	24.4
5	24,132	5	23.8	11.4	6	27.3	15.2	8	43.8	24.0	10	80.3	37.9
6	95,364	5	28.0	17.4	6	20.2	15.3	7	41.1	15.4	9	74.6	31.1
7	379,140	5	18.6	14.6	7	16.3	14.2	7	37.4	16.4	14	73.9	34.7
8	1,511,940	5	20.4	15.2	8	24.3	14.9	7	39.6	18.4	11	75.4	33.3

Table 4.24: Number of non-linear and average number of preconditioning iterations for various values of κ with tol = 1e-5, $\nu = 1$ and $\nu_m = 10$.

4.6.2 Magnetic Decoupling

This section will look at preconditioning results for the (MD) scheme outlined in Section 3.3.2. As with the Picard iterations results we will consider parameter tests to assess the robustness of this decoupling scheme. All tests considered here will have a GMRES stopping tolerance of 1e-6 for the Navier-Stokes subproblem and a MINRES stopping tolerance of 1e-6 for the Maxwell subproblem.

Preconditioned parameter tests

We will again start of by seeing how the fluid's viscosity affects the preconditioner. The results are shown in Table 4.25.

		$\nu = 0.01$			$\nu = 0.1$			$\nu = 1$			$\nu = 10$		
ℓ	Dofs	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}
5	24,132	13	360.3	3.3	8	24.1	3.3	5	22.0	3.4	4	22.0	3.5
6	95,364	13	105.2	3.3	8	25.1	3.4	5	21.2	3.4	4	20.8	3.5
7	379,140	13	71.5	3.4	8	25.3	3.3	5	21.2	3.4	4	21.0	3.5
8	1,511,940	13	65.3	3.4	8	25.9	3.4	5	21.4	3.4	4	21.3	3.5

Table 4.25: Number of non-linear iterations and average number of iterations to solve the Navier-Stokes and Maxwell's subproblem for the MD scheme with tol = 1e-4, $\kappa = 1$, $\nu = 1$ and $\nu_m = 10$.

From Table 4.25 we see that for $\nu = 0.01$ the Navier-Stokes solver converges slowly to the solution for the lower mesh levels ($\ell \leq 6$). A possible reason for this is that the PCD preconditioner for the Navier-Stokes equations does not always converge for small viscosities and small meshes (large mesh size h). However, we see that for $\nu = 0.1$, $\nu = 1$ and $\nu = 10$ both the Navier-Stokes and Maxwell's subproblems exhibit iterations independent of the mesh level.

		$\kappa = 0.1$			$\kappa = 1$			$\kappa = 10$			$\kappa = 100$		
ℓ	Dofs	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}	Its_{NL}	Its_{NS}	Its_{M}
5	24,132	4	22.1	4.5	5	22.0	3.4	10	21.4	2.3	-	21.7	2.2
6	95,364	4	21.5	4.5	5	21.2	3.4	10	21.1	2.3	-	21.5	2.3
7	379,140	4	21.5	4.8	5	21.2	3.4	10	21.1	2.4	-	21.6	2.2
8	1,511,940	4	21.5	4.8	5	21.4	3.4	10	21.1	2.4	-	21.7	2.2

Table 4.26: Number of non-linear iterations and average number of iterations to solve the Navier-Stokes and Maxwell's subproblem for the MD scheme with tol = 1e-4, $\nu = 1$ and $\nu_m = 10$.

Table 4.26 shows the iteration count of the Magnetic Decoupling scheme when the coupling number, κ, is varied. We see that as κ increases the number of non-linear iterations increases. In fact, for $\kappa = 100$ the scheme does not converge. If we recall Table 4.22 we note that for direct solves $\kappa = 100$ does converge to the solution. Therefore, it seems that the Krylov tolerance of 1e-6 is not tight enough for (MD) to converge.

Table 4.27 shows large scaled results for the (MD) scheme with up to 24 million degrees of freedom. For all mesh levels the Navier-Stokes and Maxwell solves take roughly 21 and 3 iterations each. We also note that for $\ell = 10$, the total solve time was a little under 5 and a half hours.

ℓ	Dofs	Av solve time	Total time	Its_{NL}	Its_{NS}	Its_{M}
5	24,132	0.5	7.3	5	22.0	3.4
6	95,364	2.5	30.4	5	21.2	3.4
7	379,140	13.1	134.7	5	21.2	3.4
8	1,511,940	69.8	627.3	5	21.4	3.4
9	6,038,532	407.7	3159.7	5	21.6	3.2
10	24,135,684	3022.1	19668.3	5	21.6	3.4

Table 4.27: Number of non-linear iterations and average number of iterations to solve the Navier-Stokes and Maxwell's subproblem for the MD scheme with tol = 1e-4, $\kappa = 1$, $\nu = 1$ and $\nu_m = 10$.

Preconditioned 3D results

Tests were also run in three-dimensions and the results are represented below. Table 4.28 lists the number of iterations to solve the Navier-Stokes and Maxwell subproblems. The results seem to be independent of the mesh level.

l	Dofs	Av solve time	Total time	Its_{NL}	Its_{NS}	Its_{M}
1	963	0.04	2.7	5	23.2	3.4
2	5977	0.20	17.2	5	34.2	3.0
3	41805	4.86	151.1	5	34.2	3.4
4	312,085	242.1	2222.8	5	32.4	3.4
5	2,410,533	30222.3	159032.8	5	30.8	3.2

Table 4.28: Number of non-linear iterations and average number of iterations to solve the Navier-Stokes and Maxwell's subproblem for the MD scheme with tol = 1e-4, $\kappa = 1$, $\nu = 1$ and $\nu_m = 10$ in 3D.

4.6.3 Complete Decoupling

Finally, we consider preconditioning the Complete Decoupling scheme presented in Section 3.3.3. As for (P) and (MD) we will consider varying both the fluid's viscosity as well as the coupling number. To solve the Stokes

block of (3.14) we will use MINRES, as the Stokes equations are symmetric. Results are shown with 1e-6 as the convergence tolerance for both the Stokes and Maxwell solvers for all tests.

Preconditioned parameter tests

Table 4.21 shows that the (CD) iteration scheme breaks down for $\nu = 0.1$. Therefore, we consider tests for $\nu = 1$ and $\nu = 10$ and the results are given in Table 4.29. From Table 4.29 we see that the number average iterations for

		$\nu = 1$			$\nu = 10$		
ℓ	Dofs	Its_{NL}	Its_{S}	Its_{M}	Its_{NL}	Its_{S}	Its_{M}
5	24,132	11	29.0	3.3	4	29.0	3.5
6	95,364	11	28.5	3.4	4	29.0	3.5
7	379,140	11	27.5	3.4	4	29.0	3.5
8	1,511,940	11	28.3	3.5	4	29.0	3.5

Table 4.29: Number of non-linear iterations and average number of iterations to solve the Stokes and Maxwell's subproblem for the CD scheme with tol = 1e-4 $\kappa = 1$, and $\nu_m = 10$.

the Stokes and Maxwell subproblem solves remains constant. This is to be expected as we are in a regime (large enough ν) where the (CD) iterations converge. We also note that for larger values of ν the number of non-linear iterations increases.

To finish the parameter tests in this thesis, we again consider varying the coupling number. The results in Table 4.30 shows that for $\kappa = 0.1$ to 10 the number of iterations to solve the Stokes and Maxwell's problems seem both independent of the mesh size and the coupling number. The number of non-linear iterations increase for larger κ, in fact, for $\kappa = 100$ the scheme does not converge to the solution. We again consult with Table 4.22 where

		$\kappa = 0.1$			$\kappa = 1$			$\kappa = 10$			$\kappa = 100$		
ℓ	Dofs	Its_{NL}	Its_{S}	Its_{M}	Its_{NL}	Its_{S}	Its_{M}	Its_{NL}	Its_{S}	Its_{M}	Its_{NL}	Its_{S}	Its_{M}
5	24,132	9	29.0	4.4	11	29.0	3.3	18	29.0	2.2	-	-	-
6	95,364	9	29.0	4.6	11	28.5	3.4	18	29.0	2.3	-	-	-
7	379,140	8	28.5	4.5	11	27.6	3.4	18	29.0	2.3	-	-	-
8	1,511,940	8	28.3	4.6	11	28.3	3.5	18	29.0	2.4	-	-	-

Table 4.30: Number of non-linear iterations and average number of iterations to solve the Stokes and Maxwells subproblem for the CD scheme with tol = 1e-4, $\nu = 1$ and $\nu_m = 10$.

a direct application of the coefficient matrices was considered and we see that the corresponding iteration test there does converge for $\kappa = 100$. We therefore speculate, as for (MD), that a Krylov tolerance of 1e-6 is not tight enough when $\kappa = 100$.

As with the (MD) scheme we test the (CD) iterations in a large scale manner. From Table 4.31 we see that for $l = 10$ the total time is approximately 7 hours 50 minutes where as the for the same level using (MD) it was 5 hours 30 minutes. This increase in time, even so the average solve time is less, is due the fact that (CD) requires 6 more non-linear iterations to converge to the solution.

ℓ	Dofs	Av solve time	Total time	Its_{NL}	Its_{S}	Its_{M}
5	24,132	0.4	7.7	11	29.0	3.3
6	95,364	2.6	38.8	11	28.5	3.4
7	379,140	13.0	181.7	11	27.5	3.4
8	1,511,940	66.5	888.0	11	28.3	3.5
9	6,038,532	358.0	4565.4	11	28.3	3.4
10	24,135,684	2335.7	28337.4	11	27.4	3.5

Table 4.31: Number of non-linear iterations and average number of iterations to solve the Stokes and Maxwell's subproblem for the CD scheme with tol = 1e-4, $\kappa = 1$, $\nu = 1$ and $\nu_m = 10$.

Preconditioned 3D results

The final test run in this thesis is a three-dimensional example using the (CD) scheme. The results are shown in Table 4.32. This table, as with the 2D results, show that the average iterations to solve Stokes and Maxwell subproblems are independent with respect to the mesh level.

ℓ	Dofs	Av solve time	Total time	Its_{NL}	Its_{S}	Its_{M}
1	963	0.03	1.8	6	30.0	3.5
2	5,977	0.20	9.9	6	45.7	3.2
3	41,805	4.32	89.2	6	43.0	2.8
4	312,085	214.3	1786.5	6	42.3	2.8
5	2,410,533	26954.4	165671.1	6	41.3	2.8

Table 4.32: Number of non-linear iterations and average number of iterations to solve the Stokes and Maxwell's subproblem for the CD scheme with tol = 1e-4, $\kappa = 1$, $\nu = 1$ and $\nu_m = 10$ in 3D.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis, we developed an approach for the numerical solution of an incompressible magnetohydrodynamics (MHD) model, with the goal of solving large scale problems. To that end, we generated a large scale code base, utilising both the finite element software package `FEniCS` [40] and linear algebra software `PETSc` [9, 10].

Our mixed finite element discretisation approach is based on using Taylor-Hood elements for the fluid variables and on a mixed Nédélec pair for the magnetic unknowns. The three linearised iteration strategies for the MHD model range from standard Picard iterations to completely decoupled schemes. For these iteration schemes, we followed the preliminary preconditioning results from [35]. The proposed preconditioning ideas are motivated by state of the art preconditioners for the mixed Maxwell and Navier-Stokes subproblems.

We have introduced two preconditioners for the Navier-Stokes equations. When applying the Least-Squares Commutator preconditioner we observed a lack of scalability with respect to the mesh size and the viscosity parameter but for the Pressure Convection-Diffusion (PCD) preconditioner we obtained

scalable results. This lead us to using PCD within the preconditioner for the non-linear iteration schemes. The preconditioning results for Maxwell's equations exhibited iteration counts independent of the mesh size and the magnetic viscosity.

We have presented results for the three iteration schemes the linearised MHD model. After confirming that the numerical results meet the expected a priori convergences rates, parameter tests (using direct solvers for the matrix coefficients of the iteration schemes) were carried out to see the performance of the two decoupling schemes as well as the Picard iteration. From the parameter tests we saw that the robustness of the three schemes greatly depends on the non-dimensional parameters in the problem. The Picard iterations behaved similarly to the Magnetic Decoupling (MD) scheme for relevant parameters based on physical applications. However, the Complete Decoupling (CD) scheme did not converge to the solution for small kinematic viscosities (i.e., $\nu = 0.1$).

Finally, we showed preconditioning results for the MHD linearisations resulting from the three non-linear schemes. One of the principal objectives of this thesis was to perform large scale preconditioning tests. Here 2D scalable experiments were run with 24 million degrees of freedom for the two decoupled iterations. For the Picard iteration, results were only presented for up to 1.5 million degrees of freedom. We were able to go to much greater problem sizes for the decoupled iterations ((MD) and (CD)) as the preconditioning approach is much less involved (no inner-outer iterations).

We also considered the same parameter tests as above but using the preconditioning approaches; the results examined the robustness of the de-

coupled preconditioned techniques. Again, the numerical results showed iterations counts independent of the problem size. From the results the convergence of all three iterations ((P), (MD) and (CD)) was hindered by small values of ν. In fact, only the (MD) scheme converged for a $\nu = 0.01$. We stress the need to find more robust preconditioning techniques for small values of the viscosity; see Section 5.2. Similar conclusions can be drawn from the scalable 3D numerical results presented here.

From the preconditioned and non-preconditioned parameter tests in this thesis, it seems that (MD) performs best for feasible physical parameters (small values of ν, $\kappa \approx 1$ and $\nu_m \approx 10$). The Picard iteration is significantly more costly due to the inner-outer iterations and hence, should only be used in the ranges where (MD) or (CD) do not converge. If memory and the values of the parameters are moderate (e.g., $\nu \geq 1$) then the (CD) scheme is best. This is because the coefficient matrix for (CD) is symmetric and therefore we use the cheaper MINRES Krylov solver.

The experiments carried out in the thesis are quite idealised. We employed uniform regular grids, and tested problems for smooth solutions with relatively harmless parameter values. For many practical applications the problems may involve complex 3D geometries with varying, discontinuous or even non-linear material coefficients. In addition, MHD flows may be convection-dominated. Tackling these difficulties in a computationally efficient manner may require additional computational tools such as automatic mesh generation in complicated 3D domains, adaptive mesh refinement techniques, as well as refined finite element models to handle strong convection and more complex material laws.

As for solution methods, in this thesis we have considered problems of sufficiently small size which enabled us to use direct solves on the preconditioners. In practice, however, large real-world 3D problems require fully scalable preconditioned iterative solution methods, e.g., specialised multigrid techniques

The scalable results presented in this thesis provide a starting point for further in-depth studies into these issues in the context of MHD applications.

5.2 Future work

We finish this thesis with some considerations of possible areas for future work.

1. **Scalable inner solvers:**

   In this thesis, the application of the preconditioners have been carried out using direct solves for the component blocks. However, to construct fully scalable preconditioners we require an iterative approach to solve these systems of equations. For the most part, this can be readily achieved by using standard algebraic or geometric multigrid cycles and/or iterative methods. We note though that the scalable iterative solution for the shifted curl-curl operator in (3.8) is more involved. One option is to employ the multigrid solver developed in [31]. We expect the implementation of scalable inner solves to significantly improve the efficiency of the overall approach, especially for large scale 3D problems and we should obtain a fully scalable iterative solver.

2. **Release code on a public repository:**

   Once the code has been cleaned up and properly commented, the idea is to release it for public use on either GitHub `https://github.com/` or Bitbucket `https://bitbucket.org/`.

3. **Parallelisation of the code:**

   The code to discretise and solve the MHD system (1.11), (1.12) is currently executed sequentially. For 3D problems of large dimension, an efficient parallel implementation is expected to greatly decrease setup and solve time, in particular for the decoupling schemes.

4. **Robustness with respect to kinematic viscosity:**

   The results presented in Chapter 4 indicate the efficient performance with respect to the mesh size and with respect to the non-dimensional parameters (ν, ν_m and κ). However, for small fluid viscosities, the performance of all the proposed preconditioners for our discretisation starts to degrade. The development of preconditioners and convection stabilised discretisations that work well together and are more robust with respect to ν is another area of possible future work. The solution of convection-dominated flow is very challenging and is a subject of active research in many other areas.

5. **High frequencies in Maxwell's equations:**

   The Maxwell formulation considered in this MHD model only considers low frequencies. The consideration of high frequencies introduces hyperbolicity into the system (for the magnetic field $\boldsymbol{b}$). It may still be possible to use the proposed discretisation in space, but the

time discretisation and keeping the consistency of the previously made assumptions (e.g., non-relativistic motion) may introduce additional challenges.

6. **Other non-linear solvers:**

   The non-linear solver currently used is simply the linearised Oseen form (i.e., a linearly convergent Picard iteration) for the MHD system. A possible next step includes considering other non-linear solvers that have faster convergence properties. One such scheme is the Newton iteration, which converges quadratically for a sufficiently accurate initial guess. One challenge here is that the convergence is local, and it is necessary to use various means to compute initial iterates that are close enough to the solution. A possible approach is to start with a few Picard iterations then use the Newton iterations.

7. **Different mixed finite element discretisations:**

   We used Taylor-Hood elements [59] for the fluid variables. This choice of elements is a very common option when considering mixed finite element discretisations of fluid flow equations. However, when using continuous pressure elements additional errors may arise due to poor mass conservation on the discrete level. A possible example of this phenomenon has been studied in [39] in the context of colliding flows. One option to overcome this difficulty is to use exactly divergence-free elements such as those proposed in [11, 25]. The elements there are based on using divergence-conforming elements for the velocities and on discontinuous pressure elements. A discontinuous Galerkin

approach is employed to ensure H^1-continuity of the velocity fields. The preconditioning approaches in this thesis are based [35], where exactly divergence-free elements have been used for the fluid unknowns to approximate MHD problem (1.11). One major challenge on the linear algebra front is the development of a scalable iterative solver for such elements. A large-scale study of exactly divergence-free elements for the fluid unknowns, and more generally the investigation of other mixed finite element approximations with discontinuous pressures for MHD models is another area of possible future work.

Bibliography

[1] M. Alnæs. UFL: a finite element form language. In A. Logg, K. A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*, chapter 17. Springer, 2012.

[2] M. Alnæs, A. Logg, K. A. Mardal, O. Skavhaug, and H. P. Langtangen. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 4(4):231–244, 2009.

[3] M. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, 40(2):9:1–9:37, 2014.

[4] M. S. Alnæs, A. Logg, and K. A. Mardal. UFC: a finite element code generation interface. In A. Logg, K. A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*, chapter 16. Springer, 2012.

[5] P. R. Amestoy, I. S. Duff, and J. Y. L'Excellent. Multifrontal parallel

distributed symmetric and unsymmetric solvers. *Computer Methods in Applied Mechanics and Engineering*, 184(2):501–520, 2000.

[6] P. R. Amestoy, I. S. Duff, J. Y. L'Excellent, and J. Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[7] P. R. Amestoy, A. Guermouche, J. Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Computing*, 32(2):136–156, 2006.

[8] F. Armero and J. C. Simo. Long-term dissipativity of time-stepping algorithms for an abstract evolution equation with applications to the incompressible MHD and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 131(1):41–90, 1996.

[9] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang. PETSc User's Manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[10] S. Balay, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2014.

[11] B. Cockburn, G. Kanschat, and D. Schötzau. A note on discontinuous Galerkin divergence-free solutions of the Navier-Stokes equations. *Journal of Scientific Computing*, 31(1-2):61–73, 2007.

[12] R. Codina and N. Hernández-Silva. Stabilized finite element approximation of the stationary magneto-hydrodynamics equations. *Computational Mechanics*, 38(4-5):344–355, 2006.

[13] M. Costabel and M. Dauge. Singularities of electromagnetic fields in polyhedral domains. *Archive for Rational Mechanics and Analysis*, 151(3):221–276, 2000.

[14] P. A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2001.

[15] T. A. Davis. Algorithm 832: UMFPACK v4.3—an unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):196–199, June 2004.

[16] T. A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Transactions on Mathematical Software*, 30(2):165–195, June 2004.

[17] L. Demkowicz and L. Vardapetyan. Modelling of electromagnetic absorption/scattering problems using *hp*-adaptive finite elements. *Computer Methods in Applied Mechanics and Engineering*, 152(1):103–124, 1998.

[18] 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, 2005.

[19] R. D. Falgout and U. Yang. *hypre*: A library of high performance preconditioners. In *Computational Science ICCS 2002*, volume 2331 of *Lecture Notes in Computer Science*, pages 632–641. Springer Berlin Heidelberg, 2002.

[20] R. Fletcher. Conjugate gradient methods for indefinite systems. In G.Alistair Watson, editor, *Numerical Analysis*, volume 506 of *Lecture Notes in Mathematics*, pages 73–89. Springer Berlin Heidelberg, 1976.

[21] R. W. Freund and N. M. Nachtigal. QMR: a quasi-minimal residual method for non-Hermitian linear systems. *Numerische Mathematik*, 60(3):315–339, 1991.

[22] J. F. Gerbeau. A stabilized finite element method for the incompressible magnetohydrodynamic equations. *Numerische Mathematik*, 87(1):83–111, 2000.

[23] J. F. Gerbeau, C. Le. Bris, and T. Lelièvre. *Mathematical Methods for the Magnetohydrodynamics of Liquid Metals*. Oxford University Press, 2006.

[24] G. H. Golub and C. Greif. On solving block-structured indefinite linear systems. *SIAM Journal on Scientific Computing*, 24(6):2076–2092, 2003.

[25] C. Greif, Li, D. Schötzau, and X. Wei. A mixed finite element method with exactly divergence-free velocities for incompressible magnetohydrodynamics. *Computer Methods in Applied Mechanics and Engineering.*, 199(45-48):2840–2855, 2010.

[26] C. Greif and D. Schötzau. Preconditioners for saddle point linear systems with highly singular (1, 1) blocks. *Electronic Transactions on Numerical Analysis, Special Volume on Saddle Point Problems*, 22:114–121, 2006.

[27] C. Greif and D. Schötzau. Preconditioners for the discretized time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, 14(4):281–297, 2007.

[28] M. D. Gunzburger, A. J. Meir, and J. S. Peterson. On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics. *Mathematics of Computation*, 56(194):523–563, 1991.

[29] P. Hénon, P. Ramet, and J. Roman. PASTIX: a high-performance parallel direct solver for sparse symmetric positive definite systems. *Parallel Computing*, 28(2):301–321, 2002.

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

[31] R. Hiptmair and J. Xu. Nodal auxiliary space preconditioning in $H(\mathrm{curl})$ and $H(\mathrm{div})$ spaces. *SIAM Journal on Numerical Analysis*, 45(6):2483–2509, 2007.

[32] R. C. Kirby. Algorithm 839: FIAT, a new paradigm for computing finite element basis functions. *ACM Transactions on Mathematical Software*, 30(4):502–516, 2004.

[33] R. C. Kirby. FIAT: Numerical construction of finite element basis functions,. In A. Logg, K. A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*, chapter 13. Springer, 2012.

[34] R. C. Kirby and A. Logg. A compiler for variational forms. *ACM Transactions on Mathematical Software*, 32(3), 2006.

[35] D. Li. *Numerical Solution of the Time-Harmonic Maxwell equations and Incompressible Magnetohydrodynamics Problems*. PhD thesis, University of British Columbia, 2010.

[36] D. Li, C. Greif, and D. Schötzau. Parallel numerical solution of the time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, 19(3):525–539, 2012.

[37] X. S. Li. An overview of SuperLU: Algorithms, implementation, and user interface. *ACM Transactions on Mathematical Software*, 31(3):302–325, 2005.

[38] X.S. Li, J. W. Demmel, J. R. Gilbert, L. Grigori, M. Shao, and I. Yamazaki. SuperLU Users' Guide. Technical Report LBNL-44289, Lawrence Berkeley National Laboratory, 1999. `http://crd.lbl.gov/~xiaoye/SuperLU/`. Last update: August 2011.

[39] A. Linke. Collision in a cross-shaped domain—a steady 2D Navier-Stokes example demonstrating the importance of mass conservation

in CFD. *Computer Methods in Applied Mechanics and Engineering*, 198(41-44):3278–3286, 2009.

[40] A. Logg, K. A. Mardal, and G. N. Wells, editors. *Automated solution of differential equations by the finite element method*, volume 84 of *Lecture Notes in Computational Science and Engineering*. Springer, Heidelberg, 2012. The FEniCS book.

[41] A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells. FFC: the FEniCS form compiler. In A. Logg, K. A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*, chapter 11. Springer, 2012.

[42] A. Logg and G. N. Wells. DOLFIN: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2):20:1–20:28, 2010.

[43] A. Logg, G. N. Wells, and J. Hake. DOLFIN: a C++/Python finite element library. In A. Logg, K. A. Mardal, and G. N. Wells, editors, *Automated Solution of Differential Equations by the Finite Element Method, Volume 84 of Lecture Notes in Computational Science and Engineering*, chapter 10. Springer, 2012.

[44] P. Monk. *Finite Element Methods for Maxwell's Equations*. Oxford University Press, 2003.

[45] U. Müller and L. Bühler. *Magnetofluiddynamics in Channels and Containers*. Springer, 2001.

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

[47] J. C. Nédélec. Mixed finite elements in $\mathbb{R}^3$. *Numerische Mathematik*, 35(3):315–341, 1980.

[48] K. B. Ølgaard and G. N. Wells. Optimisations for quadrature representations of finite element tensors through automated code generation. *ACM Transactions on Mathematical Software*, 37:23, 2010.

[49] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[50] E. G. Phillips, H. C. Elman, E. C. Cyr, J. N. Shadid, and R. P. Pawlowski. A block preconditioner for an exact penalty formulation for stationary MHD. Technical Report CS-TR-5032, University of Maryland, Computer Science, 2014.

[51] P. H. Roberts. *An Introduction to Magnetohydrodynamics*. Longmans, London, 1967.

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

[53] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, PA, second edition, 2003.

[54] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[55] D. Schötzau. Mixed finite element methods for stationary incompressible magneto–hydrodynamics. *Numerische Mathematik*, 96(4):771–800, 2004.

[56] D. J. 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.

[57] T. A. Davis and I. S. Duff. An unsymmetric-pattern multifrontal method for sparse LU factorization. *SIAM Journal on Matrix Analysis and Applications*, 18(1):140–158, 1997.

[58] T. A. Davis and I. S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Transactions on Mathematical Software*, 25(1):1–20, 1999.

[59] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers and Fluids*, 1(1):73–100, 1973.

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

[61] A. J. Wathen and D. J. Silvester. Fast iterative solution of stabilised

Stokes systems Part I: Using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis*, 30(3):630–649, 1993.

Appendices

Appendix A

Curl operators and cross products

A.1 2D curl

Given 2D vector fields $\boldsymbol{b}(x, y) = (b_1, b_2)$, $\boldsymbol{u}(x, y) = (u_1, u_2)$ and the scalar function $r(x, y)$, the curl and cross products are

$$\begin{aligned} \nabla \times \boldsymbol{b} &= \frac{\partial b_2}{\partial x} - \frac{\partial b_1}{\partial y}, \\ \nabla \times r &= \Big(\frac{\partial r}{\partial y}, -\frac{\partial r}{\partial x}\Big), \\ \boldsymbol{u} \times \boldsymbol{b} &= u_1 b_2 - u_2 b_1. \end{aligned}$$

Note that taking the curl of a 2D vector field results in a scalar function which is the component in the normal direction to the 2D field (z-component).

A.2 3D curl

Given 3D vector fields $\boldsymbol{b}(x, y, z) = (b_1, b_2, b_3)$ and $\boldsymbol{u}(x, y, z) = (u_1, u_2, u_3)$, the curl and cross products are

$$\nabla \times \boldsymbol{b} = \begin{pmatrix} \frac{\partial b_3}{\partial y} - \frac{\partial b_2}{\partial z} \\ \frac{\partial b_1}{\partial z} - \frac{\partial b_3}{\partial x} \\ \frac{\partial b_2}{\partial x} - \frac{\partial b_1}{\partial y} \end{pmatrix},$$

$$\boldsymbol{u} \times \boldsymbol{b} = \begin{pmatrix} u_2 b_3 - u_3 b_2 \\ u_3 b_1 - u_1 b_3 \\ u_1 b_2 - u_2 b_1 \end{pmatrix}.$$

Appendix B

Krylov subspace methods

PDE discretisation usually leads to sparse linear systems. For problems of small dimension, a popular choice to solve such a system are sparse direct methods. However, for systems of large dimension, it is often not possible to use direct methods due to computational memory restrictions, time constraints, etc. Therefore an iterative solution method, based on matrix-vector products, is required. One state of the art approach is to consider preconditioned Krylov subspace methods [53].

To review Krylov subspace methods, consider iteratively solving the linear system

$$Ax = b,$$

where A is a real non-singular $n \times n$ matrix with vectors x and b of the same dimension. Suppose our initial guess is given by x_0. Then the initial residual is given by $r_0 = b - Ax_0$. Usually in practice this is taken to be zero. Using the initial residual r_0, the m-dimensional Krylov subspace is defined as:

$$\mathscr{K}_m(A; r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \cdots, A^{m-1} r_0\}.$$

A Krylov subspace method is based on finding a solution, x_m, in the Krylov

subspace:

$$A^{-1}b \approx x_m = x_0 + y_m,$$

where $y_m \in \mathscr{K}_m(A; r_0)$ satisfies some optimality criteria. One could characterise the three main components to Krylov subspace methods by:

1. Computing an orthogonal or bi-orthogonal basis to the Krylov subspace;

2. Defining an optimality criterion within the Krylov subspace;

3. Preconditioning approaches.

Basis computations

Orthogonal basis:

Consider a general real unsymmetric matrix A. Then the process in which it is possible to construct an orthogonal basis to the Krylov subspace $\mathscr{K}_m(A; r_0)$ is known as the Arnoldi process. The Arnoldi process is essentially the modified Gram-Schmidt process applied to the Krylov subspace. This process is done iteratively, so at each iteration a new orthogonal vector is added to the basis. The Arnoldi process is written as:

$$AQ_m = Q_{m+1}H_{m+1,m}, \tag{B.1}$$

where Q_{m+1} is the matrix whose $m+1$ columns form an orthogonal basis for the subspace $\mathscr{K}_m(A; r_0)$. Q_m has the same leading m columns as in Q_{m+1} and $H_{m+1,m} \in \mathbb{R}^{(m+1)\times m}$ is an upper Hessenberg matrix. From (B.1) one

can deduce the following form:

$$Q_m^T A Q_m = H_{m,m}, \tag{B.2}$$

where $H_{m,m}$ is the matrix containing the first m rows of $H_{m+1,m}$. If A is symmetric then from (B.2) $H_{m,m}$ must also be symmetric. This leads to the Lanzcos method where $H_{m,m}$ is tridiagonal and hence, we denote it as $T_{m,m}$ in this case. Then the Lanzcos process is as follows

$$A Q_m = Q_{m+1} T_{m+1,m}, \tag{B.3}$$

where $T_{m+1,m}$ has one more extra row than $T_{m,m}$. See [53] for more details about both the Arnoldi and Lanczos processes.

Bi-Orthogonal basis:

For non-symmetric matrices we cannot expect short term recurrences (resulting from the orthogonal basis construction) as well as retain optimality in the same way that symmetric solvers do. For example, GMRES minimises the residual but entails long recurrences. The class of biconjugate methods preserve the short term recurrences property and hence, have less memory requirements and are very popular as a result.

To introduce the bi-orthogonal basis construction, we define a second Krylov subspace:

$$\mathscr{K}_m(A^T; r_0) = \text{span}\{r_0, A^T r_0, (A^T)^2 r_0, \cdots, (A^T)^{m-1} r_0\}.$$

Let V_m and W_m be biorthogonal basis matrices with respect to the subspaces $\mathscr{K}_m(A; r_0)$ and $\mathscr{K}_m(A^T; r_0)$, that is $W_m^T V_m = I$. Then the idea is to perform two "Lanzcos-type" processes for A and A^T. This leads to the procedure:

$$AV_m = V_{m+1}\hat{T}_{m+1,m}, \tag{B.4a}$$

$$A^T W_m = W_{m+1} S_{m+1,m}, \tag{B.4b}$$

where $\hat{T}_{m+1,m}$ and $S_{m+1,m}$ are tridiagonal matrices of the same structure as $T_{m+1,m}$ as in (B.3). Pre-multiplying (B.4a) by W_m^T gives

$$W_m^T A V_m = W_m^T V_{m+1} \hat{T}_{m,m} = T_{m,m},$$

as $W_m^T V_m = I$ by construction. This is known as (non-symmetric) Lanzcos biorthogonalisation and leads to such methods as BiCG [20], BiCGstab [60] and QMR [21].

Optimality criteria

There are two popular choices for how to define the optimality criterion for a Krylov solver. These are minimising the L^2-norm within the Krylov subspace (minimum residual approach) and ensuring that the residual is orthogonal to the Krylov subspace (Galerkin approach).

For the three Krylov subspace methods that will be used within this thesis, namely GMRES [54] for unsymmetric systems, MINRES [49] for symmetric and Conjugate Gradients [30] (CG) for symmetric positive definite systems, the optimality criteria are as follows:

- GMRES: $\min \|b - Ax_m\|_2$, i.e., minimum residual approach
- MINRES: $\min \|b - Ax_m\|_2$, i.e., minimum residual approach
- CG: $\min \|x_m - x\|_A$, the energy norm of the error which is equivalent to orthogonalising with respect to the search space, i.e., Galerkin approach

For more details about these methods and other Krylov subspace methods we refer the reader to [53].

Preconditioning

Preconditioning is an integral part of using any Krylov subspace method. The convergence of CG as well as other methods primarily depend on the spread of the eigenvalues. Preconditioning the linear system $Ax = b$ amounts to solving the system $M^{-1}Ax = M^{-1}b$, where M is a *preconditioner*. There are certain properties that are required to create a good preconditioner:

1. the preconditioner M must approximate A so that the eigenvalues of $M^{-1}A$ are clustered;
2. systems associated with M should be much easier to solve than systems with A.

Loosely speaking, one could classify preconditioners into two types. The first type is an operator-based preconditioner. By this we mean that given a certain problem, usually based on a PDE discretisation, it may be possible to design certain operators that lead to an optimal or nearly optimal preconditioner. These operators are derived from an understanding of the physical

problem and its discretisation. Often these operators are based on spectral equivalence: the eigenvalues of $M^{-1}A$ are independent of discretisation parameters. Weak dependency is sometimes sufficient.

The second type is more of a "black-box" method, where the preconditioner is purely based on the matrix coefficients. The preconditioning strategies employed in this thesis are mainly based on the first type.