title

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 Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

June 2014

Abstract

Preface

Table of Contents

Abstract ii

Preface iii

Table of Contents iv

List of Tables vii

List of Figures ix

1 Introduction 1

1.1 Magnetohydrodynamics 1

1.1.1 Navier-Stokes 1

1.1.2 Maxwell 1

1.2 A model problem 1

1.3 Finite element methods 2

1.4 Iterative Methods 2

1.5 Objectives and contributions 2

2 Finite element discretisation 3

2.1 Variational formulation 3

2.2 Mixed finite element discretisation 6
2.2.1 Matrix representation 8
2.3 Picard iteration (P) 10
2.4 Decoupled iterations 12
2.4.1 Magnetic decoupling (MD) 13
2.4.2 Complete decoupling 14

3 Preconditioning 16
3.1 Navier-Stokes equations 16
3.1.1 Pressure Convection-Diffusion (PCD) 17
3.1.2 Least Squares Commutator 19
3.2 Maxwell's equations 21
3.2.1 An ideal preconditioner 22
3.2.2 A practical preconditioner 22
3.3 A preconditioner for the MHD problem 23
3.4 Preconditioners for MD and CD 24
3.4.1 Magnetic decoupling 25
3.4.2 Complete decoupling 25

4 Numerical Results 27
4.1 Software 27
4.2 Problem setup 28
4.3 Navier-Stokes equations 31
4.3.1 2D smooth solution 31
4.3.2 3D smooth solution 32
4.3.3 Least Squares Commutator 33

4.3.4 Pressure Convection-Diffusion 34
4.4 Maxwell's equations 36
4.4.1 2D smooth solution 36
4.4.2 3D smooth solution 37
4.4.3 Preconditioning 38
4.5 MHD problem 40
4.5.1 2D smooth solution 40
4.5.2 3D smooth solution 42
4.5.3 Parameter decoupling tests 44
4.6 Preconditioned MHD problem 46
4.6.1 MHD problem 46
4.6.2 Magnetic decoupling 47
4.6.3 Complete decoupling 47

5 Conclusion 49

Bibliography 50

List of Tables

4.1 Convergence table for 2D Navier-Stokes smooth solution - Picard tolerance 1e-10 31

4.2 Convergence table for 3D Navier-Stokes smooth solution - Picard tolerance 1e-5 33

4.3 Iteration table for LSC preconditioner for 2D example - Picard tolerance 1e-10 34

4.4 Iteration table for PCD preconditioner for 2D example - Picard tolerance 1e-10 35

4.5 Iteration table for PCD preconditioner for 3D example - Picard tolerance 1e-10 36

4.6 Convergence table for 2D Maxwell smooth solution - Magnetic field 36

4.7 Convergence table for 2D Maxwell smooth solution - multiplier field 37

4.8 Convergence table for 3D Maxwell smooth solution - magnetic field 38

4.9 Convergence table for 3D Maxwell smooth solution - multiplier field 38

4.10 Iteration table for Maxwell preconditioner for 2D example - direct application of preconditioner 39
4.11 Iteration table for Maxwell preconditioner for 3D example - direct application of preconditioner 40
4.12 Convergence table for 2D MHD smooth solution - Picard tolerance 1e-8 41
4.13 Convergence table for 2D MHD smooth solution - Magnetic field 42
4.14 Convergence table for 2D MDH smooth solution - multiplier field 42
4.15 Convergence table for 3D MHD smooth solution - Picard tolerance 1e-8 43
4.16 Convergence table for 3D MHD smooth solution - Magnetic field 43
4.17 Convergence table for 3D MDH smooth solution - multiplier field 44
4.18 Number of non-linear iterations for various values of ν. Let the Picard tolerance to be $1e-5$, $\kappa = 1$ and $\nu_m = 10$. 45
4.19 Number of non-linear iterations for various values of ν_m. LLet the Picard tolerance to be $1e-5$, $\nu = 1$ and $\nu_m = 1e2$. 46
4.20 $tol = 1e-5 kappa = 1.0/2 Mu_m = 1e2 MU = 0.1$ 47
4.21 $tol = 1e-5 kappa = 1.0/2 Mu_m = 1e2 MU = 10$ 48

List of Figures

4.1 Level 3 grid on unit square domain 29

Chapter 1

Introduction

1.1 Magnetohydrodynamics

1.1.1 Navier-Stokes

1.1.2 Maxwell

1.2 A model problem

stationary incompressible and resistive magnetohydrodynamics (MHD) system:

$$-\nu\,\Delta \boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \kappa\,(\nabla\times\boldsymbol{b})\times\boldsymbol{b} = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1.1a}$$

$$\nabla\cdot\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{1.1b}$$

$$\kappa\nu_m\,\nabla\times(\nabla\times\boldsymbol{b}) + \nabla r - \kappa\,\nabla\times(\boldsymbol{u}\times\boldsymbol{b}) = \boldsymbol{g} \qquad \text{in } \Omega, \tag{1.1c}$$

$$\nabla\cdot\boldsymbol{b} = 0 \qquad \text{in } \Omega. \tag{1.1d}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \qquad \text{on } \partial\Omega, \tag{1.2a}$$

$$\boldsymbol{n} \times \boldsymbol{b} = \boldsymbol{n} \times \boldsymbol{b}_D \qquad \text{on } \partial\Omega, \tag{1.2b}$$

$$r = 0 \qquad \text{on } \partial\Omega, \tag{1.2c}$$

1.3 Finite element methods

$\mathcal{P}_k$: the space of polynomials of total degree at most k $\boldsymbol{R}_k$: the space of homogeneous vector polynomials of total degree k that are orthogonal to the position vector $\boldsymbol{x}$

1.4 Iterative Methods

1.5 Objectives and contributions

Chapter 2

Finite element discretisation

In this chapter we introduce a mixed finite element discretisation for a steady-state incompressible MHD problem that model electrically conductive fluids. Following the setting in [36], using 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 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 (1.2) can be incorporated in a straightforward fashion.

2.1 Variational formulation

To express (1.1), (1.2) in weak form we follow [36] 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

$$
\begin{aligned}
\boldsymbol{V} &= H_0^1(\Omega)^d = \left\{ \boldsymbol{u} \in H^1(\Omega)^d \,:\, \boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega \right\}, \\
Q &= L_0^2(\Omega) = \{\, p \in L^2(\Omega) \,:\, (p\,,1)_\Omega = 0 \,\}, \\
\boldsymbol{C} &= H_0(\mathrm{curl};\Omega) = \left\{ \boldsymbol{b} \in L^2(\Omega)^d \,:\, \nabla \times \boldsymbol{b} \in L^2(\Omega)^{2d-3},\ \boldsymbol{n} \times \boldsymbol{b} = \boldsymbol{0} \text{ on } \partial\Omega \right\}, \\
S &= H_0^1(\Omega) = \{\, r \in H^1(\Omega) \,:\, r = 0 \text{ on } \partial\Omega \,\},
\end{aligned}
$$

We write $\|\cdot\|_{L^2(\Omega)}$, $\|\cdot\|_{H^1(\Omega)}$ and $\|\cdot\|_{H(\mathrm{curl};\Omega)}$ for the associated natural norms. More precisely, for a vector fields $\boldsymbol{u}, \boldsymbol{b}$ and a scalar functions r the norms are defined as follows:

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

where $\|\nabla \boldsymbol{u}\|^2_{L^2(\Omega)}$ is the L^2-norm of the gradient tensor $\nabla \boldsymbol{u}$. The weak formulation of the incompressible MHD system (1.1) and the boundary conditions

(1.2) consists in finding $(\boldsymbol{u}, p, \boldsymbol{b}, r) \in \boldsymbol{V} \times Q \times \boldsymbol{C} \times S$ such that

$$\begin{aligned} A(\boldsymbol{u}, \boldsymbol{v}) + O(\boldsymbol{u}; \boldsymbol{u}, \boldsymbol{v}) + C(\boldsymbol{b}; \boldsymbol{v}, \boldsymbol{b}) + B(\boldsymbol{v}, p) &= (\boldsymbol{f}, \boldsymbol{v})_\Omega, && \text{(2.2a)} \\ B(\boldsymbol{u}, q) &= 0, && \text{(2.2b)} \\ M(\boldsymbol{b}, \boldsymbol{c}) - C(\boldsymbol{b}; \boldsymbol{u}, \boldsymbol{c}) + D(\boldsymbol{c}, r) &= (\boldsymbol{g}, \boldsymbol{c})_\Omega, && \text{(2.2c)} \\ D(\boldsymbol{b}, s) &= 0, && \text{(2.2d)} \end{aligned}$$

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

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

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

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

In [36] it has been shown that this formulation of a the problem is discrete energy-stable and has a unique solution for small data.

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 spaces given by:

$$\begin{aligned}
\boldsymbol{V}_h &= \{\, \boldsymbol{u} \in H^1(\Omega) \,:\, \boldsymbol{u}|_K \in \mathcal{P}_k(K)^d,\, K \in \mathcal{T}_h \,\}, \\
Q_h &= \{\, p \in L^2(\Omega) \cap H^1(\Omega) \,:\, p|_K \in \mathcal{P}_{k-1}(K),\, K \in \mathcal{T}_h \,\}, \\
\boldsymbol{C}_h &= \{\, \boldsymbol{b} \in H_0(\mathrm{curl};\Omega) \,:\, \boldsymbol{b}|_K \in \mathcal{P}_{k-1}(K)^d \oplus \boldsymbol{R}_k(K),\, K \in \mathcal{T}_h \,\}, \\
S_h &= \{\, r \in H_0^1(\Omega) \,:\, r|_K \in \mathcal{P}_k(K),\, K \in \mathcal{T}_h \,\},
\end{aligned} \tag{2.3}$$

for $k \geq 2$. Here we note that we are using $\mathcal{P}_k/\mathcal{P}_{k-1}$ Taylor-Hood elements for the fluid unknowns $(\boldsymbol{u}, p)$ [39]. For the magnetic variables $(\boldsymbol{b}, r)$ we use the curl-conforming Nédélec elements of the first kind [32]. 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 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

$$A(\boldsymbol{u}_h, \boldsymbol{v}) + \tilde{O}(\boldsymbol{u}_h; \boldsymbol{u}_h, \boldsymbol{v}) + C(\boldsymbol{b}_h; \boldsymbol{v}, \boldsymbol{b}_h) + B(\boldsymbol{v}, p_h) = (\boldsymbol{f}, \boldsymbol{v}), \tag{2.4a}$$

$$B(\boldsymbol{u}_h, q) = 0, \tag{2.4b}$$

$$M(\boldsymbol{b}_h, \boldsymbol{c}) - C(\boldsymbol{b}_h; \boldsymbol{u}_h, \boldsymbol{c}) + D(\boldsymbol{c}, r_h) = (\boldsymbol{g}, \boldsymbol{c}), \tag{2.4c}$$

$$D(\boldsymbol{b}_h, s) = 0, \tag{2.4d}$$

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 to modify the form $O(\boldsymbol{w};\boldsymbol{u},\boldsymbol{v})$ in a standard fashion to ensure the energy-stability property

$$\tilde{O}(\boldsymbol{w};\boldsymbol{u},\boldsymbol{u}) = 0, \quad \forall \boldsymbol{w}, \boldsymbol{u} \in \boldsymbol{V}_h. \tag{2.5}$$

To ensure this property 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, the property (2.5) 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 [36] it has been shown that this formulation of a MHD is discrete 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 (2.5). Namely, for sufficiently smooth solutions, we

have that

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

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

2.2.1 Matrix representation

This variational formulation (2.4) now can be converted into a matrix representation. To do this, we introduce the basis function for the finite elements spaces in (2.3):

$$\boldsymbol{V}_h=\mathrm{span}\langle\boldsymbol{\psi}_j\rangle_{j=1}^{n_u},\qquad Q_h=\mathrm{span}\langle\alpha_i\rangle_{i=1}^{m_u},\tag{2.6}$$

$$\boldsymbol{C}_h=\mathrm{span}\langle\boldsymbol{\phi}_j\rangle_{j=1}^{n_b},\qquad S_h=\mathrm{span}\langle\beta_i\rangle_{i=1}^{m_b}.\tag{2.7}$$

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 $(\boldsymbol{u}_h,p_h,\boldsymbol{b}_h,r_h)$. As usual, this is done by writing the bilinear forms in (2.4) 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 $\boldsymbol{w} \in \boldsymbol{V}_h$ and $\boldsymbol{d}_h \in \boldsymbol{C}_h$ in the first argument and their associated coefficient vectors w and d as

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

Thus, the numerical solution to (1.1) consists in solving the non-linear system

$$
\begin{pmatrix}
A + O(u) & B^T & C^T(b) & 0 \\
B & 0 & 0 & 0 \\
-C(b) & 0 & M & D^T \\
0 & 0 & D & 0
\end{pmatrix}
\begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix}
=
\begin{pmatrix} f \\ 0 \\ g \\ 0 \end{pmatrix}. \tag{2.8}
$$

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. We shall omit the dependence of O and C on b and u, respectively, and simply write O and C.

2.3 Picard iteration (P)

The discrete system (2.8) 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 [14]. This involves linearising around the current velocity and solving for updates.

We adapt this approach for the full MHD system as well. 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 $(\boldsymbol{v}, q, \boldsymbol{c}, s) \in \boldsymbol{V}_h \times Q_h \times \boldsymbol{C}_h \times S_h$. 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:

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

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

In [36] it is shown that for small data the Picard iteration (P) will converge to the exact solution given 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

$$
\begin{pmatrix}
A + O & B^T & C^T & 0 \\
B & 0 & 0 & 0 \\
-C & 0 & M & D^T \\
0 & 0 & D & 0
\end{pmatrix}
\begin{pmatrix}
\delta u \\ \delta p \\ \delta b \\ \delta r
\end{pmatrix}
=
\begin{pmatrix}
r_u \\ r_p \\ r_b \\ r_r
\end{pmatrix},
\tag{2.9}
$$

with

$$r_u = f - Au - O(u)u - C(b)^T b - B^T p,$$

$$r_p = -Bu,$$

$$r_b = g - Mu + C(b)b - D^T r,$$

$$r_r = -Db.$$

Here, the matrix A is symmetric positive-definite (SPD), O is non-symmetric and $-C, C^T$ appear in a skew symmetric fashion. We also note that M is symmetric positive-semidefinite (SPSD) with nullity m_b corresponding to the discrete gradients.

2.4 Decoupled iterations

The full MHD system (1.1), (1.2) is a coupled system consisting of the incompressible Navier-Stokes and Maxwell's equations, coupled through the non-linear skew symmetric coupling term C. In addition, the convection term O 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, is small then we may completely decouple the system into a 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.

2.4.1 Magnetic decoupling (MD)

Consider the first situation where there is weak coupling within the system, that is when C is small. Then it may be possible to drop these terms to completely decouple the system into the two subproblems, the Navier-Stokes and Maxwell's equations. We will call this Magnetic decoupling. Then (2.9) amounts

$$\begin{pmatrix} A+O(u) & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \\ \delta b \\ \delta r \end{pmatrix} = \begin{pmatrix} r_u \\ r_p \\ r_b \\ r_r \end{pmatrix}, \tag{2.10}$$

with

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

From (2.10) we can see that the system is now completely decoupled. This enable us to apply solve each individual subproblem separately and possibly in parallel.

2.4.2 Complete decoupling

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

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

with

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

This is the simplest technique as it removes all non-linear terms and hence leaves the linear Stokes problem in the upper $(1,1)$ block matrix.

In this chapter we have introduced a mixed finite element approximation to the full MHD system given in (1.1) and (1.2). We followed the mixed approach outlined in [36] and expressed the MHD system in the matrix form (2.9). Using the Picard iteration (2.9) we introduced two possible decoupling schemes which maybe simpler to solve for depending on the parameters (κ, ν and ν_m). The next chapter will discuss possible preconditioning approaches

to these systems.

Chapter 3

Preconditioning

The linear system (2.9) is typically sparse and of large dimension, hence to efficiently solve for it we use a preconditioned iterative approach as proposed in [25]. 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.

3.1 Navier-Stokes equations

To start with, consider the steady state incompressible Navier-Stokes equations in isolation. Let

$$\mathcal{K}_{\mathrm{NS}} = \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix}, \tag{3.1}$$

be the discretised and linearised Navier-Stokes subproblem where $F = A + O$. Due to the convection term, O, this system is non-symmetric and we will use GMRES to solve this subproblem [35]. An excellent choice for a preconditioner for a saddle point system like this is to use a block

diagonal or block triangular based preconditioner of the form

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

where the Schur complement S is given by $S = BF^{-1}B^T$. It has been proved in [31] that for a suitable Krylov subspace method then the iterative scheme will converge in exactly 2 iterations when using the block triangular preconditioner or 3 iterations using a block diagonal where the B^T is dropped from the $(1,2)$ block in $\mathcal{M}_{\mathrm{NS}}$.

In practice it is often too expensive to form and solve for the Schur complement, hence, a good approximation to it 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. Both can be found in [14] and we will just outline the procedure how these can be applied on the discrete level.

3.1.1 Pressure Convection-Diffusion (PCD)

Before considering the PCD approach to approximate the Schur complement, we define the velocity mass matrix as $Q = (Q_{i,j})_{i,j=1}^{n_u} \in \mathbb{R}^{n_u \times n_u}$, where in terms of the basis $\{\boldsymbol{\psi}_i\}$

$$Q_{i,j} = \int_\Omega \boldsymbol{\psi}_j \cdot \boldsymbol{\psi}_i \, d\boldsymbol{x}, \quad 1 \leq i,j \leq n_u. \tag{3.3}$$

In [14, Chap. 8] the discrete commutator of the convection-diffusion operator associated with the gradient operation is introduced and given by

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

In this equation, $W = (W_{i,j})_{i,j=1}^{m_u} \in \mathbb{R}^{m_u \times m_u}$, $F_p = ((F_p)_{i,j})_{i,j=1}^{m_u} \in \mathbb{R}^{m_u \times m_u}$ and introduce $A_p = ((A_p)_{i,j})_{i,j=1}^{m_u} \in \mathbb{R}^{m_u \times m_u}$ (which will be used later) are mass matrix, convection diffusion operator and Laplacian matrix defined on the pressure space as:

$$\begin{aligned} W_{i,j} &= \int_\Omega \alpha_j \alpha_i \, dx, \quad 1 \le i,j \le m_u, \\ (F_p)_{i,j} &= \nu \int_\Omega \nabla\alpha_j \cdot \nabla\alpha_i + (\boldsymbol{w} \cdot \nabla\alpha_j)\alpha_i \, dx, \quad 1 \le i,j \le m_u, \\ (A_p)_{i,j} &= \int_\Omega \nabla\alpha_j \cdot \nabla\alpha_i \, dx, \quad 1 \le i,j \le m_u. \end{aligned} \tag{3.5}$$

These matricies are well-defined since our pressure spaces are continuous. Assuming that the commutator is small then pre and post multiplying (3.4) by $BF^{-1}Q$ and $F_p^{-1}W$, respectively, lets us separate the Schur complement to give

$$BF^{-1}B^T \approx BQ^{-1}B^T F_p^{-1} W. \tag{3.6}$$

In general, $BF^{-1}B$ is both costly to form and usually dense, so it is impractical to use. Our discretisation is inf-sup stable which means that there is spectral equivalency between $BQ^{-1}B^T$ and the pressure Laplacian, A_p, see [14, Section 5.5.1]. Hence, the Schur complement can be approximated by:

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

Applying the PCD preconditioner to the full Navier-Stokes system involves solving the system

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

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

1. Solve for y: $y = -W^{-1} F_p A_p^{-1} b$
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 (W^{-1}) and one convection-diffusion solve (F^{-1}) at each Krylov iteration. These solves will be done using direct solver unless specifically stated otherwise.

3.1.2 Least Squares Commutator

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

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

Suppose that the Q-norm is defined by $\|v\|_Q = (Qv, v)^{1/2}$. Then this time we minimise ϵ_h in the Q-norm to try to find an expression for F_p. The

minimisation is given by

$$\min \|(Q^{-1}F)(Q^{-1}B^T) - (Q^{-1}B^T)(W^{-1}F_p)\|_Q.$$

Solving this optimisation problem, as shown in [14], is equivalent to solving the following normal equations

$$W^{-1}BQ^{-1}B^TW^{-1}F_p = W^{-1}BQ^{-1}FQ^{-1}B^T.$$

This yields the following expression for F_p:

$$F_p = W(BQ^{-1}B^T)^{-1}(BQ^{-1}FQ^{-1}B^T).$$

By substitution this into expression (3.6) we obtain the LSC approximation to the Schur complement:

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

Therefore, applying the LSC preconditioner to the full Navier-Stokes system $\mathcal{K}_{\text{NS}}$ in (3.1) 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 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. In practice, we take the diagonal or lumped diagonal of Q to form $BQ^{-1}B^T$. These solves, as in with the PCD preconditioner, will be done directly.

3.2 Maxwell's equations

Next, consider the Maxwell subproblem

$$\mathcal{K}_{\mathrm{MX}} = \begin{pmatrix} M & D^T \\ D & 0 \end{pmatrix}. \tag{3.7}$$

As for the Navier-Stokes subproblem in Section 3.1, we apply a block preconditioning strategy for $\mathcal{K}_{\mathrm{MX}}$ in (3.7).

Recall that the $(1,1)$ block of $\mathcal{K}_{\mathrm{MX}}$ is the curl-curl operator, and hence the matrix M is singular with nullity m_b which corresponds to the discrete gradients. Therefore the usual Schur complement does not exist as it involves inverting M. To overcome this difficulty, we employ the approach in [16, 17] based on augmentation. More precisely, we replace M by $M + D^T\mathcal{W}^{-1}D$ where $\mathcal{W} \in \mathbb{R}^{m_b \times m_b}$ is a symmetric positive definite matrix, see [16, 17] for more details. The addition of the matrix $(D^T\mathcal{W}^{-1}D)$ removes the singularity of the $(1,1)$ block of $\mathcal{K}_{\mathrm{MX}}$ without changing the solution (since $Db = 0$). For the Maxwell subproblem the appropriate choice of $\mathcal{W}$ is the scalar Laplacian

on S_h defined as $L = (L_{i,j})_{i,j=1}^{m_b} \in \mathbb{R}^{m_b \times m_b}$ with

$$L_{i,j} = \int_\Omega \nabla\beta_j \cdot \nabla\beta_i \, d\boldsymbol{x}, \tag{3.8}$$

see [18]. Therefore we will consider preconditioning the following augmented system:

$$\bar{\mathcal{K}}_{\mathrm{MX}} = \begin{pmatrix} M + D^T L^{-1} D & D^T \\ D & 0 \end{pmatrix}. \tag{3.9}$$

3.2.1 An ideal preconditioner

It has been shown in [18] that an ideal preconditioner for $\bar{\mathcal{K}}_{\mathrm{MX}}$ in (3.9) is the block diagonal matrix

$$\mathcal{M}_{\mathrm{iMX}} = \begin{pmatrix} M + D^T L^{-1} D & 0 \\ 0 & L \end{pmatrix}. \tag{3.10}$$

Applying (3.10) as the preconditioner yields exactly two eigenvalues, 1 and -1. Therefore using this matrix as a preconditioner means that MINRES will converge in two iterations [34]. However, forming the matrix $M + D^T L^{-1} D$ is costly, hence, $\mathcal{M}_{\mathrm{iMX}}$ 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, $\mathcal{M}_{\mathrm{iMX}}$, suitable in practise. It has been shown in [18] that $M + D^T L^{-1} D$ is spectrally equivalent to $M + X$ where $X = (X_{i,j})_{i,j=1}^{n_b} \in$

$\mathbb{R}^{n_b \times n_b}$ is the mass matrix on the magnetic space and is defined as

$$X_{i,j} = \int_\Omega \boldsymbol{\psi}_j \cdot \boldsymbol{\psi}_i \, d\boldsymbol{x}. \tag{3.11}$$

Using this approximation leads to the practical preconditioner

$$\mathcal{M}_{\mathrm{MX}} = \begin{pmatrix} N & 0 \\ 0 & L \end{pmatrix}, \tag{3.12}$$

where $N = M + X$.

3.3 A preconditioner for the MHD problem

Sections 3.1 and 3.2 looked briefly at the preconditioning strategies for the Navier-Stokes and Maxwell's equations. Using these techniques we will look at possible scalable preconditioners for the full MHD problem,

$$\mathcal{K}_{\mathrm{MH}} = \begin{pmatrix} A + O & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix}. \tag{3.13}$$

Using the Navier-Stokes and Maxwell subproblem preconditioners (3.2) and (3.12) respectively, then we propose the following preconditioner for $\mathcal{K}_{\text{MH}}$

$$\mathcal{M}_{\text{MH}} = \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -S & 0 & 0 \\ -C & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{3.14}$$

Due to the coupling terms, C, the application of this precondioner is hard. To overcome this, we propose to invert $\mathcal{M}_{\text{MH}}$ by means of an inner preconditioner Krylov solver. The inner preconditioner is given by

$$\mathcal{M}_{\text{innerMH}} = \begin{pmatrix} F & B^T & 0 & 0 \\ 0 & -S & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{3.15}$$

3.4 Preconditioners for MD and CD

In Section 2.4 we introduced two decoupling schemes, namely 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 imply for these decoupling schemes.

3.4.1 Magnetic decoupling

From section 2.4.1 the the matrix to be preconditioned is as follows:

$$\mathcal{K}_{\mathrm{MD}} = \begin{pmatrix} F & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix}. \tag{3.16}$$

Recall that removing the coupling terms completely decouples the system. This therefore enables us to use the optimal preconditioners for each of the subproblems separately and in parallel. Using the subproblem preconditioners (3.12) and (3.2) then the optimal preconditioner for $\mathcal{K}_{\mathrm{MD}}$ is

$$\mathcal{M}_{\mathrm{MD}} = \begin{pmatrix} F & B^T & 0 & 0 \\ 0 & -S & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{3.17}$$

3.4.2 Complete decoupling

To form an appropriate preconditioner for the CD iteration in (2.11) we first need to 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. The Schur complement associated with the Stokes system is

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

recall that the matrix A is defined with the viscosity ν in section 2.1. It was shown in [37, 38] that the scaled pressure mass matrix, $\frac{1}{\nu}W$ defined in (3.5), is spectrally equivalent to the Schur complement (which is also a consequence of the inf-sup stability condition). Therefore the scalable Stokes preconditioner is

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

Using (3.18) together with the Maxwell subproblem preconditioner (3.12) gives the preconditioner

$$\mathcal{M}_{\mathrm{CD}} = \begin{pmatrix} A & 0 & 0 & 0 \\ 0 & \frac{1}{\nu}W & 0 & 0 \\ 0 & 0 & N & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{3.18}$$

The biggest advantage of this decoupling approach is that the matrix system is now symmetric. This means that the appropriate choice for the Krylov subspace method is MINRES for each subproblem.

Chapter 4

Numerical Results

In this chapter we present a series of convergence and preconditioning experiments. The principle 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. All numerical experiments have been carried out in the Python programming language.

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 (1.1) was FEniCS [40]. The core libraries used within FEniCS are DOLFIN [29, 30] the problem-solving interface, FFC [24, 28, 33] the compiler for finite element variational forms, FIAT [22, 23] the finite element tabulator, Instant the just-in-time compiler, UFC [2, 3] the code generator and UFL [1, 4] the form language.

Along with FEniCS we have used a number of purely linear algebra libraries. The main one that has been used is PETSc4PY which is the python

interface for PETSc [8, 9]. PETSc has been used mainly for the iterative solvers as well as the blockwise preconditioning setup. The following packages were used to solve the precondition system: HYPRE [15] as a multigrid solver and for the sparse direct solvers we use UMFPACK [10–13], PASTIX [19], SuperLU [26, 27] 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. By this we mean the mesh sequence, non-linear iteration stopping criteria and initial guess setup needs to be defined. In this small section we will briefly cover these aspects.

Mesh sequence

To test the performance of both the preconditioners from chapter 3 and to validate the code produced by FEniCS a sequence of meshes are required. Verifying the code produces the correct convergence rates, we consider levels, l, of uniform grids. The levels define the number of edges between the nodes along the boundary edge to be 2^l. For example, consider the third level, $l = 3$, hence the grid generated is the 8×8 grid given in figure 4.1. The first column of all convergence and iteration tables below will show the grid level.

Stopping criteria

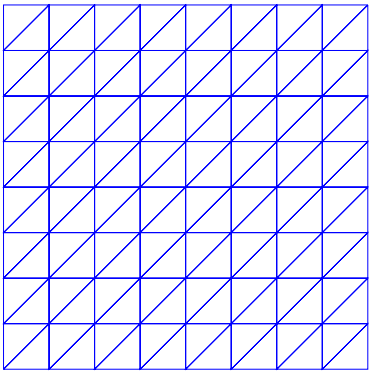

Figure 4.1: Level 3 grid on unit square domain

Recall that both the 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 is the following:

- NS: $\|\delta u\|_p + \|\delta p\|_p < \text{tol}$,
- MHD: $\|\delta u\|_p + \|\delta p\|_p + \|\delta b\|_p + \|\delta r\|_p < \text{tol}$,

where $p = \infty$ or 2. The tolerance is stated for each example run within this thesis.

Initial guess

To start the Picard iterations given in section 2.3 we require an initial

guess. To form the initial guess, we solve the decoupled Stokes problem

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

then the Maxwell subproblem

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

Here the term C corresponds the the coupling term using u (the initial guess for the velocity field).

When solving these two systems, the convergence tolerance which we use is important. In particular, inaccurate solutions cause problems with non-homogeneous boundary conditions. This is because if the matrix system with the boundary conditions applied is not solve to a sufficient accuracy then there will be errors in the boundary data. When we solve for the updates we enforce homogeneous Dirichlet boundary conditions (i.e. zero boundary conditions) on the velocity (u), magnetic (b) and multiplier (r) fields. This therefore means that if the Stokes and Maxwell solves are not done accurately enough then when solving for the updates errors will remain within the boundary conditions. Thus the errors norms for the full MHD system will be capped by the accuracy of the Stokes and Maxwell solve for the initial guess.

4.3 Navier-Stokes equations

Before considering the full discretised MHD problem, we require that the Navier-Stokes subproblem performs as expected in terms of the error estimates and preconditioner scalability. To check the validity of the FEniCS code, we introduce two smooth test problems, the first in 2D and the second in 3D.

4.3.1 2D smooth solution

Consider the domain $\Omega = (0,1)^2$ with boundary $\partial\Omega$. Since we are considering a purely Dirichlet problem then $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$. Taking the kinematic viscosity to be $\nu = 1$, then we choose the source terms $\boldsymbol{f}$ from the analytical solution

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

Running the FEniCS code produces the following table of results:

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	$\|p-p_h\|_{L^2(\Omega)}$	order
3	578/81	2.2100e-04	-	1.5645e-02	-	7.2758e-03	-
4	2,178/289	2.7598e-05	3.00	3.9133e-03	2.00	1.8190e-03	2.00
5	8,450/1,089	3.4465e-06	3.00	9.7822e-04	2.00	4.5025e-04	2.01
6	33,282/4,225	4.3072e-07	3.00	2.4455e-04	2.00	1.1341e-04	1.99
7	132,098/16,641	5.3836e-08	3.00	6.1137e-05	2.00	2.8639e-05	1.99
8	526,338/66,049	6.7291e-09	3.00	1.5284e-05	2.00	7.0146e-06	2.03

Table 4.1: Convergence table for 2D Navier-Stokes smooth solution - Picard tolerance 1e-10

For Taylor-Hood elements the expected convergence rates for the velocity field in the L^2 and H^1 integral norms are third and second order and for the pressure field the L^2 norm is second order. From table x4.1 we can see that we obtain third order in L^2, second order FOR H^1 in the velocity fields and second order in L^2 for the pressures. This is precisely what we expect from Taylor-Hood elements.

4.3.2 3D smooth solution

The three dimensional set up is very similar as with the 2D case. This time the domain is the unit cube, i.e. $\Omega = (0,1)^3$ with boundary $\partial\Omega$. Again the test problem has only Dirichlet boundary conditions, so that $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$. As before the kinematic viscosity is $\nu = 1$, then the source term $\boldsymbol{f}$ is 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 table 4.2 which shows the errors and convergence rates.

From the table we can see that the convergence rates are the same as the 2D test solution in the previous section for the velocity field. However, we see that for the pressure field we get a slightly higher than expected order, namely about 2.5. We have seen this trend for multiple different examples

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	$\|p-p_h\|_{L^2(\Omega)}$	order
1	375/27	2.6211e-02	-	4.5804e-01	-	1.7725e+00	-
2	2,187/125	3.2997e-03	2.99	1.1547e-01	1.99	2.8602e-01	2.63
3	14,739/729	4.1267e-04	3.00	2.8944e-02	2.00	4.0587e-02	2.82
4	107,811/4,913	5.1565e-05	3.00	7.2416e-03	2.00	6.4794e-03	2.65
5	823,875/35,937	6.4443e-06	3.00	1.8108e-03	2.00	1.2724e-03	2.35

Table 4.2: Convergence table for 3D Navier-Stokes smooth solution - Picard tolerance 1e-5

that have been run but never have we seen an order lower than 2.

4.3.3 Least Squares Commutator

Section 3.1 outlines the two main preconditioning techniques for the Navier-Stokes equations. In order to have a scalable preconditioner for the full MHD system we need to check the performance of both LSC and PCD for our Navier-Stokes subproblem.

First considered the LSC preconditioner. Recall that to apply the LSC preconditioner we need to apply

$$\begin{pmatrix} F & B^T \\ 0 & M_S \end{pmatrix}^{-1} \tag{4.3}$$

where

$$M_S = (BQ^{-1}B^T)(BQ^{-1}FQ^{-1}B^T)^{-1}(BQ^{-1}B^T)$$

at each GMRES iteration. This amounts to solving systems associated with F and $BQ^{-1}B^T$, which has been outlined in section 3.1. We do this by using a direct solve for F and approximate $BQ^{-1}B^T \approx B\,\mathrm{diag}(Q)^{-1}B^T$ then use

a direct solve. Since $B\,\mathrm{diag}(Q)^{-1}B^T$ is singular (the constant vector is in the null space) then we apply a small shift to the diagonal to ensure the invertability for the direct solver. Using GMRES with a relative tolerance stopping criteria of 1e-5 produces table 4.3 of iteration results.

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 4.3: Iteration table for LSC preconditioner for 2D example - Picard tolerance 1e-10

Table 4.3 shows that as the grid level increases the number of iterations GMRES takes to converge to the set tolerance increases. This is a relatively unknown problem when using Taylor-Hood elements. This therefore requires us to consider using PCD as the Navier-Stokes subproblem preconditioner instead.

4.3.4 Pressure Convection-Diffusion

As for LSC an application of (4.3) is applied at each GMRES iteration. However, for PCD the Schur complement approximation M_S is given by:

$$M_S = A_p F_p^{-1} W.$$

Instead of applying the PCD preconditioner exactly (direct solves) we will use approximate solves. We therefore use a single multigrid V-cycle to approximately solve both F and A_p and Conjugate Gradient (CG) [20] with a block Jacobi preconditioner and outer tolerance of 1e-5 for W. With the outer tolerance of GMRES the same as before we obtain the tables of iterations for both the 2D and 3D examples.

2D example

Running the 2D example gives table 4.4.

l	Dofs	Average iterations			
	$\boldsymbol{u}_h/p_h$	$\nu = 10$	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$
5	8450/1089	17	15	18	5053
6	33,282/4,225	18	15	18	144
7	132,098/16,641	18	15	18	41
8	526,338/66,049	18	15	18	40
9	2,101,250/263,169	18	15	18	40

Table 4.4: Iteration table for PCD preconditioner for 2D example - Picard tolerance 1e-10

From table 4.4 we can see that as the mesh level increases then the iteration numbers stay roughly constant. This is what we expect and require for the Navier-Stokes subproblem preconditioner. Also, the table shows iteration number for multiple different values of the kinematic viscosity ν. Again as expected, as the viscosity decreases (i.e. the equations become more convection dominated) the iterations increase.

3D example

Running the 3D code produces the following table.

l	Dofs	Average iterations			
	$\boldsymbol{u}_h/p_h$	$\nu = 10$	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$

Table 4.5: Iteration table for PCD preconditioner for 3D example - Picard tolerance 1e-10

4.4 Maxwell's equations

As for the Navier-Stokes subproblem (section 4.3), we consider a 2D and 3D test solution to validated the FEniCS code for the Maxwell subproblem.

4.4.1 2D smooth solution

Again, consider the domain $\Omega = (0,1)^2$ with boundary $\partial\Omega$. Since we are considering a purely Dirichlet problem then $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$. Now choose the source terms $\boldsymbol{g}$ that satisfy 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	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$	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 4.6: Convergence table for 2D 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	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 4.7: Convergence table for 2D Maxwell smooth solution - multiplier field

For Maxwell equations we use second order Nédélec elements of the first kind [32] and polynomial elements of the same order for the magnetic and multiplier fields respectively. Therefore, we exact second order convergence in $\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$ and $\|r-r_h\|_{H^1(\Omega)}$ and third order in $\|r-r_h\|_{L^2(\Omega)}$. Tables 4.6 and 4.7 show that we do obtain the expected rates of convergence for both the magnetic and multiplier fields.

4.4.2 3D smooth solution

As for the 3D Navier-Stokes problem the domain is the unit cube, i.e. $\Omega = (0,1)^3$ with boundary $\partial\Omega$. Again the test problem has only Dirichlet boundary conditions then $\Gamma_D = \partial\Omega$ and $\Gamma_N = \emptyset$. Using the anayltical 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 $\boldsymbol{g}$ is defined.

l	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$	order
1	436/125	5.3312e-02	0.00	3.5258e-01	0.00
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 4.8: Convergence table 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	0.00	2.8923e+00	0.00
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 4.9: Convergence table for 3D Maxwell smooth solution - multiplier field

Tables 4.8 and 4.9 show the same orders of convergence as in the 2D case from section 4.4.1.

4.4.3 Preconditioning

Section 3.2 introduces the preconditioning strategy that will be employed in this thesis. Recall that the preconditioner is:

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

where $N = M + X$ as defined in chapters 2 and 3. It is possible to form a scalable (i.e. iteration count stays the same as l increase) application

of $\mathcal{M}_{\mathrm{MX}}^{-1}$ using the algebraic multigrid method described in [21] for N and standard multigrid for L. This, however, is very complex to implement and hence application of $\mathcal{M}_{\mathrm{MX}}^{-1}$ will be done with direct solves for the numerical results in this section.

2D example

Table 4.10 shows the performance of using the preconditioner $\mathcal{M}_{\mathrm{MX}}$ with a MINRES tolerance of $1e-6$.

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 4.10: Iteration table for Maxwell preconditioner for 2D example - direct application of preconditioner

From table 4.10 we can see that as the number of MINRES iterations stays about constant as the mesh level, l, increases. Note that as the magnetic viscosity (ν_m) increases then the number of iterations remains roughly the same.

3D example

Table 4.11 presents the iteration numbers when using $\mathcal{M}_{\mathrm{MX}}$ as the preconditioner to MINRES with tolerance $1e-6$. Again, as with the 2D example we see that the number of iterations remains about constant with increasing

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

Table 4.11: Iteration table for Maxwell preconditioner for 3D example - direct application of preconditioner

mesh level and value of ν_m.

4.5 MHD problem

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

4.5.1 2D smooth solution

To validate the code, we consider the following $2D$ test problem. For this problem, let the domain be the unit square $\Omega = (0, 1)^2$ with purely Dirichlet boundary conditions on $\partial\Omega$. Let $\nu = \kappa = 1$, $\nu_m = 1e4$ then the source terms

$\boldsymbol{f}$ and $\boldsymbol{g}$ 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 asymptotic rates of convergence are given in tables 4.12 to 4.14.

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	$\|p-p_h\|_{L^2(\Omega)}$	order
1	50/9	8.1347e-02	0.00	1.2748e+00	0.00	2.8866e-01	0.00
2	162/25	1.0566e-02	2.94	3.3018e-01	1.95	3.7372e-02	2.95
3	578/81	1.3354e-03	2.98	8.3246e-02	1.99	4.8723e-03	2.94
4	2,178/289	1.6744e-04	3.00	2.0840e-02	2.00	7.2809e-04	2.74
5	8,450/1,089	2.0998e-05	3.00	5.2110e-03	2.00	1.5307e-04	2.25
6	33,282/4,225	2.6528e-06	2.98	1.3028e-03	2.00	3.7059e-05	2.05
7	132,098/16,641	3.4518e-07	2.94	3.2570e-04	2.00	9.2136e-06	2.01
8	526,338/66,049	4.9471e-08	2.80	8.1425e-05	2.00	2.3009e-06	2.00

Table 4.12: Convergence table for 2D MHD smooth solution - Picard tolerance 1e-8

The results shown in tables 4.12 to 4.14 agree with the optimal rates for $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{u}-\boldsymbol{u}_h\|_{H^1(\Omega)}$, $\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$, $\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$, $\|r-r_h\|_{L^2(\Omega)}$ and $\|r-r_h\|_{H^1(\Omega)}$. The pressure field exhibits a little higher than expected rate of convergence for $l < 5$ but settles down to second order for the higher levels.

l	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$	order
1	48/25	1.1206e-01	0.00	1.7807e-01	0.00
2	176/81	2.8439e-02	1.98	4.5429e-02	1.97
3	672/289	7.1388e-03	1.99	1.1414e-02	1.99
4	2,624/1,089	1.7867e-03	2.00	2.8571e-03	2.00
5	103,68/4,225	4.4679e-04	2.00	7.1450e-04	2.00
6	41,216/16,641	1.1171e-04	2.00	1.7864e-04	2.00
7	164,352/66,049	2.7927e-05	2.00	4.4661e-05	2.00
8	656,384/263,169	6.9818e-06	2.00	1.1165e-05	2.00

Table 4.13: Convergence table for 2D MHD 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	48/25	1.8438e-01	0.00	1.9449e+00	0.00
2	176/81	2.8307e-02	2.70	6.0694e-01	1.68
3	672/289	3.1214e-03	3.18	1.6510e-01	1.88
4	2,624/1,089	3.6035e-04	3.11	4.2417e-02	1.96
5	103,68/4,225	4.3890e-05	3.04	1.0683e-02	1.99
6	41,216/16,641	5.4485e-06	3.01	2.6757e-03	2.00
7	164,352/66,049	6.7988e-07	3.00	6.6923e-04	2.00
8	656,384/263,169	8.4948e-08	3.00	1.6733e-04	2.00

Table 4.14: Convergence table for 2D MDH smooth solution - multiplier field

4.5.2 3D smooth solution

As for the 3 dimensional examples we consider a test problem discretised on a unit cube domain with Dirichlet boundary conditions. Let $\nu = \kappa = 1$,

$\nu_m = 1\text{e}4$ and the analytical solution as:

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

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

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

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

then the source terms $\boldsymbol{f}$ and $\boldsymbol{g}$ are defined.

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	$\|p-p_h\|_{L^2(\Omega)}$	order
1	375/27	4.2196e-03	-	7.8282e-02	-	5.0614e-02	-
2	2187/125	5.2733e-04	3.00	1.9495e-02	2.01	9.0306e-03	2.49
3	14739/729	6.5749e-05	3.00	4.8664e-03	2.00	1.9035e-03	2.25
4	107811/4913	8.2092e-06	3.00	1.2161e-03	2.00	4.5311e-04	2.07

Table 4.15: Convergence table for 3D MHD smooth solution - Picard tolerance 1e-8

l	Dofs $\boldsymbol{b}_h/r_h$	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{L^2(\Omega)}$	order	$\|\boldsymbol{b}-\boldsymbol{b}_h\|_{H(curl,\Omega)}$	order
1	436/125	3.9931e-03	-	2.4817e-02	-
2	2936/729	1.1211e-03	1.83	7.4101e-03	1.74
3	21424/4913	2.9642e-04	1.92	1.9439e-03	1.93
4	163424/35937	7.5494e-05	1.97	4.9180e-04	1.98

Table 4.16: Convergence table for 3D MHD smooth solution - Magnetic field

From tables 4.15 to 4.17 it can be seen that we obtain the same conver-

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	2936/729	9.2457e-05	3.02	2.9368e-03	1.78
3	21424/4913	1.1182e-05	3.05	7.7197e-04	1.93
4	163424/35937	1.3829e-06	3.02	1.9597e-04	1.98

Table 4.17: Convergence table for 3D MDH smooth solution - multiplier field

gence orders as with the $2D$ case.

4.5.3 Parameter decoupling tests

There are two different decoupling schemes considered in thesis. These effectiveness of these approaches is likely to be limited by the parameter set up of the problem, i.e. the fluid viscosity (ν), magnetic viscosity (ν_m) and the coupling number (κ). In this section we will look at how these three schemes perform when fixing two parameters and varying the second. We will only look at varying κ and ν since ν_m appears within all three schemes. In the tables we denote (P) as the full MHD system given in (**??**), (MD) is the magnetic decoupling scheme (**??**) and (CD) is complete decoupling (**??**). Also, if the Picard iterations do not converge then this is denoted by - in the table.

Viscosity test

The first test we will consider is to vary the fluid viscosity results are represented in table 4.18. The parameter set up here is that the non-linear Picard tolerance is $1e-5$, coupling number $\kappa = 1$ and magnetic viscosity $\nu_m = 10$.

As the fluid viscosity (ν) decreases then the equations that determine

		$\nu = 1$			$\nu = 0.1$			$\nu = 0.01$			$\nu = 0.001$		
l	DoF	(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 4.18: Number of non-linear iterations for various values of ν. Let the Picard tolerance to be $1e-5$, $\kappa = 1$ and $\nu_m = 10$.

the fluid flow become more convection dominated. Thus we would expect that the complete decoupling scheme might break down for small ν as for this decoupling scheme we remove the convection term. This is precisely what we what we see from the results in table 4.18. However, the full MHD problem and magnetic decoupling cases perform very similarly in terms of the number of Picard iterations it takes to converge to the solution for $\kappa = 1$. For $\nu = 0.0001$ all schemes break down and do not converge. This maybe due to the stability of the convection discretisation.

Coupling number test

The other parameter test that we exam is to vary the coupling term. This test shows how increasing the coupling term effects the number of non-linear iterations for each of the three schemes. We expect that the full Picard (P) scheme would perform best with both the magnetic and complete decoupling schemes breaking down as κ increases. Table 4.19 shows that this is what happens. Notice that the complete decoupling scheme completely breaks down for $\kappa > 100$ whereas the number of non-linear iterations for magnetic

		$\kappa = 0.1$			$\kappa = 1$			$\kappa = 10$			$\kappa = 100$			$\kappa = 1000$		
l	DoF	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)	(P)	(MD)	(CD)
3	1,620	5	5	13	5	5	13	6	7	17	7	13	26	7	93	-
4	6,180	5	5	13	5	5	15	6	7	18	7	13	26	7	93	-
5	24,132	5	5	13	5	5	15	6	7	18	7	13	26	7	93	-
6	95,364	5	5	14	5	5	15	6	7	18	7	13	26	7	93	-
7	379,140	5	5	14	5	5	15	6	7	18	7	13	26	7	93	-
8	1,511,940	5	5	14	5	5	15	6	7	18	7	13	26	7	93	-

Table 4.19: Number of non-linear iterations for various values of ν_m. LLet the Picard tolerance to be $1e-5$, $\nu = 1$ and $\nu_m = 1e2$.

decoupling stays roughly constant for each mesh level but is larger than for the full Picard iterations.

4.6 Preconditioned MHD problem

Sections 3.3 and 3.4.1 introduced the three preconditioning techniques that will be applied to each of the decoupling schemes and the full MHD system. In this section we will look at the performances of these strategies with respect to time and iteration count for various tolerances.

4.6.1 MHD problem

Recall form section 3.3 that the preconditioner for the full MHD problem $\mathcal{K}_{\text{MH}}$ given in (2.9) is

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

where S will be the PCD approximation to the Schur complement of the Navier-Stokes subproblem. To apply $\mathcal{M}_{\text{MH}}$ as the preconditioner we employ an inner Krylov solver where the preconditioner is the same as $\mathcal{M}_{\text{MH}}$ but importantly we drop the coupling terms, i.e.

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

There are four solves (F, S, N and L) to apply $\mathcal{M}_{\text{innerMH}}$ as an inner preconditioner. Each will be a direct solve apart from the PCD approximation in which apply $S^{-1} = W^{-1}F_pA_p^{-1}$ where W^{-1} and A_p^{-1} will be direct solves as well.

4.6.2 Magnetic decoupling

l	DoF	Soln Time	picard iterations	NS iters	M iters
3	1620	0.039161	8	52	3
4	6180	0.097946	8	48	3
5	24132	0.447303	8	52	3
6	95364	2.109198	8	52	3
7	379140	9.697754	8	50	3
8	1511940	46.675882	8	51	3
9	6038532	539.038188	8	51	3

Table 4.20: $tol = 1e-5 kappa = 1.0/2 Mu_m = 1e2 MU = 0.1$

4.6.3 Complete decoupling

l	DoF	AV solve Time	Total picard time	picard iterations	Av NS iters	Av M iters
3	1620	0.030	0.32	5	29.4	2.6
4	6180	0.072	0.72	5	29.4	2.6
5	24132	0.290	2.36	5	29.0	2.8
6	95364	1.552	11.14	5	29.0	2.8
7	379140	7.164	48.96	5	29.0	2.8
8	1511940	36.180	232.27	5	29.0	2.8
9	6038532	159.17	1618.24	5	29.0	2.8

Table 4.21: $tol = 1e-5 kappa = 1.0/2 Mu_m = 1e2 MU = 10$

Chapter 5

Conclusion

Bibliography

[1] Martin S. Alnæs. *UFL: a Finite Element Form Language*, chapter 17. Springer, 2012.

[2] Martin S. Alnæs, Anders Logg, and Kent-Andre Mardal. *UFC: a Finite Element Code Generation Interface*, chapter 16. Springer, 2012.

[3] Martin S. Alnæs, Anders Logg, Kent-Andre Mardal, Ola Skavhaug, and Hans Petter Langtangen. Unified framework for finite element assembly. *International Journal of Computational Science and Engineering*, 4(4):231–244, 2009.

[4] Martin S. Alnæs, Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. Unified form language: A domain-specific language for weak formulations of partial differential equations. *ACM Transactions on Mathematical Software*, To appear, 2013.

[5] Patrick R Amestoy, Iain 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] Patrick R Amestoy, Iain S Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dy-

namic scheduling. *SIAM Journal on Matrix Analysis and Applications*, 23(1):15–41, 2001.

[7] Patrick R Amestoy, Abdou Guermouche, Jean-Yves LExcellent, and Stéphane Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel computing*, 32(2):136–156, 2006.

[8] Satish Balay, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013.

[9] Satish Balay, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, and Hong Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2014.

[10] Timothy A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):196–199, June 2004.

[11] Timothy A. Davis. A column pre-ordering strategy for the unsymmetric-pattern multifrontal method. *ACM Trans. Math. Softw.*, 30(2):165–195, June 2004.

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

[13] Timothy A. Davis and Iain S. Duff. A combined unifrontal/multifrontal method for unsymmetric sparse matrices. *ACM Trans. Math. Softw.*, 25(1):1–20, March 1999.

[14] Howard C Elman, David J Silvester, and Andrew J Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics: with Applications in Incompressible Fluid Dynamics.* Oxford University Press, 2005.

[15] Robert D Falgout and Ulrike Meier Yang. hypre: A library of high performance preconditioners. In *Computational ScienceICCS 2002*, pages 632–641. Springer, 2002.

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

[17] Chen Greif and D Schötzau. Preconditioners for saddle point linear systems with highly singular (1, 1) blocks. *ETNA, Special Volume on Saddle Point Problems*, 22:114–121, 2006.

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

[19] Pascal Hénon, Pierre Ramet, and Jean Roman. Pastix: a high-performance parallel direct solver for sparse symmetric positive definite systems. *Parallel Computing*, 28(2):301–321, 2002.

[20] Magnus Rudolph Hestenes and Eduard Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS, 1952.

[21] Ralf Hiptmair and Jinchao Xu. Nodal auxiliary space preconditioning in h (curl) and h (div) spaces. *SIAM Journal on Numerical Analysis*, 45(6):2483–2509, 2007.

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

[23] Robert C. Kirby. *FIAT: Numerical Construction of Finite Element Basis Functions,*, chapter 13. Springer, 2012.

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

[25] Dan Li. Numerical solution of the time-harmonic maxwell equations and incompressible magnetohydrodynamics problems. 2010.

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

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

[28] Anders Logg, Kristian B. Ølgaard, Marie E. Rognes, and Garth N. Wells. *FFC: the FEniCS Form Compiler*, chapter 11. Springer, 2012.

[29] Anders Logg and Garth N. Wells. Dolfin: Automated finite element computing. *ACM Transactions on Mathematical Software*, 37(2), 2010.

[30] Anders Logg, Garth N. Wells, and Johan Hake. *DOLFIN: a C++/Python Finite Element Library*, chapter 10. Springer, 2012.

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

[32] Jean-Claude Nédélec. Mixed finite elements in 3. *Numerische Mathematik*, 35(3):315–341, 1980.

[33] Kristian B. Ølgaard and Garth N. Wells. Optimisations for quadrature representations of finite element tensors through automated code generation. *ACM Transactions on Mathematical Software*, 37, 2010.

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

[35] Youcef Saad and Martin 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.

[36] Dominik Schötzau. Mixed finite element methods for stationary

incompressible magneto–hydrodynamics. *Numerische Mathematik*, 96(4):771–800, 2004.

[37] David Silvester and Andrew Wathen. Fast iterative solution of stabilised stokes systems. part i: Using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis*, 30(3):630–649, 1993.

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

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

[40] Garth Wells, Kent-Andre Mardal, and Anders Logg. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer, 2012.