Numerical Solution of the Time-harmonic Maxwell Equations and Incompressible Magnetohydrodynamics Problems

by

Dan Li

B.Eng., Beijing University of Aeronautics and Astronautics, 1999

M.Eng., Beijing University of Aeronautics and Astronautics, 2002

M.A.Sc., The University of British Columbia, 2005

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

Doctor of Philosophy

in

THE FACULTY OF GRADUATE STUDIES
(Computer Science)

The University Of British Columbia
(Vancouver)

November 2010

Abstract

The goal of this thesis is to develop efficient numerical solvers for the time-harmonic Maxwell equations and for incompressible magnetohydrodynamics problems.

The thesis consists of three components. In the first part, we present a fully scalable parallel iterative solver for the time-harmonic Maxwell equations in mixed form with small wave numbers. We use the lowest order Nédélec elements of the first kind for the approximation of the vector field and standard nodal elements for the Lagrange multiplier associated with the divergence constraint. The corresponding linear system has a saddle point form, with inner iterations solved by preconditioned conjugate gradients. We demonstrate the performance of our parallel solver on problems with constant and variable coefficients with up to approximately 40 million degrees of freedom. Our numerical results indicate very good scalability with the mesh size, on uniform, unstructured and locally refined meshes.

In the second part, we introduce and analyze a mixed finite element method for the numerical discretization of a stationary incompressible magnetohydrodynamics problem, in two and three dimensions. The velocity field is discretized using divergence-conforming Brezzi-Douglas-Marini (BDM) elements and the magnetic field is approximated by curl-conforming Nédélec elements. Key features of the method are that it produces exactly divergence-free velocity approximations, and that it correctly captures the strongest magnetic singularities in non-convex polyhedral domains. We prove that the energy norm of the error is convergent in the mesh size in general Lipschitz polyhedra under minimal regularity assumptions, and derive nearly optimal a-priori error estimates for the two-dimensional case. We present a comprehensive set of numerical experiments, which indicate optimal convergence of the proposed method for two-dimensional as well as three-dimensional problems.

Finally, in the third part we investigate preconditioned Krylov iterations for the discretized stationary incompressible magnetohydrodynamics problems. We propose a preconditioner based on efficient preconditioners for the Maxwell and Navier-Stokes sub-systems. We show that many of the eigenvalues of the preconditioned system are tightly clustered, and hence, rapid convergence is accomplished. Our numerical results show that this approach performs quite well.

Preface

This thesis describes results in three papers.

i. C. Greif, D. 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*, 199:2840-2855, 2010.

ii. D. Li. Parallel numerical solution of the time-harmonic Maxwell equations. *Lecture Notes in Computer Science on High Performance Computing and Applications*, 5938:224-229, 2010.

iii. D. Li, C. Greif, and D. Schötzau. Parallel numerical solution of the time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, Submitted (19 pages), 2010.

The first two are published and the third is in review. Chapter 2 describes the work in [ii, iii], and Chapter 3 is based on the work published in [i].

I have taken a central role in all aspects of the work on these papers, as follows. Paper [ii] is a single-authored conference publication, and I worked on it and prepared it on my own. I was the lead author in Paper [iii], which is joint with my research supervisors, Chen Greif and Dominik Schötzau. Paper [i] is joint with my research supervisors and with Xiaoxi Wei, who is a PhD student supervised by Dominik Schötzau in the UBC Mathematics Department. My focus in this paper was on the numerical issues and the implementation, while my colleague Xiaoxi Wei has focused on the finite element analysis. For this reason, my thesis mainly discusses numerical results for this part, whereas some of the theoretical results will be presented in X. Wei's thesis [95].

Table of Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisors, Chen Greif and Dominik Schötzau, for their wonderful guidance and support. I am greatly indebted for the countless hours of discussions, and for the conversations that encouraged me to continue when I was worn down. Without their help, this thesis would not have been possible. I am grateful to my supervisory committee, including Robert Bridson and Carl Ollivier-Gooch for giving me insightful and constructive feedback. I am thankful to my external examiner Sabine Le Borne, as well as my university examiners Felix Herrmann and Brian Wetton, for their valuable feedback.

I would also like to thank my friends. My graduate study would not have been so pleasant and memorable without them. I am thankful to all my friends in the Scientific Computing Lab. I am also thankful to Rong Yi, Haijun Qiao, Kangkang Yin, and Xiaoxi Wei.

Finally, I would like to thank my family. Their continuous love, support, encouragement and confidence in me have been invaluable. I could not have done this work without them. Mom and dad, this dissertation is also yours. Though you did not have a chance to pursue a PhD degree, you deserve one! Grandma and grandpa, I hope I have made you proud. My dearest husband, Penglong Wang, thank you for always being there when I need you. You are the sunshine in my life.

Dedication

To my husband, parents and grandparents

Chapter 1

Introduction

The main topic of this thesis is the efficient numerical solution of the time-harmonic Maxwell and incompressible magnetohydrodynamics equations. These two problems are central themes in the study of physical phenomena in the field of computational electromagnetics [55, 60, 73]. In this introductory chapter, we first give a brief overview of Krylov subspace methods. Following that we present an introduction to scientific parallel computing. We then provide the physics of the two problems we study. Finally, an overview of the thesis structure and contributions is given.

1.1 Krylov subspace methods

For solving linear systems of a reasonable size, direct methods are very popular in many applications in science and engineering. For large systems, iterative methods are more appropriate. We are interested in iterative solvers, particularly, preconditioned Krylov subspace solvers.

Suppose we are to solve

$$Ax = b$$

iteratively, with an initial guess x_0 and initial residual $r_0 = b - Ax_0$. For any real, $n \times n$ nonsingular matrix A and vector y of the same length, the k-dimensional

Krylov subspace of A with respect to y is defined by

$$\mathcal{K}_k(A;y) = \text{span}\{y, Ay, A^2y, \ldots, A^{k-1}y\}.$$

Krylov subspace methods are based on finding a solution within the shifted Krylov subspace $x_0 + \mathcal{K}_k(A;r_0)$ which satisfies a certain optimality criterion. There are three important building blocks for the family of Krylov subspace solvers:

1. construct an orthogonal basis for the Krylov subspace;

2. define an optimality property;

3. use an effective preconditioner.

Computing an orthogonal basis. The vectors $\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}$ span the Krylov subspace $\mathcal{K}_k(A,r_0)$, but as j increases, A^jb approaches the dominant eigenvector of A and the vectors become more linearly dependent. As a result, it is more appropriate to consider an orthonormal basis for $\mathcal{K}_k(A,r_0)$. If A is unsymmetric, the Arnoldi process is used to construct the basis iteratively. Each iteration adds an orthogonal vector to the basis. This process can be written as

$$AQ_k = Q_{k+1}H_{k+1,k},$$

where Q_{k+1} is the matrix containing the $k+1$ vectors of the orthogonal basis for the Krylov subspace, Q_k is the same matrix but with the first k columns only, and $H_{k+1,k}$ is a matrix in upper Hessenberg form of size $(k+1)\times k$. It is easy to show that

$$Q_k^T AQ_k = H_{k,k},$$

where $H_{k,k}$ a matrix containing the first k rows of $H_{k+1,k}$. When A is symmetric, $H_{k,k}$ must be symmetric. Therefore, it must be tridiagonal; let us denote it by $T_{k,k}$. In this case the Arnoldi process reduces to the well known Lanczos method. This process can be written as

$$AQ_k = Q_{k+1}T_{k+1,k}.$$

Defining an optimality property. Now that the orthogonal basis is available, we need to define an optimality criterion for a Krylov subspace solver. There are various alternatives for deciding on the type of solution we are looking for within the Krylov subspace. Two particularly popular criteria are the following:

- seek a residual with minimum ℓ_2-norm within the Krylov subspace,
- force the residual to be orthogonal to the Krylov subspace.

The first approach leads to MINRES (for minimum residual) when A is symmetric [79] and GMRES (for generalized minimum residual) when A is nonsymmetric [86]. The second approach leads to the SYMMLQ algorithm [79] when A is symmetric. When A is nonsymmetric, the FOM (for full orthogonalization method) algorithm works [85]. When A is symmetric positive definite, the second approach is equivalent to minimizing the energy norm of the error, $||x_k - x||_A$ [33, Theorem 6.8] and after some manipulation we get the CG (conjugate gradient) algorithm [53].

In this thesis, the numerical solvers we use are MINRES, GMRES and CG. We discuss these three algorithms briefly.

- GMRES for general matrices. By the Arnoldi algorithm, we have that $AQ_k = Q_{k+1}H_{k+1,k}$. We can write

  $$x_k = x_0 + Q_k z,$$

  and aim to find x_k which minimizes $||b - Ax_k||$. We solve the corresponding least-squares problem through the QR factorization and derive that

  $$z = R_{k,k}^{-1} U_{k+1,k}^T ||r_0|| e_1.$$

  Here,

  $$H_{k+1,k} = U_{k+1,k} R_{k,r},$$

  where $R_{k,k}$ is an upper triangular matrix and $U_{k+1,k}$ consists of k orthonormal vectors. The vector e_1 is $(1,0,\ldots,0)^T$. Once we have z, we compute the kth iteration simply by setting $x_k = x_0 + Q_k z$.

- MINRES for symmetric matrices. For symmetric matrices A, Arnoldi is replaced by Lanczos and the upper Hessenberg matrix is just tridiagonal.

The same mechanism as GMRES can be applied but the resulting iterative method is simpler and requires short three-term recurrence relations; this is MINRES.

- CG for symmetric positive definite matrices. We write

  $$x_k = x_0 + Q_k z,$$

  and impose $Q_K^T r_k = 0$. It can be shown that

  $$z = T_{k,k}^{-1} ||r_0|| e_1.$$

  Further manipulations related to the Cholesky decomposition of $T_{k,k}$ lead to the elegant CG algorithm; see, e.g., [33, Chatper 6].

Preconditioning The convergence rate of Krylov subspace solvers depends on the distribution of A's eigenvalues. Preconditioning means replacing the system $Ax = b$ with the system $M^{-1}Ax = M^{-1}b$, where M is an approximation to A. There are several important properties a preconditioner should have [10, 33, 93]:

- the system $Mz = s$ must be easy to solve,
- the matrix $M^{-1}A$ is well conditioned and/or has strongly clustered eigenvalues,
- the matrix M must not be too difficult to construct.

Loosely speaking, there are two general approaches to constructing preconditioners [10]. One approach is to design specialized algorithms in applications involving partial differential equations (PDEs); see, e.g., [37]. These algorithms are usually optimal (or nearly so) for a narrow class of problems, but they require complete knowledge of the problem at hand, including the original (continuous) equations, the domain of integration, the boundary conditions, details of the discretization, and so forth. The second approach is to design general-purpose, purely algebraic methods that use only information contained in the coefficient matrix A; see, e.g., [85]. In this thesis, since we have an underlying PDE, we design our preconditioners based on the first methodology.

1.2 Scientific parallel computing

One of the core tasks in the numerical simulation of mathematical models is solving very large linear systems of equations. Indeed, the number of equations may reach millions very quickly as we refine the mesh, especially in three dimensions. Sequential single-processor solution procedures are often prohibitive in terms of CPU time and memory requirements. In this thesis we develop a parallel computing paradigm to derive efficient scalable linear solvers.

Parallel computing is the application of two or more processing units to solve a single problem [81, 90]. These units can be physical processors or logical processes. Our parallel numerical examples are run on a cluster with distributed memory machines with multiple processors. Each processor executes the same program, but with different data.

To accomplish the task of developing an efficient parallel code, a few issues need to be addressed.

Data access. Since processors calculate values that are needed by other processors, we need a way to distinguish among those data at different processors. It is useful to have a processor-centric mechanism whereby the data of a processor under consideration are considered *on-processor* data and all the rest are *off-processor* data. Accessing off-processor data is often referred to as *communication*. In using multiple processors to solve a problem, varying degrees of coordination are required. Coordination primarily resolves accessing off-processor data required by some processors to compute its tasks. There are two ways to access off-processor data.

1. *Direct memory access.* In this method, the processor requesting information will access the off-processor information directly from memory where it was written. This requires some form of synchronization so that the value in memory is accessed when it is valid. The approach is typically used in shared memory parallelism. An intrinsic characteristic of shared memory machines is a strategy for memory coherence and a fast tightly coupled network for distributing data from a commonly accessible memory system.

2. *Data exchange via messages.* This is another method to acquire off-processor information using messages. In a message, the required information is pack-

aged, identified, and sent from the processor that defined it to the processor that requires it. Thus, messages have both the information and synchronization. This approach to accessing off-processor data is typically used in distributed memory architectures. We use messages to exchange data between processors in this thesis.

Load balancing. The overall time spent in parallel programs includes the computational time and communication time. Load balancing is very important in parallel computing. If the work is not distributed equally, then one processor may end up taking longer than the others. Since we are doing a cooperative project, the entire job cannot be finished until the slowest subtask is finished. To reduce the computational time, in addition to optimizing the parallel algorithms, one requirement is that the work to be done by each processor needs to be balanced among all the processors. Reducing communication cost is another import factor affecting efficiency of parallel programs. Even if the load is well balanced, the communication cost can be high, if all processors need to access a lot of off-processor data. To reduce the communication time, we need to design the data structures that each processor will require minimum off-processor data.

Numerical libraries. To write sophisticated parallel application codes that involve the numerical simulation of large-scale engineering applications, we require the use of portable, efficient numerical libraries. Over the last few decades, various agencies world-wide have invested heavily in making parallel computing usable for more than very special high-budget custom projects. There are many numerical libraries available for this purpose, such as MUMPS [1], SuperLU_DIST [71], PARDISO [87], Trilinos [52], Hypre [38], and PETSc [6–8]. We adopt PETSc (the Portable, Extensible Toolkit for Scientific Computation) as the framework of our parallel code in this thesis. PETSc uses the MPI standard [46] for all message passing communications. To accomplish maximum flexibility PETSc utilizes the object-oriented methodology, which in turn seamlessly allows for a high level of abstraction; see Figure 1.1. The user initiates communications of sequential and parallel phases of computations, but PETSc handles the detailed message passing required during the coordination of the computations. This provides a good balance between ease of use and efficiency of implementation. Each part of the

PETSc library manipulates a particular family of objects (vectors, matrices, Krylov subspaces, preconditioners, etc.) and operations that can be performed on them. Matrices are distributed as block rows in PETSc (i.e., block distribution, see [81, Chapter 5] for details), such that each process holds an array of contiguous rows of the matrix and owns complete rows. Our data structures are designed to meet this data distribution scheme; see Section 2.3.4 for details.

There are many open source finite element libraries available, such as FEMSTER [16], NGSolve [88], and Deal.II [9]. Instead of using any of these libraries we have written our own code, because it gives us easy access to the matrices and allows us to numerically test the various solvers studied in this work.

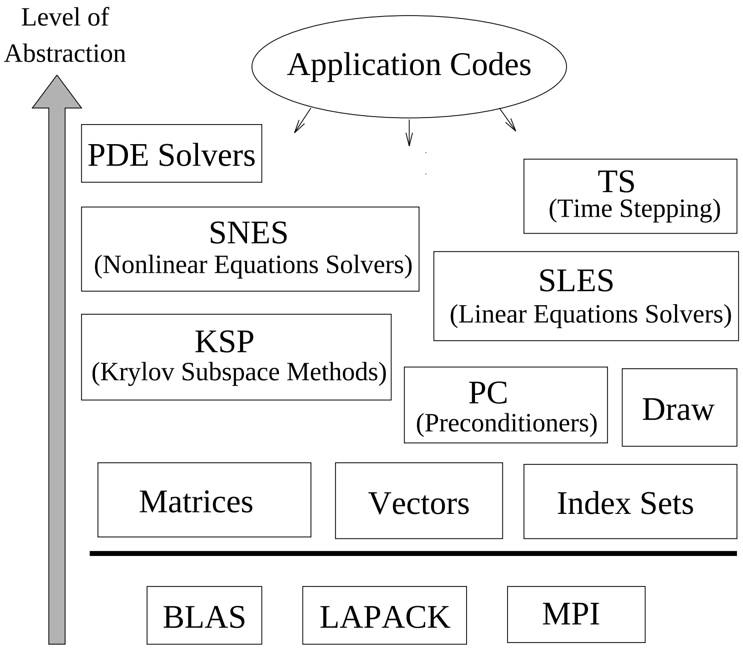


Figure 1.1: Organization of the PETSc library. The objects above the line are manipulated by PETSc. Objects at a higher level of abstraction are based on objects below them.

1.3 The problems and their physics

1.3.1 The time-harmonic Maxwell equations

The modern theory of electromagnetism was founded by James Clerk Maxwell (1831-1879) in 1873 with the publication of his *A Treatise on Electricity and Magnetism* [72]. The Maxwell equations are a system of PDEs that describe how the electric and magnetic fields relate to their sources, charge density and current density.

The classical macroscopic electromagnetic field is described by four vector functions of position $\mathbf{x} \in \mathbb{R}^3$ and time $t \in \mathbb{R}$, denoted by $\mathscr{E}$, $\mathscr{D}$, $\mathscr{H}$ and $\mathscr{B}$ [73]. The fundamental field vectors are $\mathscr{E}$ and $\mathscr{H}$, called the electric and magnetic field intensities, respectively. The vector functions $\mathscr{D}$ and $\mathscr{B}$ are called the electric displacement and magnetic induction, respectively. They will later be eliminated from the description of the electromagnetic field via suitable constitutive relations. Our exposition in this section mainly follows [32, 73].

An electromagnetic field is created by a distribution of sources consisting of static electric charges and the directed flow of electric charge, the current. The distribution of charges is given by a scalar charge density function ρ, while currents are described by the vector current density function $\mathscr{J}$. The Maxwell equations then state that the field variables and sources are related by the following equations:

Farady's law:
$$\frac{\partial \mathscr{B}}{\partial t} + \nabla \times \mathscr{E} = \mathbf{0}, \tag{1.1a}$$

Coulomb's law:
$$\nabla \cdot \mathscr{D} = \rho, \tag{1.1b}$$

Ampère's law:
$$\frac{\partial \mathscr{D}}{\partial t} - \nabla \times \mathscr{H} = -\mathscr{J}, \tag{1.1c}$$

Gauss's law:
$$\nabla \cdot \mathscr{B} = 0. \tag{1.1d}$$

Since charge is conserved, ρ and $\mathscr{J}$ are connected by the relation

$$\nabla \cdot \mathscr{J} + \frac{\partial \rho}{\partial t} = 0. \tag{1.2}$$

We assume the current density and charge density are time-harmonic:

$$\begin{aligned}
\mathscr{J}(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\mathbf{J}}(\mathbf{x})),\\
\rho(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\rho}(\mathbf{x})),
\end{aligned}$$

where $\mathrm{i} = \sqrt{-1}$, $\omega > 0$ is the temporal frequency and $\Re(\cdot)$ denotes the real part of the expression. Then the electromagnetic fields are time-harmonic:

$$\begin{aligned}
\mathscr{E}(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\mathbf{E}}(\mathbf{x})),\\
\mathscr{D}(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\mathbf{D}}(\mathbf{x})),\\
\mathscr{H}(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\mathbf{H}}(\mathbf{x})),\\
\mathscr{B}(\mathbf{x},t) &= \Re(\exp(-\mathrm{i}\omega t)\hat{\mathbf{B}}(\mathbf{x})).
\end{aligned}$$

The field $\hat{\mathbf{E}}$ is now a complex-valued vector function of position but not time.

Substituting these relations into (1.1) leads to the time-harmonic Maxwell equations:

$$-\mathrm{i}\omega\hat{\mathbf{B}} + \nabla\times\hat{\mathbf{E}} = \mathbf{0}, \tag{1.3a}$$

$$\nabla\cdot\hat{\mathbf{D}} = \hat{\rho}, \tag{1.3b}$$

$$-\mathrm{i}\omega\hat{\mathbf{D}} - \nabla\times\hat{\mathbf{H}} = -\hat{\mathbf{J}}, \tag{1.3c}$$

$$\nabla\cdot\hat{\mathbf{B}} = 0, \tag{1.3d}$$

where the time-harmonic charge density $\hat{\rho}$ is given via charge conservation (1.2) and hence can be eliminated from the equations.

The equations in (1.3) must be augmented by two constitutive laws that relate $\hat{\mathbf{E}}$ and $\hat{\mathbf{H}}$ to $\hat{\mathbf{D}}$ and $\hat{\mathbf{B}}$, respectively. These laws depend on the properties of the matter in the domain occupied by the electromagnetic field. We assume the linear relation

$$\hat{\mathbf{D}} = \varepsilon\hat{\mathbf{E}} \quad \text{and} \quad \hat{\mathbf{B}} = \mu\hat{\mathbf{H}}, \tag{1.4}$$

where ε and μ denote, respectively, the *permittivity* (measured in F/m in SI units) and *permeability* (H/m), which are positive, bounded, scalar functions of position. In a vacuum, we have $\varepsilon \equiv \varepsilon_0$ and $\mu \equiv \mu_0$, where $\varepsilon_0 = 8.854\times10^{-12}F/m$ and

$\mu_0 = 4\pi \times 10^{-7} H/m$.

There is one additional constitutive relation. In a conducting material, the electromagnetic field itself gives rise to currents. If the field strengths are not large, we can assume that Ohm's law holds so that:

$$\hat{\mathbf{J}} = \sigma\hat{\mathbf{E}} + \hat{\mathbf{J}}_a, \tag{1.5}$$

where σ is called the *conductivity* (S/m) and is a non-negative function of position. The vector function $\hat{\mathbf{J}}_a$ describes the applied current density.

Using the constitutive equations in (1.4) and the constitutive relation for the currents in (1.5), we use (1.3) to obtain the following time-harmonic Maxwell system:

$$-\mathrm{i}\omega\mu\hat{\mathbf{H}} + \nabla\times\hat{\mathbf{E}} = \mathbf{0}, \tag{1.6a}$$

$$\nabla\cdot(\varepsilon\hat{\mathbf{E}}) = \frac{1}{\mathrm{i}\omega}\nabla\cdot(\sigma\hat{\mathbf{E}} + \hat{\mathbf{J}}_a), \tag{1.6b}$$

$$-\mathrm{i}\omega\varepsilon\hat{\mathbf{E}} + \sigma\hat{\mathbf{E}} - \nabla\times\hat{\mathbf{H}} = -\hat{\mathbf{J}}_a, \tag{1.6c}$$

$$\nabla\cdot(\mu\hat{\mathbf{H}}) = 0. \tag{1.6d}$$

Define

$$\mathbf{E} = \varepsilon_0^{1/2}\hat{\mathbf{E}} \quad \text{and} \quad \mathbf{H} = \mu_0^{1/2}\hat{\mathbf{H}}.$$

Using (1.6) and defining the relative permittivity and permeability by

$$\varepsilon_r = \frac{1}{\varepsilon_0}(\varepsilon + \frac{\mathrm{i}\sigma}{\omega}) \quad \text{and} \quad \mu_r = \frac{\mu}{\mu_0},$$

the following version of the first-order Maxwell system is obtained:

$$-\mathrm{i}k\mu_r\mathbf{H} + \nabla\times\mathbf{E} \;=\; \mathbf{0}, \tag{1.7a}$$

$$-\mathrm{i}k\varepsilon_r\mathbf{E} - \nabla\times\mathbf{H} \;=\; -\frac{1}{\mathrm{i}k}\mathbf{f}, \tag{1.7b}$$

where $\mathbf{f} = \mathrm{i}k\mu_0^{1/2}\hat{\mathbf{J}}_a$ and the wave number is $k = \omega\sqrt{\varepsilon_0\mu_0}$. The following divergence conditions follow from (1.6):

$$
\begin{aligned}
\nabla \cdot (\varepsilon_r \mathbf{E}) &= -\frac{1}{k^2} \nabla \cdot \mathbf{f}, \\
\nabla \cdot (\mu_r \mathbf{H}) &= 0.
\end{aligned}
$$

The magnetic field **H** can be eliminated by solving (1.7a) for **H** and substituting into (1.7b). Thus, we obtain the second-order Maxwell system

$$
\nabla \times (\mu_r^{-1} \nabla \times \mathbf{E}) - k^2 \varepsilon_r \mathbf{E} = \mathbf{f}. \tag{1.8}
$$

Equation (1.8) is the model problem we will consider in Chapter 2. We will consider a lossless medium ($\sigma = 0$) in which case the material coefficients are real; and the real and imaginary parts of (1.8) decouple into two problems of the same form. From now on, we assume **E** and **f** are real.

Computer simulation has become an essential tool in the design and analysis of electronic and electromechanical devices, such as micro-chips, medical equipment, transformers, sensors and coils. Nowadays, there are many commercial software packages available, such as Maxwell by Ansoft (www.ansoft.com), Microwave Office by AWR (web.awrcorp.com), CST EM studio by CST (www.cst.com), and COMSOL Multiphysics by COMSOL (www.comsol.com).

There are a number of solvers available for (1.8). For example, a parallel solver based on virtual space preconditioners has been implemented in Hypre [38] and a Maxwell solver based on AMG preconditioners has been developed in [39].

1.3.2 Incompressible magnetohydrodynamics

The field of magnetohydrodynamics (MHD) studies the behavior of electrically conducting fluids (such as liquid metals, plasmas, salt water, etc.) in electromagnetic fields [30, 42, 74]. The equations of electromagnetics and fluid dynamics are coupled through two fundamental effects: first, the motion of a conducting material in the presence of a magnetic field induces an electric current that modifies the existing electromagnetic field (this effect is often referred to as electromotive force (EMF)). Secondly, the current and the magnetic field generate the Lorentz force, which accelerates the fluid particles in the direction normal to both the magnetic field and the electric current. Our focus is on incompressible viscous fluids whose electric resistivity is non-negligible [42]. The corresponding incompressible MHD model is a system of PDEs, where the Navier-Stokes equations are coupled with the Maxwell equations. Incompressible MHD has a number of technological and industrial applications such as metallurgical engineering, electromagnetic pumping, stirring of liquid metals, and measuring flow quantities based on induction; cf. [28, 42].

In the MHD equations, the non-relativistic motion of Newtonian fluids is governed by the Navier-Stokes equations as follows (see, e.g., [18])

$$\frac{\partial \mathscr{U}}{\partial t} - \nu_v \Delta \mathscr{U} + (\mathscr{U} \cdot \nabla)\mathscr{U} - \frac{1}{\rho}\nabla \bar{\mathscr{P}} = \mathbf{f}_L + \mathbf{f}, \tag{1.9a}$$

$$\nabla \cdot \mathscr{U} = 0. \tag{1.9b}$$

Here $\mathscr{U}$ is the velocity field, ν_v is the kinematic viscosity of the fluid, $\bar{\mathscr{P}}$ is the pressure of the fluid, $\mathbf{f}_L$ is the Lorentz force per mass, and $\mathbf{f}$ is the external body force per mass. The constraint (1.9b) corresponds to conservation of mass.

Electromagnetic effects are governed by the Maxwell equations in (1.1). Recall that ε and μ are functions of position only. We assume

$$\mathscr{D} = \varepsilon \mathscr{E} \quad \text{and} \quad \mathscr{B} = \mu \mathscr{H}. \tag{1.10}$$

If the densities of positive and negative charges are assumed to be equal in any sizable region (quasi-neutrality assumption, see, e.g., [59] for details), Ohm's law simplifies to:

$$\mathscr{J} = \sigma(\mathscr{E} + \mathscr{U} \times \mathscr{B}). \tag{1.11}$$

Here, $\mathscr{U} \times \mathscr{B}$ represents the electric field induced by the fluid flow.

If we further assume that phenomena involving high frequency are not considered, the displacement current $\frac{\partial \mathscr{D}}{\partial t}$ in (1.1c) can be neglected, leading to

$$\mathscr{J} = \nabla \times \mathscr{H}. \tag{1.12}$$

Combining (1.11)–(1.12), we solve for $\mathscr{E}$

$$\begin{aligned}\mathscr{E} &= \frac{1}{\sigma}\mathscr{J} - \mathscr{U} \times \mathscr{B} \\ &= \frac{1}{\sigma}\nabla \times \mathscr{H} - \mathscr{U} \times \mathscr{B} \\ &= \frac{1}{\sigma\mu}\nabla \times \mathscr{B} - \mathscr{U} \times \mathscr{B}.\end{aligned} \tag{1.13}$$

We substitute (1.13) into (1.1a) and obtain

$$\frac{\partial \mathscr{B}}{\partial t} + \frac{1}{\sigma\mu}\nabla \times \nabla \times \mathscr{B} - \nabla \times (\mathscr{U} \times \mathscr{B}) = \mathbf{0},$$

where the term $\nabla \times (\mathscr{U} \times \mathscr{B})$ counts for the electromotive force.

On the other hand, the Lorentz's force $\mathbf{f}_L$ is the body force exerted by the electromagnetic effects on the fluid. Based on (1.10) and (1.12), it is given by

$$\mathbf{f}_L = \frac{1}{\rho}\mathscr{J} \times \mathscr{B} = \frac{1}{\rho\mu}(\nabla \times \mathscr{B}) \times \mathscr{B}. \tag{1.14}$$

Substituting (1.14) into (1.9a), we obtain

$$\frac{\partial \mathscr{U}}{\partial t} - \nu_v \Delta \mathscr{U} + (\mathscr{U} \cdot \nabla)\mathscr{U} - \frac{1}{\rho}\nabla \bar{\mathscr{P}} = \frac{1}{\rho\mu}(\nabla \times \mathscr{B}) \times \mathscr{B} + \mathbf{f}.$$

It is convenient to rewrite the MHD equations in non-dimensional form. For this purpose, we introduce characteristic values: the characteristic magnetic induction B, characteristic velocity U and characteristic length L. The resulting system of the incompressible MHD equations is cast in terms of the non-dimensional space variable $\frac{\mathbf{x}}{L}$, time variable $\frac{tU}{L}$ (again denoted by $\mathbf{x}$ and t), as well as the non-dimensional fields $\mathbf{u} = \frac{\mathscr{U}}{U}$, $\mathbf{b} = \frac{\mathscr{B}}{B}$, $p = \frac{\bar{\mathscr{P}}}{\rho U^2}$ and $\mathbf{f} = \frac{\mathbf{f}L}{U^2}$. The resulting nondimensionalized system of equations is

$$
\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} - \nu\, \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \kappa\,(\nabla \times \mathbf{b}) \times \mathbf{b} &= \mathbf{f}, \\
\frac{\partial \mathbf{b}}{\partial t} + \nu_m \nabla \times (\nabla \times \mathbf{b}) - \nabla \times (\mathbf{u} \times \mathbf{b}) &= \mathbf{0}, \\
\nabla \cdot \mathbf{u} &= 0, \\
\nabla \cdot \mathbf{b} &= 0,
\end{aligned}
$$

where $\nu = \frac{1}{\mathrm{Re}}$, $\nu_m = \frac{1}{\mathrm{Rm}}$, and Re, Rm and κ are three non-dimensional numbers. The parameter $\mathrm{Re} = \frac{UL}{\nu_v}$ is the hydrodynamic Reynolds number. The parameter $\mathrm{Rm} = \sigma\mu UL$ is the magnetic Reynolds number. The parameter $\kappa = \frac{B^2}{\mu\rho U^2}$ is the coupling number. For further discussion of these parameters and their typical values, we refer the reader to [3, 42, 84].

In this thesis, we are interested in the steady-state MHD equations given by

$$
\begin{aligned}
-\nu\, \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p - \kappa\,(\nabla \times \mathbf{b}) \times \mathbf{b} &= \mathbf{f}, && \text{(1.15a)} \\
\nu_m \nabla \times (\nabla \times \mathbf{b}) - \nabla \times (\mathbf{u} \times \mathbf{b}) &= \mathbf{0}, && \text{(1.15b)} \\
\nabla \cdot \mathbf{u} &= 0, && \text{(1.15c)} \\
\nabla \cdot \mathbf{b} &= 0. && \text{(1.15d)}
\end{aligned}
$$

The PDE system (1.15) is the focus of our interest in Chapters 3 and 4. We will transform the system into mixed form, making it amenable to discretized curl-confirming elements.

1.4 Thesis overview and contributions

This thesis is organized in five chapters. In Chapter 2, we discuss the numerical solution of the Maxwell equations in (1.8) and present a fully scalable parallel iterative solver. It is based on a mixed finite element discretization on tetrahedral meshes, using the lowest order Nédélec elements of the first kind [76] for the approximation of the vector field, and standard nodal elements for the Lagrange multiplier associated with the divergence constraint. The corresponding linear system has a saddle point form, and is solved using preconditioned MINRES [79]. We adopt and extend the block diagonal preconditioning approach derived in [43], with a nodal auxiliary space preconditioning technique as an inner iteration for the shifted curl-curl operator [56].

Our main contribution is the development and parallel implementation of this approach for large-scale problems in complex three-dimensional domains with variable coefficients. We demonstrate the performance of our parallel solver on problems with up to approximately 40 million degrees of freedom. Our numerical results indicate very good scalability with the mesh size, on uniform, unstructured and locally refined meshes. A preliminary version of this work has been published in the conference proceedings of the second international conference on High Performance Computing and Applications (Shanghai, China, 2009) in *Lecture Notes in Computer Science on High Performance Computing and Applications* [69]. A complete version has been submitted to *Numerical Linear Algebra with Applications* in June, 2010 [70].

In Chapter 3, we move on to consider the more complex incompressible MHD problem (1.15), of which the Maxwell equations discussed in Chapter 2 are a subproblem. We propose a new finite element method for the numerical discretization of a stationary incompressible MHD problem in two and three dimensions. The novelty of the method is two-fold: first, the method produces exactly divergence-free velocity approximations and the resulting discretization is provably energy-stable; secondly, it captures the strongest magnetic singularities. The velocity field is discretized using divergence-conforming Brezzi-Douglas-Marini (BDM) elements [15] and the magnetic field is approximated by curl-conforming Nédélec elements. The H^1-continuity of the velocity field is enforced by a discontinuous Galerkin (DG) approach. The energy norm of the error is convergent in the mesh size in general possibly non-convex Lipschitz polyhedra under minimal regularity assumptions, and we have nearly optimal a-priori error estimates for the two-dimensional case, and slightly suboptimal ones for the three-dimensional case. We present a comprehensive set of numerical experiments, which indicate optimal convergence of the proposed method for two-dimensional as well as three-dimensional problems. We numerically confirm that the strongest magnetic singularities can be correctly captured. This work has been published in *Computer Methods in Applied*

Mechanics and Engineering [44].

In Chapter 4, we focus on the specific issue of developing efficient solvers for the linear system arising from our finite element discretization, introduced in Chapter 3. Here, the main difficulty is the presence of possibly strong skew-symmetric coupling terms. We present preliminary results for preconditioning techniques applied to the discretized MHD equations. We propose a preconditioning approach based on combining efficient preconditioners of the Maxwell sub-system (discussed in Chapter 2) and Navier-Stokes sub-system [37, Chapter 8]. We provide some preliminary analysis for our approach. We present numerical results to show that our approach performs reasonably well. We did not try to parallelize the solver for the MHD problems. This is left as an interesting topic for future work.

Conclusions and an outline of related future work are provided in Chapter 5.

Chapter 2

Solution of the discretized time-harmonic Maxwell equations in mixed form

In this chapter we develop a fully scalable parallel solver for the time-harmonic Maxwell equations in complicated domains in three dimensions. The preconditioned iterations are based on outer iterations of the form introduced in [43], extended to the variable coefficient case, and inner iterations of the form derived in [56]. We use algebraic multigrid solvers for each elliptic problem, and accomplish almost linear complexity in the number of degrees of freedom. With our implementation we can solve problems of dimensions of up to approximately 40 million degrees of freedom. Our numerical results scale very well with the mesh size, on uniform, unstructured, and locally refined meshes.

This chapter is structured as follows. First, we discuss related work in Section 2.1. In Section 2.2 we present the mixed finite element discretization and analyze the properties of the discrete operators. The preconditioning approach is presented in Section 2.3. In Section 2.4 we provide numerical examples to demonstrate the scalability and performance of the proposed solvers.

2.1 Background and related work

To find the solution to (1.8), we need to know the boundary conditions associated with the domain Ω. We assume $\partial\Omega$ is a perfectly conducting connected boundary of Ω, and have the following boundary condition:

$$\mathbf{n} \times \mathbf{E} = 0 \quad \text{on} \quad \partial\Omega,$$

where $\mathbf{n}$ denotes the outward normal unit vector to $\partial\Omega$.

We are interested in numerically solving (1.8) using mixed finite element methods. Mixed formulations of the Maxwell equations have been used in [17, 80, 94]. Such formulations may improve the stability for vanishing wave numbers [94] and yield a well defined problem. A mixed approach has been used in [17] for treating non-matching meshes. To derive a mixed formulation we apply the Helmholtz decomposition [80], and write

$$\mathbf{E} = \mathbf{u} + \nabla \hat{p},$$

where $\varepsilon_r\mathbf{u}$ is divergence-free, and $\hat{p}$ has homogeneous Dirichlet boundary conditions. Setting $p = -k^2\hat{p}$, we obtain the following mixed problem: find the vector field $\mathbf{u}$ and the scalar multiplier p such that

$$\begin{aligned}
\nabla \times \mu_r^{-1} \nabla \times \mathbf{u} - k^2 \varepsilon_r \mathbf{u} + \varepsilon_r \nabla p &= \mathbf{f} \quad \text{in} \quad \Omega, && (2.1a)\\
\nabla \cdot (\varepsilon_r \mathbf{u}) &= 0 \quad \text{in} \quad \Omega, && (2.1b)\\
\mathbf{n} \times \mathbf{u} &= 0 \quad \text{on} \quad \partial\Omega, && (2.1c)\\
p &= 0 \quad \text{on} \quad \partial\Omega. && (2.1d)
\end{aligned}$$

We assume that

$$0 < \mu_{\min} \leq \mu_r \leq \mu_{\max} < \infty \qquad \text{and} \qquad 0 < \varepsilon_{\min} \leq \varepsilon_r \leq \varepsilon_{\max} < \infty.$$

The wave number k is assumed small,

$$k \ll 1.$$

In the two-dimensional case, the curl operator $\nabla\times$ applied to a vector $\mathbf{u} = (u_1, u_2)$ is defined as $\nabla \times \mathbf{u} = \frac{\partial u_2}{\partial x} - \frac{\partial u_1}{\partial y}$, while the curl of a scalar function p is determined by $\nabla \times p = (\frac{\partial p}{\partial y}, -\frac{\partial p}{\partial x})$. Similarly, the cross product of two vectors $\mathbf{u} = (u_1, u_2)$ and $\mathbf{v} = (v_1, v_2)$ is given by $\mathbf{u} \times \mathbf{v} = u_1 v_2 - u_2 v_1$.

Finite element discretization using Nédélec elements of the first kind [76] for the approximation of $\mathbf{u}$ and standard nodal elements for p yields a saddle point linear system of the form

$$\mathscr{K} x \equiv \begin{pmatrix} A - k^2 M & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \equiv b, \tag{2.2}$$

whose size is $(m+n) \times (m+n)$. The matrix $A \in \mathbb{R}^{n\times n}$ corresponds to the μ_r^{-1}-weighted discrete curl-curl operator; $B \in \mathbb{R}^{m\times n}$ is the ε_r-weighted divergence operator with full row rank; $M \in \mathbb{R}^{n\times n}$ is the ε_r-weighted vector mass matrix; $f \in \mathbb{R}^n$ is now the load vector associated with the right-hand side in (2.1a), and the vectors $u \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ represent the finite element coefficients. Note that A is symmetric positive semidefinite with nullity m.

In [43], block diagonal preconditioners were designed for iteratively solving system (2.2) with constant coefficients. These preconditioners were motivated by spectral equivalence properties. Each iteration of the scheme requires inverting

$$A + \gamma M,$$

where $\gamma > 0$ is a given parameter. There exist several effective multigrid methods for doing so. When a hierarchy of structured meshes is available, geometric multigrid can be applied [54]; for unstructured meshes, algebraic multigrid (AMG) approaches have been explored and implemented in [13, 39, 61, 83], using the smoothers introduced in [54]. A different approach has been proposed in [56]: a nodal auxiliary space preconditioner, which reduces the problem into solving two scalar elliptic problems on the nodal finite element space. A parallel solver based on the nodal auxiliary space preconditioners is implemented in [38, 67].

In this chapter we develop a fully scalable parallel implementation for solving (2.2) in complicated domains in three dimensions. The preconditioned iterations are based on outer iterations of the form introduced in [43], extended to the variable coefficient case, and inner iterations of the form derived in [56]. We use algebraic multigrid solvers for each elliptic problem, and accomplish almost linear complexity in the number of degrees of freedom. With our implementation we can solve problems of dimensions of up to approximately 40 million degrees of freedom. Our numerical results scale very well with the mesh size, on uniform, unstructured, and locally refined meshes.

2.2 Finite element discretization

In this section, we present a mixed finite element method that employs curl-conforming elements. Then, we discuss properties of the discrete operators.

2.2.1 Variational formulation

We assume Ω is a bounded simply-connected Lipschitz polytope in $\mathbb{R}^d$ ($d = 2$ or 3), with a connected boundary $\partial\Omega$. To write (2.2) in weak form, we denote by $(\cdot,\cdot)_\Omega$ the inner product in $L^2(\Omega)$ or $L^2(\Omega)^d$. For the computational domain $\Omega \subset \mathbb{R}^d$, we let

$$\begin{aligned} \mathbf{C} &= H_0(\text{curl};\Omega) = \{\mathbf{u} \in L^2(\Omega)^d : \nabla \times \mathbf{u} \in L^2(\Omega)^d, \mathbf{n} \times \mathbf{u} = 0 \quad \text{on} \quad \partial\Omega\}, \\ S &= H_0^1(\Omega) = \{p \in H^1(\Omega) : p = 0 \quad \text{on} \quad \partial\Omega\}. \end{aligned} \tag{2.3}$$

The variational formulation of the Maxwell equations (2.1) is to find $(\mathbf{u}, p) \in \mathbf{C} \times S$ such that

$$A(\mathbf{u},\mathbf{v}) - k^2 M(\mathbf{u},\mathbf{v}) + B(\mathbf{v},p) = (\mathbf{f},\mathbf{v})_\Omega, \tag{2.4a}$$

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

for all $(\mathbf{v}, q) \in \mathbf{C} \times S$. The variational forms are given by

$$\begin{aligned} A(\mathbf{u},\mathbf{v}) &= \int_\Omega \mu_r^{-1} (\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v})\, d\mathbf{x}, \\ M(\mathbf{u},\mathbf{v}) &= \int_\Omega \varepsilon_r \mathbf{u} \cdot \mathbf{v}\, d\mathbf{x}, \\ B(\mathbf{u},q) &= \int_\Omega \varepsilon_r \mathbf{u} \cdot \nabla q\, d\mathbf{x}. \end{aligned}$$

Recall that the wave number k is given by $k = \omega\sqrt{\varepsilon_0\mu_0}$ and $\omega \neq 0$ is the angular frequency. We assume throughout that $k^2\varepsilon_r$ is not a Maxwell eigenvalue. The well-posedness of (2.1) has been proved in [32, 80]. Note that (2.1) is well-posed for $k = 0$. In this case, if $\varepsilon_r \equiv 1$, this corresponds to the mixed formulation of the magnetostatic problem in terms of the vector potential $\mathbf{u}$, along with Coulomb's gauge $\nabla \cdot \mathbf{u} = 0$. In our numerical tests, we will also consider $k = 0$ and variable ε_r.

2.2.2 Mixed discretization

We consider a family of regular and quasi-uniform triangulations $\mathscr{T}_h$ of a sufficiently small mesh size h that partition the domain Ω into simplices $\{K\}$ (i.e., triangles for $d = 2$ and tetrahedra for $d = 3$).

For $k \geq 1$, we wish to approximate the solution of (2.1) by finite element functions $(\mathbf{u}_h, p_h) \in \mathbf{C}_h \times S_h$, where

$$\begin{aligned} \mathbf{C}_h &= \{\, \mathbf{u} \in H_0(\mathrm{curl};\Omega) \,:\, \mathbf{u}|_K \in \mathscr{P}_{k-1}(K)^d \oplus R_k(K), K \in \mathscr{T}_h \,\}, \\ S_h &= \{\, p \in H_0^1(\Omega) \,:\, p|_K \in \mathscr{P}_k(K), K \in \mathscr{T}_h \,\}. \end{aligned} \tag{2.5}$$

Here, we denote by $\mathscr{P}_k(K)$ the space of polynomials of total degree at most k on element K, and by $R_k(K)$ the space of homogeneous vector polynomials of total degree k that are orthogonal to the position vector $\mathbf{x}$.

The space $\mathbf{C}_h$ represents the first family of curl-conforming Nédélec elements (cf. [76, Chapter 5]); its degrees of freedom are defined for the tangential components of functions along faces. Figure 2.1 shows the degrees of freedom on the lowest order Nédélec elements in 2D and 3D.

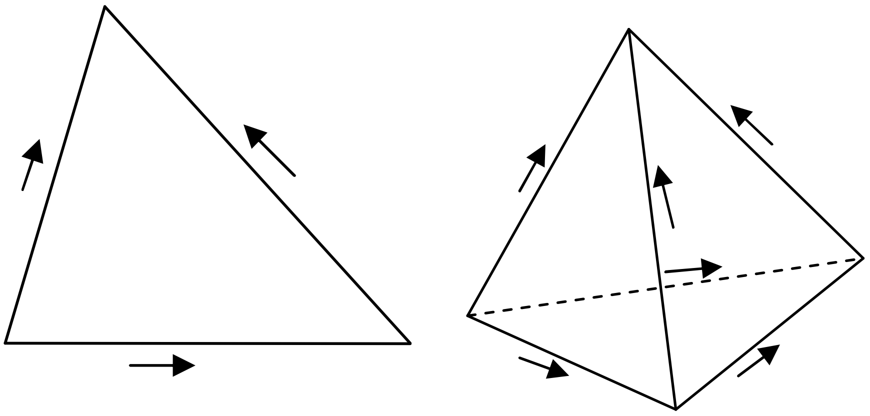
Now we consider the following finite element method: find $(\mathbf{u}_h, p_h) \in \mathbf{C}_h \times S_h$ such that

$$A(\mathbf{u}_h, \mathbf{v}) - k^2 M(\mathbf{u}_h, \mathbf{v}) + B(\mathbf{v}, p_h) = (\mathbf{f}, \mathbf{v})_\Omega, \tag{2.6a}$$

$$B(\mathbf{u}_h, q) = 0, \tag{2.6b}$$

for all $(\mathbf{v}, q) \in \mathbf{C}_h \times S_h$.

Let $\langle \psi_j \rangle_{j=1}^n$ and $\langle \phi_i \rangle_{i=1}^m$ be finite element bases for the spaces $\mathbf{C}_h$ and S_h re-



(a) 2D

(b) 3D

Figure 2.1: A graphical illustration of the degrees of freedom for the lowest order Nédélec element in 2D and 3D. Degrees of freedom are average values of tangential component of vector fields on each edge.

spectively:

$$\mathbf{C}_h = \mathrm{span}\langle \psi_j \rangle_{j=1}^{n}, \qquad S_h = \mathrm{span}\langle \phi_i \rangle_{i=1}^{m}. \tag{2.7}$$

Then, the weak formulation in (2.4) yields a linear system of the form (2.2), where the entries of the matrices and the load vector are given by

$$\begin{aligned}
A_{i,j} &= \int_\Omega \mu_r^{-1} \left(\nabla \times \psi_j\right) \cdot \left(\nabla \times \psi_i\right) d\mathbf{x}, && 1 \le i, j \le n, \\
M_{i,j} &= \int_\Omega \varepsilon_r \psi_j \cdot \psi_i \, d\mathbf{x}, && 1 \le i, j \le n, \\
B_{i,j} &= \int_\Omega \varepsilon_r \psi_j \cdot \nabla \phi_i \, d\mathbf{x}, && 1 \le i \le m, \ 1 \le j \le n, \\
f_i &= \int_\Omega \mathbf{f} \cdot \psi_i \, d\mathbf{x}, && 1 \le i \le n.
\end{aligned}$$

The mixed finite element discretization (2.6) has a unique solution; see the proof in [32].

Let us introduce a few additional matrices that play an important role in this formulation. First, note that $\nabla S_h \subset \mathbf{C}_h$, and define the matrix $C \in \mathbb{R}^{n\times m}$ by

$$\nabla\phi_j = \sum_{i=1}^{n} C_{i,j}\psi_i, \qquad j = 1,\dots,m. \tag{2.8}$$

For a function $q_h \in S_h$ given by $q_h = \sum_{j=1}^{m} q_j\phi_j$, we then have

$$\nabla q_h = \sum_{i=1}^{n}\sum_{j=1}^{m} C_{i,j} q_j \psi_i,$$

so Cq is the coefficient vector of ∇q_h in the basis $\langle\psi_i\rangle_{i=1}^{n}$. In the lowest order case, the entries of C are

$$C_{i,j} = \begin{cases} 1 & \text{if node } j \text{ is the head of edge } i, \\ -1 & \text{if node } j \text{ is the tail of edge } i, \\ 0 & \text{otherwise.} \end{cases}$$

Define the ε_r-weighted scalar Laplacian on S_h as $L = (L_{i,j})_{i,j=1}^{m} \in \mathbb{R}^{m\times m}$ with

$$L_{i,j} = \int_\Omega \varepsilon_r \nabla\phi_j \cdot \nabla\phi_i \, d\mathbf{x}. \tag{2.9}$$

Finally, we set $Q = (Q_{i,j})_{i,j=1}^{m} \in \mathbb{R}^{m\times m}$ as the ε_r-weighted scalar mass matrix on S_h, that is

$$Q_{i,j} = \int_\Omega \varepsilon_r \phi_j \cdot \phi_i \, d\mathbf{x}.$$

2.2.3 Properties of the discrete operators

Let us state a few stability results that extend the analysis in [43] from constant material coefficients to the variable coefficient case.

Denote by $\langle\cdot,\cdot\rangle$ the standard Euclidean inner product in $\mathbb{R}^n$ or $\mathbb{R}^m$, and by $\mathrm{null}(\cdot)$ the null space of a matrix. For a given positive (semi)definite matrix W and a vector x, we define the (semi)norm

$$|x|_W = \sqrt{\langle Wx, x\rangle}.$$

Proposition 1 *The following stability properties of the matrices A and B hold:*

(i) Continuity of A:

$$|\langle Au, v\rangle| \leq |u|_A |v|_A, \qquad u, v \in \mathbb{R}^n.$$

(ii) Continuity of B:

$$|\langle Bv, q\rangle| \leq |v|_M |q|_L, \qquad v \in \mathbb{R}^n,\ q \in \mathbb{R}^m.$$

(iii) The matrix A is positive definite on null(B) *and*

$$\langle Au, u\rangle \geq \alpha \left(|u|_A^2 + |u|_M^2\right), \qquad u \in \mathrm{null}(B),$$

with a stability constant α which is independent of the mesh size.

(iv) The matrix B satisfies the discrete inf-sup condition

$$\inf_{0 \neq q \in \mathbb{R}^m} \sup_{0 \neq v \in \mathrm{null}(A)} \frac{\langle Bv, q\rangle}{|v|_M |q|_L} \geq 1.$$

Proof The first two properties follow directly from the Cauchy-Schwarz inequality.

To show (iii), we first recall the discrete Poincaré–Friedrichs inequality from [55, Theorem 4.7]. Let $u \in \mathrm{null}(B)$ and let u_h be the associated finite element function. Then, we have

$$\int_\Omega |\nabla \times u_h|^2 \, d\mathbf{x} \geq \beta \int_\Omega |u_h|^2 \, d\mathbf{x},$$

where $\beta > 0$ is independent of the mesh size.

Consequently, we bound $\langle Au,u\rangle$ as follows

$$\begin{aligned}
\langle Au,u\rangle &= \frac{1}{2}\langle Au,u\rangle + \frac{1}{2}\langle Au,u\rangle \\
&= \frac{1}{2}|u|_A^2 + \frac{1}{2}\int_\Omega \mu_r^{-1}|\nabla\times u_h|^2\,d\mathbf{x} \\
&\geq \frac{1}{2}|u|_A^2 + \frac{1}{2\mu_{\max}}\int_\Omega |\nabla\times u_h|^2\,d\mathbf{x} \\
&\geq \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}}\int_\Omega |u_h|^2\,d\mathbf{x} \\
&\geq \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}\varepsilon_{\max}}\int_\Omega \varepsilon_r|u_h|^2\,d\mathbf{x} \\
&= \frac{1}{2}|u|_A^2 + \frac{\beta}{2\mu_{\max}\varepsilon_{\max}}|u|_M^2, \\
&\geq \min\left(\frac{1}{2},\frac{\beta}{2\mu_{\max}\varepsilon_{\max}}\right)(|u|_A^2+|u|_M^2) = \alpha(|u|_A^2+|u|_M^2),
\end{aligned}$$

where $\alpha = \min\left(\frac{1}{2},\frac{\beta}{2\mu_{\max}\varepsilon_{\max}}\right)$. Note that, since $\langle Au,u\rangle = |u|_A^2$, we must have $0 < \alpha < 1$, and then also

$$|u|_A^2 \geq \bar{\alpha}|u|_M^2, \qquad u\in \operatorname{null}(B), \tag{2.10}$$

with

$$\bar{\alpha} = \frac{\alpha}{1-\alpha}. \tag{2.11}$$

To prove (iv), let $0\neq q_h\in S_h$ and v be the coefficient vector of $\mathbf{v}_h = \nabla q_h$ in the basis $\langle\psi_i\rangle_{i=1}^n$. Then it follows that $v\in\operatorname{null}(A)$ and

$$\begin{aligned}
\sup_{0\neq v\in\operatorname{null}(A)} \frac{\langle Bv,q\rangle}{|v|_M|q|_L} &= \sup_{0\neq v\in\operatorname{null}(A)} \frac{\int_\Omega \varepsilon_r \mathbf{v}_h\cdot\nabla q_h\,d\mathbf{x}}{\left(\int_\Omega \varepsilon_r\mathbf{v}_h\cdot\mathbf{v}_h d\mathbf{x}\right)^{\frac{1}{2}}|q|_L} \\
&\geq \frac{\int_\Omega \varepsilon_r\nabla q_h\cdot\nabla q_h\,d\mathbf{x}}{\left(\int_\Omega \varepsilon_r\nabla q_h\cdot\nabla q_h d\mathbf{x}\right)^{\frac{1}{2}}|q|_L} = \frac{|q|_L^2}{|q|_L^2} = 1,
\end{aligned}$$

which shows (iv). ▮

The properties stated in Proposition 1 and the theory of mixed finite element methods [15, Chapter 2] ensure that the saddle point system (2.2) is invertible (provided that the mesh size is sufficiently small).

2.3 The solver

To iteratively solve the saddle point system (2.2) we use MINRES as an *outer solver*. This is discussed in Section 2.3.1. To perform each outer iteration, we apply an *inner solver* based on [56] and presented in Section 2.3.2. In Section 2.3.3, we outline the complete solution procedure, and provide details on our scalable parallel implementation in Section 2.3.4.

2.3.1 The outer solver

Following the analysis for constant coefficients in [43], we propose the following block diagonal preconditioner to iteratively solve (2.2):

$$\mathscr{P}_{M,L} = \begin{pmatrix} \mathscr{P}_M & 0 \\ 0 & L \end{pmatrix}, \tag{2.12}$$

where

$$\mathscr{P}_M = A + \gamma M \tag{2.13}$$

and

$$\gamma = 1 - k^2. \tag{2.14}$$

We have the following result:

Theorem 2 *The preconditioned matrix $\mathscr{P}_{M,L}^{-1}\mathscr{K}$ has two eigenvalues $\lambda_+ = 1$ and $\lambda_- = -\frac{1}{1-k^2}$, each with algebraic multiplicity m. The remaining eigenvalues satisfy the bound*

$$\frac{\bar{\alpha} - k^2}{\bar{\alpha} + 1 - k^2} < \lambda < 1, \tag{2.15}$$

where $\bar{\alpha}$ is the constant in (2.11).

Proof The proof follows along the lines of [43, Theorem 5.2]. First, we verify that the following relations hold:

(1) $\mathbb{R}^n = \text{null}(A) \oplus \text{null}(B)$.

(2) For any $u \in \operatorname{null}(A)$, there is a unique $q \in \mathbb{R}^m$ such that $u = Cq$.

(3) $BC = L$.

(4) $MC = B^T$.

(5) $\langle Mu, Cq\rangle = \langle Bu, q\rangle$ for $u \in \mathbb{R}^n$ and $q \in \mathbb{R}^m$.

(6) $\langle MCp, Cq\rangle = \langle Lp, q\rangle$ for $p, q \in \mathbb{R}^m$.

(7) Let $u \in \operatorname{null}(A)$ with $u = Cp$. Then $|u|_M = |p|_L$.

(8) For any $u \in \operatorname{null}(A)$, we have $\langle B^T L^{-1} Bu, u\rangle = |u|_M$.

The relations (1) and (2) follow from the discrete Helmholtz decomposition [73, Section 7.2.1].

To show (3), we have, for $1 \le i, j \le m$,

$$(BC)_{i,j} = \sum_{k=1}^{n} B_{i,k} C_{k,j} = \int_\Omega \left(\sum_{k=1}^{n} C_{k,j} \varepsilon_r \psi_k \right) \cdot \nabla\phi_i \, d\mathbf{x} = \int_\Omega \varepsilon_r \nabla\phi_j \cdot \nabla\phi_i \, d\mathbf{x} = L_{i,j}.$$

Relation (4) follows similarly. Relation (5) follows from relation (4).

To show (6), if ∇p_h and ∇q_h are the finite element functions associated with the vectors Cp and Cq, then we have

$$\langle MCp, Cq\rangle = \int_\Omega \varepsilon_r \nabla p_h \cdot \nabla q_h \, d\mathbf{x} = \langle Lp, q\rangle.$$

Relation (7) follows from (6).

Finally, to see (8), from relation (2), we have $u = Cp$ for a vector $p \in \mathbb{R}^m$. Using relations (3) and (7), we obtain

$$\langle B^T L^{-1} Bu, u\rangle = \langle L^{-1} Bu, Bu\rangle = \langle L^{-1} BCp, BCp\rangle = \langle Lp, p\rangle = |p|_L = |u|_M.$$

An orthogonality property with respect to the inner product $\langle M\cdot, \cdot\rangle$ is obtained as follows. Let $u_A \in \operatorname{null}(A)$ and $u_B \in \operatorname{null}(B)$. Setting $u_A = Cq$, we have

$$\langle Mu_A, u_B\rangle = \langle Mu_B, Cq\rangle = \langle Bu_B, q\rangle = 0, \tag{2.16}$$

by relation (5).

Second, the eigenvalue problem for $\mathscr{P}_{M,L}^{-1}\mathscr{K}$ now is

$$\begin{pmatrix} A-k^2M & B^T \\ B & 0 \end{pmatrix}\begin{pmatrix} v \\ q \end{pmatrix} = \lambda \begin{pmatrix} A+(1-k^2)M & 0 \\ 0 & L \end{pmatrix}\begin{pmatrix} v \\ q \end{pmatrix}.$$

Setting $q = \frac{1}{\lambda}L^{-1}Bv$ and multiplying the resulting equation for v by λ, we have

$$\left[(\lambda^2-\lambda)A+((1-k^2)\lambda^2+k^2\lambda)M\right]v = B^TL^{-1}Bv.$$

Suppose $v = v_A + v_B$, where $v_A \in \text{null}(A)$ and $v_B \in \text{null}(B)$, by relation (1). We then have

$$(\lambda^2-\lambda)Av_B + ((1-k^2)\lambda^2+k^2\lambda)M(v_A+v_B) = B^TL^{-1}Bv_A.$$

There are at least m linearly independent vectors v that satisfy $v_A \neq 0$. For m such vectors, taking the inner product with v_A and noting that by relation (8) $\langle B^TL^{-1}Bu_A, u_A\rangle = |u|_M^2$ and that by (2.16) we have $\langle M(v_A+v_B), v_A\rangle = \langle Mv_A, v_A\rangle = |v_A|_M^2$, we get

$$(1-k^2)\lambda^2|v_A|_M^2 + k^2\lambda|v_A|_M^2 = |v_A|_M^2,$$

from which λ_+ and λ_- are obtained as solutions of the quadratic equation

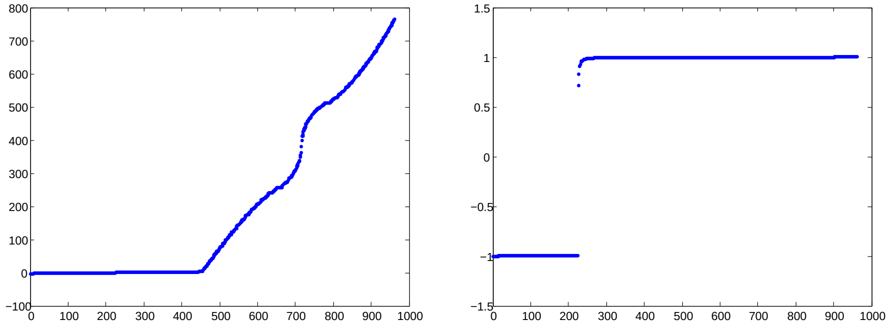
$$(1-k^2)\lambda^2 + k^2\lambda = 1.$$

For the remaining eigenvectors we must have $v_B \neq 0$. Then, we take the inner product with v_B and use that $\langle B^TL^{-1}Bv_A, v_B\rangle = \langle L^{-1}Bv_A, Bv_B\rangle = 0$ and by (2.16) $\langle M(v_A+v_B), v_B\rangle = \langle Mv_B, v_B\rangle = |v_B|_M^2$. It follows that

$$(\lambda-\lambda^2)|v_B|_A^2 = \left((1-k^2)\lambda^2+k^2\lambda\right)|v_B|_M^2.$$

Recall that we assume $k \ll 1$ and $\lambda \neq 0$. Therefore, we must have $\lambda < 1$. Applying (2.10), we obtain (2.15). ∎

In Figure 2.2, we show the eigenvalues of $\mathscr{K}$ and $\mathscr{P}_{M,L}^{-1}\mathscr{K}$. We use the lowest order Nédélec elements on $\Omega = (-1,1)^2$ and compute the eigenvalues on a uniform triangular mesh of moderate size, for $k=0$. Figure 2.2 illustrates the effect of preconditioning, as described in Theorem 2: the preconditioned eigenvalues are strongly clustered, whereas the original ones become unbounded with the mesh size.



(a) (b)

Figure 2.2: Eigenvalues of (a) the matrix $\mathscr{K}$ and (b) the preconditioned matrix $\mathscr{P}_{M,L}^{-1}\mathscr{K}$, for $k=0$. For the non-preconditioned matrix $\mathscr{K}$, the $m=225$ negative eigenvalues range in value from -2.80 to -0.20 and the remaining $n=736$ positive eigenvalues range from 0.19 to 764.89. On the other hand, most of the eigenvalues of the preconditioned matrix $\mathscr{P}_{M,L}^{-1}\mathscr{K}$ are extremely close or equal to either -1 or 1.

2.3.2 The inner solver

The overall computational cost of using $\mathscr{P}_{M,L}$ depends on the ability to efficiently solve linear systems whose associated matrices are $\mathscr{P}_M$ in (2.13) and L in (2.9).

The linear system L arises from a standard scalar elliptic problem, for which many efficient solution methods exist. On the other hand, efficiently inverting $\mathscr{P}_M$ is the computational bottleneck in the inner iteration. Recently, Hiptmair and Xu proposed effective auxiliary space preconditioners for linear systems arising from conforming finite element discretizations of $H(\text{curl})$-elliptic variational problems [56], based on fictitious spaces as developed in [45, 96]. The preconditioner is:

$$\mathscr{P}_V^{-1} = \text{diag}(\mathscr{P}_M)^{-1} + P(\bar{L}+\gamma\bar{Q})^{-1}P^T + \gamma^{-1}C(L^{-1})C^T, \qquad (2.17)$$

with γ as in (2.14). The matrix $\bar{L} = \operatorname{diag}(L, L, L)$ is the ε_r-weighted vector Laplacian on S_h^3, $\bar{Q} = \operatorname{diag}(Q, Q, Q)$ is the ε_r-weighted vector mass matrix on S_h^3, C is the null-space matrix in (2.8), and P is the matrix representation of the nodal interpolation operator $\Pi_h^{\text{curl}} : S_h^3 \to \mathbf{C}_h$. In the lowest order case, the operator Π_h^{curl} is based on path integrals along edges; for a finite element function $\mathbf{w}_h \in S_h^3$ it is given by

$$\Pi_h^{\text{curl}} \mathbf{w}_h = \sum_j \left(\int_{e_j} \mathbf{w}_h \cdot d\mathbf{s} \right) \psi_j,$$

where e_j is the interior edge associated with the basis function ψ_j. We have $P = [P^{(1)}, P^{(2)}, P^{(3)}]$, where $P \in \mathbb{R}^{n \times 3m}$ and $P^{(k)}$ $(k = 1, 2, 3)$ are matrices in $\mathbb{R}^{n \times m}$. The entries of $P^{(k)}$ are given by

$$P_{i,j}^{(k)} = \begin{cases} 0.5\, d_i\, t_i^{(k)} & \text{if node } j \text{ is the head/tail of edge } i, \\ 0 & \text{otherwise,} \end{cases}$$

where $t_i^{(k)}$ is the k-th component of the unit tangential vector on edge i, and d_i is the length of edge i.

In the constant coefficient case, it was shown in [56, Theorem 7.1] that for $0 < \gamma \leq 1$, the spectral condition number $\kappa_2(\mathscr{P}_V^{-1} \mathscr{P}_M)$ is independent of the mesh size. Even though there seems to be no theoretical analysis available for the variable coefficient case, the preconditioner $\mathscr{P}_V$ was experimentally shown to be effective in this case as well [56].

2.3.3 Solution algorithm

We run preconditioned MINRES as the outer solver for the linear system (2.2). The preconditioner is the block diagonal matrix $\mathscr{P}_{M,L}$, defined in (2.12). For each outer iteration, we need to solve a linear system of the form

$$\begin{pmatrix} \mathscr{P}_M & 0 \\ 0 & L \end{pmatrix} \begin{pmatrix} v \\ q \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}. \tag{2.18}$$

Two Krylov subspace solvers are applied as the inner iterations. The linear system associated with the $(1,1)$ block,

$$\mathscr{P}_M v = c, \tag{2.19}$$

is solved using CG with the preconditioner $\mathscr{P}_V$, which is defined in (2.17). In each CG iteration, we need to solve a linear system of the form

$$\mathscr{P}_V w = r. \tag{2.20}$$

Following (2.17), this can be done by solving the two linear systems

$$\begin{aligned} (\bar{L}+\gamma\bar{Q})y &= s, && (2.21a)\\ \bar{L}z &= t, && (2.21b) \end{aligned}$$

where $s = P^T r$ and $t = C^T r$. We run one AMG V-cycle to compute y and z, and we set

$$w = \operatorname{diag}(\mathscr{P}_M)^{-1} r + Py + \gamma^{-1} Cz. \tag{2.22}$$

The linear system associated with the $(2,2)$ block of (2.18),

$$Lq = d, \tag{2.23}$$

is solved using CG with an AMG preconditioner.

Our approach is summarized in Algorithm 1. The inner iteration for (2.19) is initialized in line 4 and laid out in lines 5–10, where CG iterations preconditioned with $\mathscr{P}_V$ are used. The inner iteration for (2.23) is initialized in line 11 and provided in lines 12–15, where a CG scheme with an AMG preconditioner is used. Once the two iterative solvers converge, we update the approximated solution x for the next outer iterate in line 16. The stopping criteria are discussed in Section 2.4.

2.3.4 Scalable parallel implementation

We solve the Maxwell equations using the lowest order discretization with Nédélec elements of the first family and linear nodal elements. The degrees of freedom

Algorithm 1 Solve $\mathscr{K} x = b$ in (2.2)

```
initialize MINRES for (2.2)
while MINRES not converged do
    set $c$, $d$ to be the right-hand side for the current inner iteration; see (2.18)
    initialize CG for (2.19)
    while CG not converged do
        run one AMG V-cycle to approximate $(\bar{L} + \gamma\bar{Q})^{-1}$; update $y$ in (2.21a)
        run one AMG V-cycle to approximate $L^{-1}$; update $z$ in (2.21b)
        update $w$ in (2.20), using (2.22)
        update $v$ in (2.19)
    end while
    initialize CG for (2.23)
    while CG not converged do
        apply AMG preconditioner to approximate $L$
        update $q$ in (2.23)
    end while
    update $x$ in (2.2)
end while
```

in this discretization are associated with vertices and edges of the tetrahedra; cf. Figure 2.1. To distribute data on parallel computers, a standard domain partitioning approach is used, see, e.g., [65]. To illustrate the procedure for accomplishing load-balanced data distribution, we provide a few details on the steps that are taken:

(1) **Meshing.** We use our own mesher to generate structured meshes, or use TetGen [91] for unstructured and locally refined meshes.

(2) **Element partitioning.** Given the mesh connectivity graph, we use METIS [62] to partition the elements into non-overlapping subdomains. There are two primary metrics in judging the quality of a partition: the subdomain size and the number of edge cuts in the connectivity graph. Since our meshes are comprised of tetrahedra only, each subdomain should contain a roughly equal number of elements so that the resulting domain partition is load-balanced across all processors. The number of edge cuts indicates the interprocessor communication required by the parallel solver, which is minimized in METIS [63].

(3) **Element re-numbering.** We re-number all elements according to the rank of the processor they belong to. Elements stored on a processor with a lower rank have smaller indices. Elements stored on the same processor have contiguous indices.

(4) **Distribution of degrees of freedom.** Based on the partitioning of elements, we distribute all degrees of freedom across all processors. We do so by looping over all elements sequentially, starting from the ones stored on the processor with the lowest rank.

(5) **Re-numbering of degrees of freedom.** We re-number all degrees of freedom such that the numbering is contiguous on the processor that owns them. The reordering is done sequentially, starting with the processor with the lowest rank in the communicator.

Once a partition is obtained, we assemble all system matrices and vectors in parallel. We use PETSc [7], BoomerAMG [51] and Hypre [38] in our own preconditioned iterative solution code. The way we order the degrees of freedom and assign them to processors is consistent with the sparse matrix partitioning scheme employed in PETSc.

Example Figure 2.3 gives an example of how the mesh partitioning is done. The numbers bounded by squares are indices of elements; the circled numbers are indices of vertices; and the rest are indices of edges. Figure 2.3(a) shows the original numbering of elements, vertices, and edges as the output of a mesher. Suppose that we partition the mesh across two processors. Based on the two metrics described in step 2 and the connectivity graph provided to METIS, it outputs the partitioning. For this example we get that processor 1 owns elements 1, 2, 6, 8 and processor 2 owns elements 3–5, 7. We now as per step 3 reorder the elements. The new indices of elements 1–8 become 1, 2, 7, 6, 5, 3, 8, 4 as shown in Figure 2.3(b). Next we consider how to distribute the degrees of freedom associated with the vertices and edges across the two processors. According to step 4, we loop over all elements starting from those stored on processor 1, whose new indices are 1–4. For element 1, vertices 1, 4, 9 and edges 1, 2, 3 are associated with it. They have not been assigned to any processor previously, so they belong to processor 1. Element 2 has vertices 4, 5, 9 and edges 2, 4, 5. Vertices 4, 9 and edge 2 have been allocated to processor 1, so we only need to store vertex 5 and edges 4, 5 to processor 1. We continue to do this until we have looped over all elements. The degrees of freedom associated with the indices with underlines in Figure 2.3(c) are stored on processor 1. The remaining ones are stored on processor 2. Once the partition is obtained, we perform the reordering procedure as in step 5. We start with elements stored on processor 1 from element 1. Now the new indices of vertices 1, 4, 9 become 1, 2, 3 and the indices of edges 1, 2, 3 remain the same. For element 2, vertices 4, 5, 9 are associated with it. Since vertices 4 and 9 have been re-numbered in element 1, we only need to re-number vertex 5, which now becomes 4. Edge 2 has been re-numbered in element 1. We only need to renumber edges 4 and 5. They have the same indices in the re-ordering procedure. Once we loop over all elements, we have the reordered mesh as shown in Figure 2.3(d). In our implementation, steps 4–5 are combined together for better efficiency.

2.4 Numerical results

This section is devoted to assessing the numerical performance and parallel scalability of our implementation on different test cases. In all experiments, the relative residual of the outer iteration is set to 1e-6 and the relative residual of the inner iteration is set to 1e-8, unless explicitly specified. The code is executed on a cluster with up to 12 nodes. Each node has eight 2.6GHz Intel processors and 16G RAM.

The following notation is used to record our results: np denotes the number of processors of the run, its is the number of outer MINRES iterations, its_{i_1} is the number of inner CG iterations for solving $\mathscr{P}_M$, its_{i_2} is the number of CG iterations for solving L, while t_s and t_a denote the average times needed for the solve phase and the assemble phase in seconds, respectively. The parameter t_{AMG} is the time spent in seconds in one BoomerAMG V-cycle for solving L.

2.4.1 Example 1: structured mesh

The first example is a simple domain with a structured mesh. The domain is a cube, $\Omega = (-1,1)^3$. We test both homogeneous and inhomogeneous coefficient cases. In the homogeneous case, we set $\mu_r = \varepsilon_r = 1$. In the variable coefficient

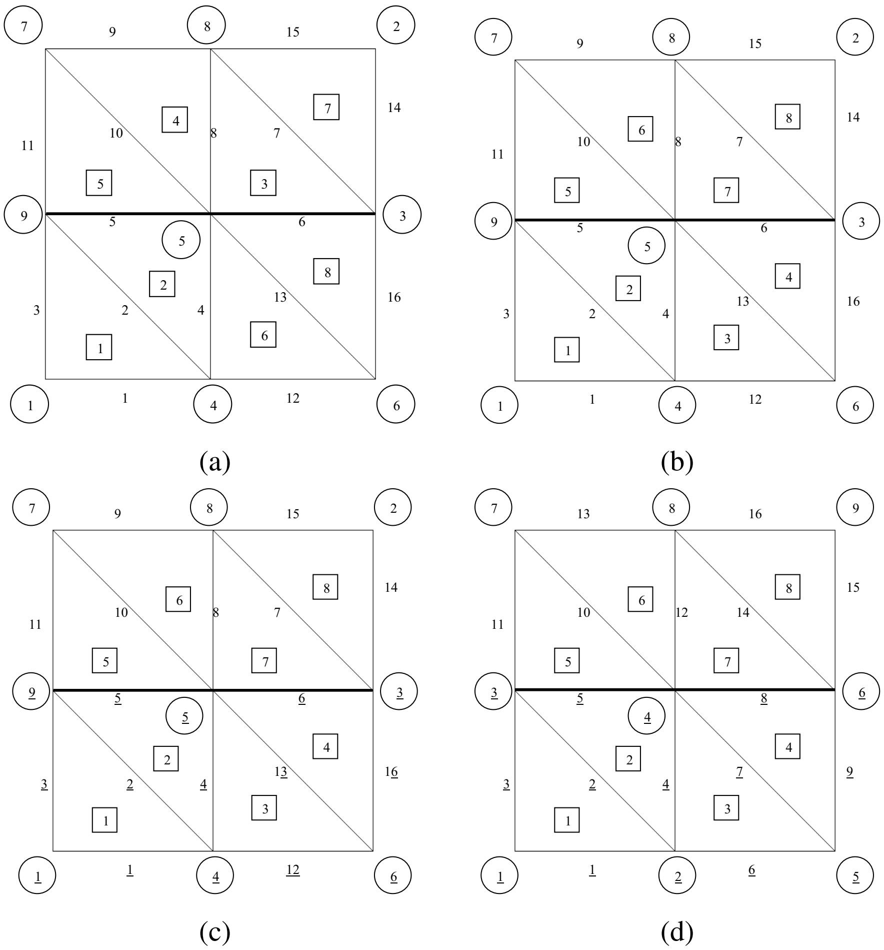


Figure 2.3: Mesh partitioning example. (a) Original mesh after step 1. The thick line (in all four figures) indicates the partitioning of elements after step 2. (b) Reordering of elements after step 3. (c) Distribution of degrees of freedom after step 4. (d) Reordered mesh after step 5.

case, we assume that there are eight subdomains in the cube as shown in Figure 2.4 and each subdomain has piecewise constant coefficients.

The coefficients are

$$\mu_r = \varepsilon_r = \begin{cases} 1a & \text{if } x < 0 \text{ and } y < 0 \text{ and } z < 0, \\ 2a & \text{if } x > 0 \text{ and } y < 0 \text{ and } z < 0, \\ 3a & \text{if } x < 0 \text{ and } y > 0 \text{ and } z < 0, \\ 4a & \text{if } x > 0 \text{ and } y > 0 \text{ and } z < 0, \\ 5a & \text{if } x < 0 \text{ and } y < 0 \text{ and } z > 0, \\ 6a & \text{if } x > 0 \text{ and } y < 0 \text{ and } z > 0, \\ 7a & \text{if } x < 0 \text{ and } y > 0 \text{ and } z > 0, \\ 8a & \text{otherwise,} \end{cases} \tag{2.24}$$

where a is a constant. We set the right-hand side so that the solution of (2.1) is given by

$$\mathbf{u}(x,y,z) = \begin{pmatrix} u_1(x,y,z) \\ u_2(x,y,z) \\ u_3(x,y,z) \end{pmatrix} = \begin{pmatrix} (1-y^2)(1-z^2) \\ (1-x^2)(1-z^2) \\ (1-x^2)(1-y^2) \end{pmatrix} \tag{2.25}$$

and

$$p(x,y,z) = (1-x^2)(1-y^2)(1-z^2). \tag{2.26}$$

In this example, the homogeneous boundary conditions in (2.1) are satisfied.

Uniformly refined meshes are constructed as shown in Figure 2.5(a). The number of elements and matrix sizes are given in Table 2.1. Figure 2.5(b) shows how grid C1 is partitioned across 3 processors. Elements with the same color are stored on the same processor. Elements with the same color are clustered together, which means the communication cost is minimal. Table 2.2 shows the local numbers of elements and degrees of freedom on each processor for grid C1. The number of degrees of freedom on each processor is roughly the same, which indicates the load is balanced.

In our first experiment, material coefficients are homogeneous. Scalability results are shown in Table 2.3 and results with different values of k are shown in Table 2.4. To test scalability, we refine the mesh and increase the number of processors in a proportional manner, so that the problem size per processor remains

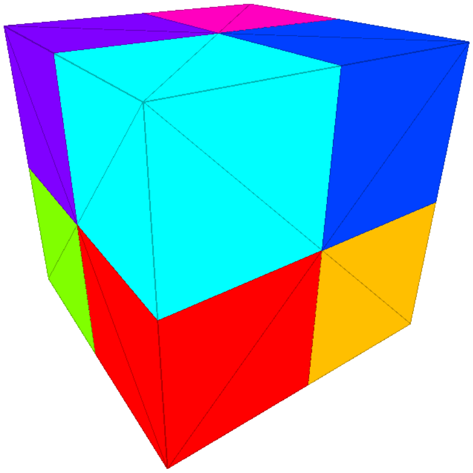
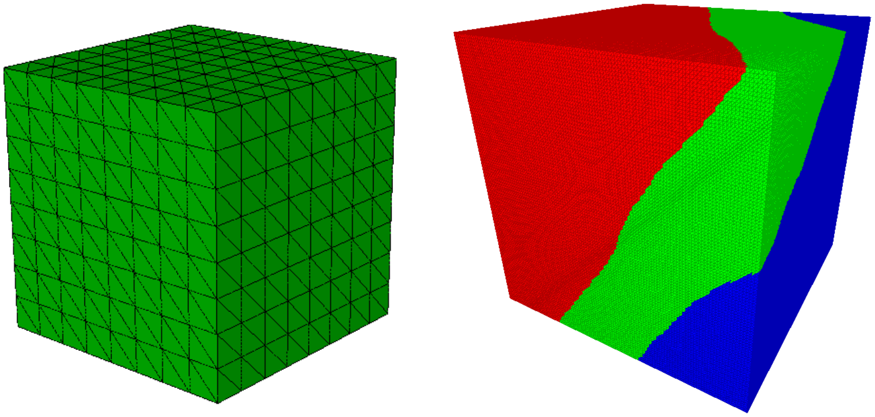


Figure 2.4: Example 1. Distribution of material coefficients.



(a)

(b)

Figure 2.5: Example 1. (a) Structured mesh. (b) grid C1 partitioned on 3 processors.

constant. Full scalability would then imply that the computation time also remains constant. We observe in Table 2.3 that when the mesh is refined, the numbers of outer and inner iterations stay constant, which demonstrates the scalability of our method. The time spent in the assembly also scales very well. The time spent in the

grid	Nel	$n+m$
C1	7,146,096	9,393,931
C2	14,436,624	19,034,163
C3	29,478,000	38,958,219

Table 2.1: Example 1. Number of elements (Nel) and the size of the linear systems ($n+m$) for grids C1–C3.

processor	local elements	local DOFs
1	2,359,736	3,119,317
2	2,424,224	3,199,406
3	2,362,136	3,075,208

Table 2.2: Example 1. Partitioning of grid C1.

solve increases slightly. This is because each BoomerAMG V-cycle seems to take more time when the mesh is refined. For different values of k, we have observed very similar computation times as in Table 2.3. Table 2.4 shows that the iteration counts stay the same for the values of k that we have selected. This demonstrates the scalability of our solver.

Notice that the highest wave number in the tables, $k = \frac{1}{4}$, corresponds to frequency $\omega = 7.9\text{e}7$. The wave length is then 8π, which is still considerably larger than our domain. Thus, our discretization well resolves the length scale of the time-harmonic waves.

np	grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
3	C1	5	34	7	1,473.58	44.83	16.03
6	C2	5	35	9	1,634.17	45.26	20.55
12	C3	5	34	9	1,879.06	48.93	25.39

Table 2.3: Example 1. Iteration counts and computation times for various grids, $k = 0$.

Setting the outer tolerance as 1e-6, we test our solver with different inner tolerances. The results are given in Table 2.5. We see that when the inner tolerance is looser, fewer inner iterations are required, however if the tolerance is too loose, more outer iterations are required. For this example, 1e-6 is the optimal inner

		$k=0$			$k=\frac{1}{8}$			$k=\frac{1}{4}$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
3	C1	5	34	7	5	34	7	5	34	7
6	C2	5	35	9	5	35	9	5	35	9
12	C3	5	34	9	5	34	9	5	34	9

Table 2.4: Example 1. Iteration counts for various values of k.

tolerance.

In the remaining examples, we stick to 1e-6 and 1e-8 as outer and inner tolerances respectively. We select a tight inner tolerance since one of our goals is to investigate the speed of convergence of outer iterations.

inner tol	its	its_{i_1}	its_{i_2}	t_s(sec)
1e-10	5	43	9	557.08
1e-9	5	39	8	505.30
1e-8	5	35	8	440.92
1e-7	5	30	7	383.34
1e-6	6	21	7	375.91
1e-5	8	20	6	409.57
1e-4	21	16	5	794.15

Table 2.5: Example 1. Iteration counts and computation times for grid C1 on 16 processors for inexact inner iterations, $k=0$.

Next we test the variable coefficient case. Table 2.6 shows the iteration counts for different variable coefficient cases. Note that the larger a is, the more variant the coefficients are in different regions. As expected, the eigenvalue bound depends on the coefficients. Table 2.6 shows that as a increases, the material coefficients vary more dramatically, the eigenvalue bounds in Theorem 2 get looser and the iteration counts increase. In the variable coefficient case, both the inner and outer iterations are not sensitive to changes of the mesh size, which demonstrates that our method is scalable in the variable coefficient case.

		$a=1$			$a=10$			$a=20$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
3	C1	10	81	8	58	147	8	107	135	8
6	C2	10	87	10	59	154	10	105	145	10
12	C3	10	87	11	58	178	10	107	187	10

Table 2.6: Example 1. Iteration counts for various values of a, $k=0$.

2.4.2 Example 2: unstructured mesh

In this example, we test the problem in a complicated domain with a quasi-uniform mesh. The domain is a complicated 3D gear, which is bounded in $(0.025, 0.975) \times (0.025, 0.975) \times (0.025, 0.15292)$. We test two different cases: constant and variable coefficient cases. In the constant coefficient test, we set $\mu_r = \varepsilon_r = 1$. In the variable coefficient case, we assume that there are four subdomains in the gear as shown in Figure 2.6 and each subdomain has piecewise constant coefficients. The coefficients are

$$\mu_r = \varepsilon_r = \begin{cases} 1a & \text{if } x < 0.5 \text{ and } y < 0.5, \\ 2a & \text{if } x > 0.5 \text{ and } y < 0.5, \\ 3a & \text{if } x < 0.5 \text{ and } y > 0.5, \\ 4a & \text{otherwise,} \end{cases}$$

where a is a constant. In both tests, we set the right-hand side function so that the exact solution is given by (2.25) and (2.26), and enforce the inhomogeneous boundary conditions in a standard way.

The domain is meshed with quasi-uniform tetrahedra as shown in Figure 2.7(a). The number of elements and matrix sizes are given in Table 2.7. Figure 2.7(b) shows how to partition G1 across 4 processors.

First, we test our approach with constant coefficients. The scalability results are shown in Table 2.8. Results with different values of k are shown in Table 2.9. We observe that when the mesh is refined, both the inner and outer iteration counts stay constant. Again, the time spent in the assembly scales very well. The time spent in the solve is increasing slightly, which can be explained by the increase in t_{AMG}. We note that the computational times for different values of k are similar to

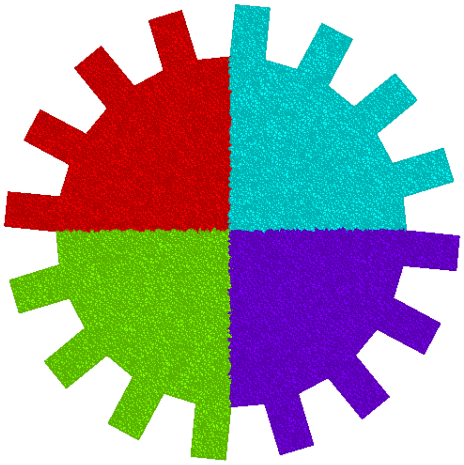
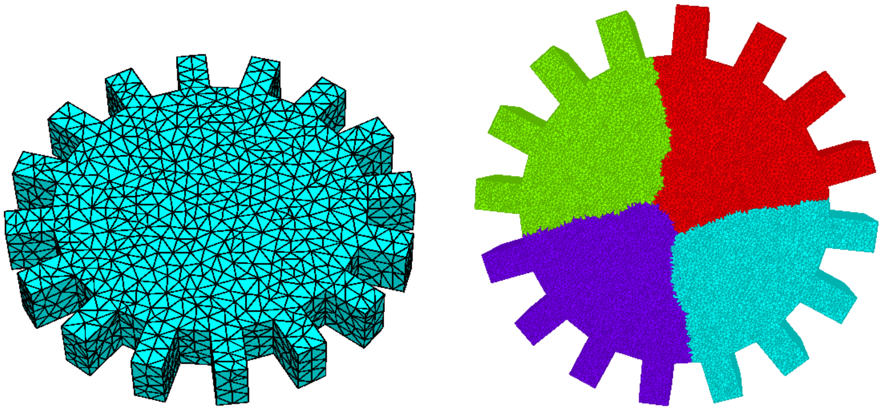


Figure 2.6: Example 2. Distribution of material coefficients.



(a)

(b)

Figure 2.7: Example 2. (a) Quasi-uniform mesh on the gear. (b) grid G1 partitioned on 4 processors.

those reported for $k = 0$ in Table 2.8. Table 2.9 shows that the iteration counts do not change for different k values.

Next we test the variable coefficient example. Table 2.10 shows the results for different variable coefficient cases. Again, we observe that when the variation of

grid	Nel	$n+m$
G1	723,594	894,615
G2	1,446,403	1,810,413
G3	2,889,085	3,650,047
G4	5,778,001	7,354,886

Table 2.7: Example 2. Number of elements (Nel) and the size of the linear systems ($n+m$) for grids G1–G4.

np	grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
4	G1	4	58	8	146.15	5.86	0.73
8	G2	4	61	9	345.32	8.07	1.23
16	G3	4	61	9	399.49	7.92	1.66
32	G4	4	64	10	467.57	8.17	2.12

Table 2.8: Example 2. Iteration counts and computation times for various grids, $k=0$.

		$k=0$			$k=\frac{1}{8}$			$k=\frac{1}{4}$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	G1	4	58	8	4	58	8	4	58	8
8	G2	4	61	9	4	61	9	4	61	9
16	G3	4	61	9	4	61	9	4	61	9
32	G4	4	64	10	4	64	10	4	64	10

Table 2.9: Example 2. Iteration counts for various values of k.

the material coefficients increases, the iteration counts increase. We also observe that both the inner and outer iterations are not sensitive to changes of the mesh size for the example with unstructured meshes.

2.4.3 Example 3: locally refined mesh

The last example is the Fichera corner problem. We are interested in testing our solver on a series of locally refined meshes. The domain $\Omega = (-1,1)^3 \backslash [0,1) \times [0,-1) \times [0,1)$ is a cube with a missing corner. We also test both homogeneous and inhomogeneous coefficients. In the homogeneous coefficient case, we set $\mu_r =$

		$a=1$			$a=10$			$a=20$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	G1	5	159	8	21	294	8	35	250	8
8	G2	5	167	9	20	328	9	35	296	9
16	G3	5	177	9	20	389	9	34	342	9
32	G4	5	185	10	20	435	10	34	404	10

Table 2.10: Example 2. Iteration counts for various values of a, $k=0$.

$\varepsilon_r = 1$. In the inhomogeneous case, we assume that there are seven subdomains in the domain as shown in Figure 2.8 and each subdomain has piecewise constant coefficients. The coefficients are the same as (2.24). In both tests, we set the right-hand side function so that the exact solution is given by (2.25) and (2.26), and enforce the inhomogeneous boundary conditions in a standard way.

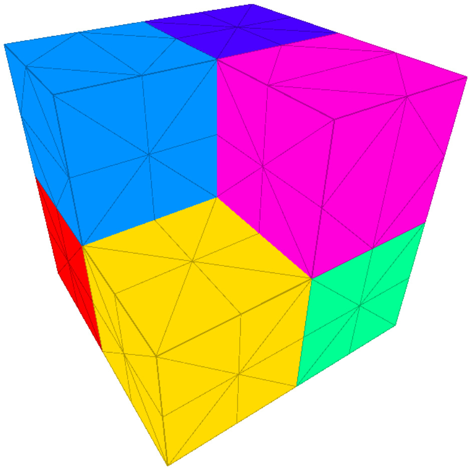


Figure 2.8: Example 3. Distribution of material coefficients.

The domain is discretized with locally refined meshes towards the corner. Figure 2.9 shows an example of a sequence of locally refined meshes. The number of elements and matrix sizes are given in Table 2.11. Figure 2.10 shows the partitioning of F1 on 4 processors.

First, we assume that the material coefficients are homogeneous. The scalabil-

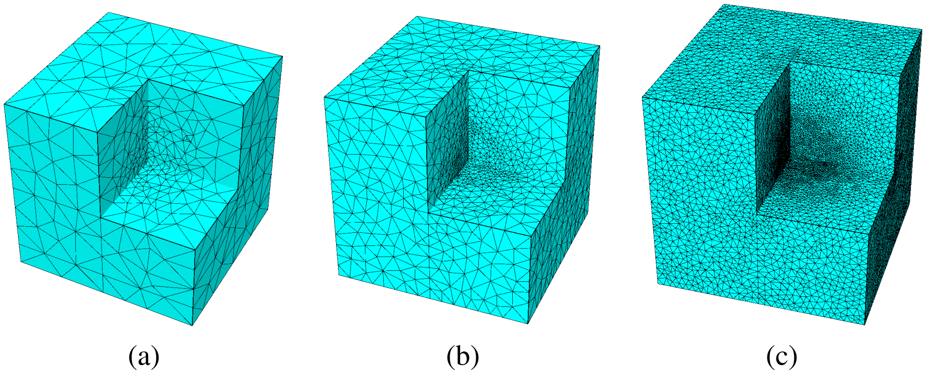


Figure 2.9: Example 3. A sequence of locally refined meshes.

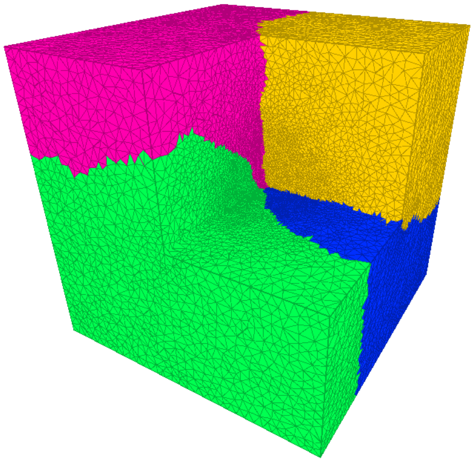


Figure 2.10: Example 3. Illustration of grid F1 partitioned on 4 processors.

grid	Nel	$n+m$
F1	781,614	957,277
F2	1,543,937	1,917,649
F3	3,053,426	3,832,895
F4	6,072,325	7,689,953

Table 2.11: Example 3. Number of elements (Nel) and the size of the linear systems $(n+m)$ for grids F1–F4.

ity results are shown in Table 2.12. The results with different values of k are shown in Table 2.13. On the locally refined meshes, we also observe that when the mesh is refined, both the inner and outer solvers are scalable. Again, the time spent in the assembly scales very well. The time spent in the solve is increasing, which is due to the increasing cost of BoomerAMG V-cycles. Table 2.13 shows that the iteration counts do not change for different wave numbers. As in the previous examples, in our experiments the computational times are also roughly the same as in Table 2.12 for different values of k.

np	grid	its	its_{i_1}	its_{i_2}	t_s(sec)	t_a(sec)	t_{AMG}(sec)
4	F1	5	59	8	204.82	6.54	0.88
8	F2	5	56	9	213.16	6.39	1.17
16	F3	5	58	10	253.81	6.29	1.65
32	F4	5	62	10	314.88	6.38	1.99

Table 2.12: Example 3. Iteration counts and computation times for various grids, $k = 0$.

		$k = 0$			$k = \frac{1}{8}$			$k = \frac{1}{4}$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	F1	5	59	8	5	59	8	5	59	8
8	F2	5	56	9	5	56	9	5	56	9
16	F3	5	58	10	5	58	10	5	58	10
32	F4	5	62	10	5	62	10	4	63	10

Table 2.13: Example 3. Iteration counts for various values of k.

Next, we test the variable coefficient case. Table 2.14 shows the results for different variable coefficient cases. Again, we observe that when the coefficients vary more, the iteration counts increase, but the scalability is very good.

		$a = 1$			$a = 10$			$a = 20$		
np	grid	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}	its	its_{i_1}	its_{i_2}
4	F1	7	85	9	40	157	9	74	134	9
8	F2	7	92	9	40	178	9	73	157	9
16	F3	7	100	10	39	192	10	72	167	10
32	F4	7	107	10	39	216	10	71	206	10

Table 2.14: Example 3. Iteration counts for various values of a, $k = 0$.

Chapter 3

A new mixed finite element method for the incompressible MHD equations

In this chapter[1] we design and numerically test a new finite element discretization for the steady-state incompressible MHD equations that model electrically conducting fluids in the presence of a magnetic field. We use divergence-confirming elements for the approximation of the velocity field and curl-confirming elements for the magnetic field. This discretization yields exactly divergence-free velocity approximations. Here, we recapitulate the theoretical properties of our method.

This chapter is structured as follows. We start by discussing related work in Section 3.1. Section 3.2 is devoted to the finite element discretization. In Section 3.3 we present a series of numerical experiments validating the theoretical results.

[1] The notation in this chapter is slightly different than the notation in Chapter 2. The variables $\mathbf{u}$ and p are now used to denote the velocity field and pressure field in the MHD system, respectively. The forms A, B, M are also different than in Chapter 2, but the meaning should be obvious from the context.

3.1 Background and related work

We consider a standard form of the incompressible MHD equations as derived in (1.15); see also [3, Section 2] and [41, 42, 49]. That is, we neglect phenomena involving high frequency as well as the convection current, and consider a non-polarizable, non-magnetizable and homogeneous medium. In addition, to make the curl-curl operator arising in the Maxwell equations amenable to discretization with Nédélec elements, we use the mixed formulation proposed in [89]. The governing equations are then of the form

$$-\nu\,\Delta\mathbf{u}+(\mathbf{u}\cdot\nabla)\mathbf{u}+\nabla p-\kappa\,(\nabla\times\mathbf{b})\times\mathbf{b}=\mathbf{f}\qquad\text{in }\Omega,\tag{3.1a}$$

$$\kappa\nu_m\nabla\times(\nabla\times\mathbf{b})+\nabla r-\kappa\nabla\times(\mathbf{u}\times\mathbf{b})=\mathbf{g}\qquad\text{in }\Omega,\tag{3.1b}$$

$$\nabla\cdot\mathbf{u}=0\qquad\text{in }\Omega,\tag{3.1c}$$

$$\nabla\cdot\mathbf{b}=0\qquad\text{in }\Omega.\tag{3.1d}$$

Here, **u** is the velocity, **b** the magnetic field, *p* the hydrodynamic pressure, and *r* is a Lagrange multiplier associated with the divergence constraint on the magnetic field **b**. The functions **f** and **g** represent external force terms.

We consider the following homogeneous Dirichlet boundary conditions:

$$\mathbf{u}=\mathbf{0}\qquad\text{on }\partial\Omega,\tag{3.2a}$$

$$\mathbf{n}\times\mathbf{b}=\mathbf{0}\qquad\text{on }\partial\Omega,\tag{3.2b}$$

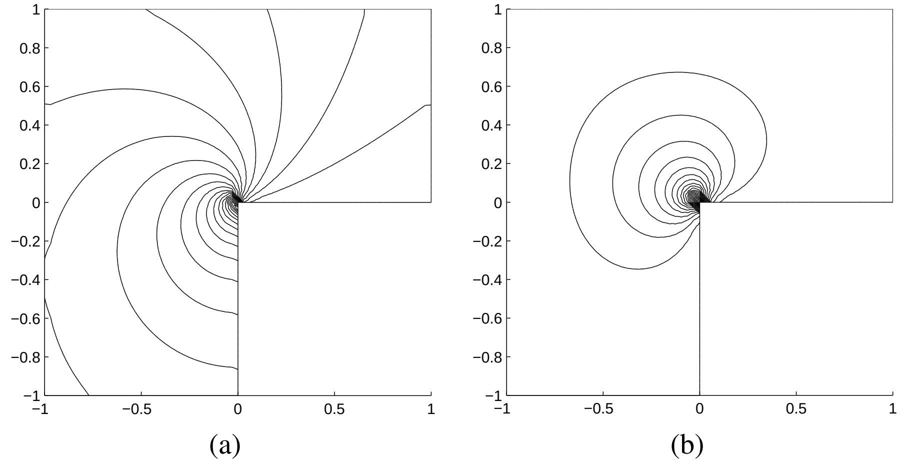
$$r=0\qquad\text{on }\partial\Omega,\tag{3.2c}$$

with **n** being the unit outward normal on $\partial\Omega$. By taking the divergence of the magnetostatic equation (3.1b), we obtain the Poisson problem

$$\Delta r=\nabla\cdot\mathbf{g}\quad\text{in }\Omega,\qquad r=0\quad\text{on }\partial\Omega.\tag{3.3}$$

Since **g** is typically divergence-free in physical applications, the multiplier *r* is typically zero and its primary purpose is to ensure stability; see [32, Section 3].

Various finite element methods for discretizing linear and nonlinear MHD systems can be found in the literature. The magnetic field is often approximated by standard nodal (i.e., H^1-conforming) finite elements [3, 41, 47–49]. However, since the strongest magnetic singularities have regularity below H^1 straightforwardly applied nodal elements may fail to resolve them in non-convex polyhedral domains; see [26] and the references therein. This is illustrated in Figure 3.1, for the strongest magnetic singularity in an L-shaped domain in two dimensions. A number of remedies have been proposed for electromagnetic problems, for example the weighted regularization approach in [27] or the approach in [14], whereby the divergence of the electric field is stabilized in $H^{-\alpha}$ with $\frac{1}{2} < \alpha < 1$. In [50], weighted regularization has been applied to a full incompressible MHD system.



(a)

(b)

Figure 3.1: Failure of H^1-conforming elements. (a) Contours of the first component of the strongest magnetic singularity in an L-shaped domain; (b) nodal approximation with piecewise linear elements. It is evident from the plots that, on a sufficiently fine mesh, nodal elements cannot correctly resolve the magnetic field.

In the mixed formulation of [89] the above mentioned difficulties associated with nodal elements are seamlessly avoided without the need for stabilizing the divergence. This approach amounts to introducing the Lagrange multiplier r, and yields the PDE system (3.1). As a result, it is possible to use curl-conforming Nédélec elements for approximating the magnetic field. This makes this approach feasible in situations of highly singular magnetic fields [55, 73, 76]. In the context of incompressible magnetohydrodynamics, a related mixed approach for the discretization of the magnetic unknowns was presented in [34].

We are interested in discretizations for incompressible MHD problems that are based on DG methods; see, e.g., the surveys [20, 21, 31] and the references therein. In [48], an interior penalty technique is applied to enforce continuity of the magnetic variable across domains with different electromagnetic properties, while nodal elements are employed in the interior. A full DG method is proposed in [58] for a linearized variant of the system (3.1), whereby all the variables are approximated in discontinuous finite element spaces, based on existing discretizations for the Oseen and Maxwell equations [22, 23, 57]. However, this approach requires a large number of degrees of freedom. Furthermore, a straightforward extension to the nonlinear setting in a locally conservative fashion would require a post-processing procedure for smoothing the DG velocity approximations throughout the nonlinear iteration [23].

In this chapter, we design a new finite element discretization, in an attempt to overcome the above mentioned difficulties. Instead of discontinuous elements for all unknowns, we use divergence-conforming BDM elements [15, 24] for the approximation of the velocity field, and curl-conforming Nédélec elements [76] for the magnetic field, thereby substantially reducing the total number of the coupled degrees of freedom. The H^1-continuity of the velocity field is again enforced by a DG technique. A central feature of this discretization is that it yields exactly divergence-free velocity approximations, guaranteeing stability of the linearized system within each Picard iteration, without any other modifications. We note that divergence-conforming discretizations were originally proposed and analyzed for the incompressible Navier-Stokes equations in [24]. For the magnetic approximation we have a discrete version of the desirable property (3.3), in contrast to the full DG method presented in [58]. Our method correctly captures the strongest magnetic singularities in non-convex domains.

We have nearly optimal a-priori error estimates for the two-dimensional case, and slightly suboptimal ones for the three-dimensional case. A full description of our theoretical results can be found in [44, 95]. Here we recapitulate the main results and present an extensive set of numerical tests. The numerical results clearly indicate optimal convergence rates in two and three dimensions.

3.2 Mixed finite element discretization

In this section, we introduce a mixed finite element method that employs divergence-conforming elements for the approximation of the velocity field and curl-conforming elements for the magnetic field. The H^1-continuity of the velocity is enforced by a DG technique.

3.2.1 Variational formulation

We assume Ω is a bounded simply-connected Lipschitz polytope in $\mathbb{R}^d$ ($d = 2$ or 3), with a connected boundary $\partial\Omega$. Recall the spaces $\mathbf{C}$ and S defined in (2.3). Upon setting

$$\begin{aligned}
\mathbf{V} &= H_0^1(\Omega)^d = \{\,\mathbf{u} \in H^1(\Omega)^d \,:\, \mathbf{u} = \mathbf{0} \text{ on } \partial\Omega\,\},\\
Q &= L_0^2(\Omega) = \{\, p \in L^2(\Omega) \,:\, (p\,,1)_\Omega = 0\,\},
\end{aligned}$$

the variational formulation of the incompressible MHD system (3.1)–(3.2) amounts to finding $(\mathbf{u},\mathbf{b},p,r) \in \mathbf{V}\times\mathbf{C}\times Q\times S$ such that

$$\begin{aligned}
A(\mathbf{u},\mathbf{v}) + O(\mathbf{u},\mathbf{u},\mathbf{v}) + C(\mathbf{b},\mathbf{v},\mathbf{b}) + B(\mathbf{v},p) &= (\mathbf{f},\mathbf{v})_\Omega, && (3.4a)\\
M(\mathbf{b},\mathbf{c}) - C(\mathbf{b},\mathbf{u},\mathbf{c}) + D(\mathbf{c},r) &= (\mathbf{g},\mathbf{c})_\Omega, && (3.4b)\\
B(\mathbf{u},q) &= 0, && (3.4c)\\
D(\mathbf{b},s) &= 0, && (3.4d)
\end{aligned}$$

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

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

Furthermore, we define the norm of the source terms by

$$\|\|(\mathbf{f},\mathbf{g})\|\| = \left(\|\mathbf{f}\|_{L^2(\Omega)}^2 + \|\mathbf{g}\|_{L^2(\Omega)}^2\right)^{\frac{1}{2}}.$$

Finally, we introduce the parameters

$$\bar{\nu} = \min\{\nu, \kappa\nu_m\},$$
$$\bar{\kappa} = \max\{1, \kappa\}.$$

We assume that there is a constant $c_1 > 0$ only depending on Ω such that for small data with

$$c_1 \frac{\bar{\kappa}}{\bar{\nu}^2} \|\|(\mathbf{f},\mathbf{g})\|\| < 1, \tag{3.5}$$

the MHD problem (3.4) has a unique solution $(\mathbf{u},\mathbf{b},p,r)$ in $\mathbf{V}\times\mathbf{C}\times Q\times S$; see the proof in [89, Corollary 2.18 and Remark 2.14].

3.2.2 Mixed discretization

We consider a family of regular and quasi-uniform triangulations $\mathscr{T}_h$ of mesh size h that partition the domain Ω into simplices $\{K\}$ (i.e., triangles for $d = 2$ and tetrahedra for $d = 3$). We denote by $\mathscr{F}_h$ the set of all edges ($d = 2$) or faces ($d = 3$) of $\mathscr{T}_h$. In the following, we generically refer to elements in $\mathscr{F}_h$ as faces. As usual, h_K denotes the diameter of the element K, and h_F is the diameter of the face F. Finally, we write $\mathbf{n}_K$ for the unit outward normal vector on the boundary ∂K of K.

The average and jump operators are defined as follows. Let $F = \partial K \cap \partial K'$ be an interior face shared by K and K', and let $\mathbf{x} \in F$. Let ϕ be a generic piecewise smooth function and denote by ϕ and ϕ' the traces of ϕ on F taken from within the interior of K and K', respectively. Then, we define the mean value $\{\{\cdot\}\}$ at $\mathbf{x} \in F$ as

$$\{\{\phi\}\} = \frac{1}{2}(\phi + \phi').$$

Furthermore, for a piecewise smooth vector-valued function ϕ, we define the jump:

$$[\![\phi]\!] = \phi \otimes \mathbf{n}_K + \phi' \otimes \mathbf{n}_{K'},$$

where $\phi \otimes \mathbf{n} = (\phi_i n_j)_{1 \le i,j \le d}$. On a boundary face $F = \partial K \cap \partial\Omega$, we set accordingly

$$\{\!\{\phi\}\!\} = \phi, \qquad [\![\phi]\!] = \phi \otimes \mathbf{n}.$$

For $k \ge 1$, we wish to approximate the solution of (3.1)–(3.2) by finite element functions $(\mathbf{u}_h, \mathbf{b}_h, p_h, r_h) \in \mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$, where

$$\mathbf{V}_h = \{\, \mathbf{u} \in H_0(\mathrm{div};\Omega) : \mathbf{u}|_K \in \mathscr{P}_k(K)^d, K \in \mathscr{T}_h \,\},$$
$$Q_h = \{\, p \in L^2_0(\Omega) : p|_K \in \mathscr{P}_{k-1}(K), K \in \mathscr{T}_h \,\},$$

and the spaces $\mathbf{C}_h$ and S_h are defined in (2.5). Here, we denote by $H_0(\mathrm{div};\Omega)$ the space

$$H_0(\mathrm{div};\Omega) = \left\{ \mathbf{u} \in L^2(\Omega)^d : \nabla \cdot \mathbf{u} \in L^2(\Omega), \mathbf{u} \cdot \mathbf{n} = 0 \text{ on } \partial\Omega \right\}.$$

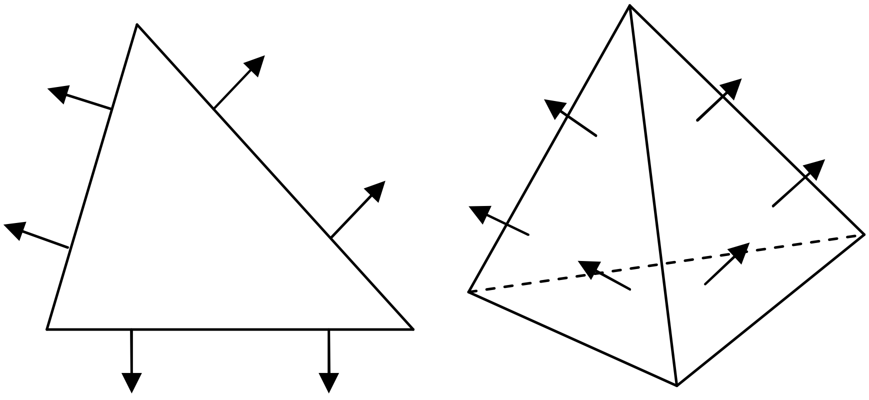
The space $\mathbf{V}_h$ is the divergence-conforming BDM space (see [15, Section III.3] for details); it has degrees of freedom specified for the normal components of functions along faces. Figure 3.2 shows the degrees of freedom on the lowest order BDM elements in 2D and 3D. We notice that the finite element spaces $\mathbf{C}_h$, Q_h and S_h are conforming in $\mathbf{C}$, Q and S, respectively, while $\mathbf{V}_h$ is non-conforming in $\mathbf{V}$.

Now we consider the following finite element method: find $(\mathbf{u}_h, \mathbf{b}_h, p_h, r_h) \in \mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$ such that

$$\begin{aligned} A_h(\mathbf{u}_h, \mathbf{v}) + O_h(\mathbf{u}_h, \mathbf{u}_h, \mathbf{v}) + C(\mathbf{b}_h, \mathbf{v}, \mathbf{b}_h) + B(\mathbf{v}, p_h) &= (\mathbf{f}, \mathbf{v})_\Omega, && (3.6a)\\ M(\mathbf{b}_h, \mathbf{c}) - C(\mathbf{b}_h, \mathbf{u}_h, \mathbf{c}) + D(\mathbf{c}, r_h) &= (\mathbf{g}, \mathbf{c})_\Omega, && (3.6b)\\ B(\mathbf{u}_h, q) &= 0, && (3.6c)\\ D(\mathbf{b}_h, s) &= 0, && (3.6d) \end{aligned}$$

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

The form A_h associated with the Laplacian is chosen as the standard interior



(a) 2D

(b) 3D

Figure 3.2: A graphical illustration of the degrees of freedom for the lowest order BDM element in 2D and 3D.

penalty form [4, 5]:

$$\begin{aligned} A_h(\mathbf{u},\mathbf{v}) = \int_\Omega \nu \nabla_h \mathbf{u} : \nabla_h \mathbf{v}\, d\mathbf{x} - \sum_{F\in\mathscr{F}_h} \int_F \{\!\{\nu \nabla_h \mathbf{u}\}\!\} : [\![\mathbf{v}]\!]\, ds \\ - \sum_{F\in\mathscr{F}_h} \int_F \{\!\{\nu \nabla_h \mathbf{v}\}\!\} : [\![\mathbf{u}]\!]\, ds + \sum_{F\in\mathscr{F}_h} \frac{a_0 \nu}{h_F} \int_F [\![\mathbf{u}]\!] : [\![\mathbf{v}]\!]\, ds. \end{aligned}$$

Here, ∇_h is the elementwise gradient operator, and $a_0 > 0$ is the interior penalty stabilization parameter; it has to be chosen larger than a threshold value which is independent of h, ν, κ and ν_m. For the convection form, we take the standard upwind form [68]:

$$\begin{aligned} O_h(\mathbf{w},\mathbf{u},\mathbf{v}) = \sum_{K\in\mathscr{T}_h} \int_K (\mathbf{w}\cdot\nabla)\mathbf{u}\cdot\mathbf{v}\, d\mathbf{x} \\ + \sum_{K\in\mathscr{T}_h} \int_{\partial K\setminus\partial\Omega} \frac{1}{2}(\mathbf{w}\cdot\mathbf{n}_K - |\mathbf{w}\cdot\mathbf{n}_K|)(\mathbf{u}^e - \mathbf{u})\cdot\mathbf{v}\, ds \\ - \int_{\partial\Omega} \frac{1}{2}(\mathbf{w}\cdot\mathbf{n} - |\mathbf{w}\cdot\mathbf{n}|)\mathbf{u}\cdot\mathbf{v}\, ds. \end{aligned}$$

Here, $\mathbf{u}^e$ is the trace of $\mathbf{u}$ taken from the exterior of K. The remaining forms are the same as in the continuous case. Notice that due to the presence of the upwind terms the form $O_h(\mathbf{w},\mathbf{u},\mathbf{v})$ is not linear in the first argument.

By choosing the divergence-conforming BDM elements as the approximating space for the velocity, the method gives exactly divergence-free velocity approximations; cf. [24]. Moreover, the Lagrange multiplier r_h vanishes identically for divergence-free source terms, thereby mimicking the continuous property in (3.3).

Proposition 3 *Let* $(\mathbf{u}_h, \mathbf{b}_h, p_h, r_h)$ *solve (3.6). Then we have:*

(i) $\nabla \cdot \mathbf{u}_h = 0$ *in* Ω*.*

(ii) the Lagrange multiplier r_h *is the solution of*

$$(\nabla r_h, \nabla s)_\Omega = (\mathbf{g}, \nabla s)_\Omega \qquad \forall s \in S_h.$$

In particular, if $\mathbf{g}$ *is solenoidal, then* $r_h \equiv 0$*.*

Proof To prove item (i), we proceed as in [24]. We note that $\nabla \cdot \mathbf{u}_h$ has vanishing mean value on Ω, and is a discontinuous polynomial of degree $k-1$. Thus, we have $\nabla \cdot \mathbf{u}_h \in Q_h$. Equation (3.6c) then implies that $\nabla \cdot \mathbf{u}_h$ is orthogonal to all functions $q \in Q_h$. Therefore, it is equal to zero.

To prove item (ii), we take $\mathbf{c} = \nabla s$ in equation (3.6b) (noting that $\nabla S_h \subset \mathbf{C}_h$) and obtain

$$(\mathbf{g}, \nabla s)_\Omega = M(\mathbf{b}_h, \nabla s) - C(\mathbf{b}_h, \mathbf{u}_h, \nabla s) + D(\nabla s, r_h) = D(\nabla s, r_h).$$

Here, we have used the fact that $\nabla \times \nabla s = \mathbf{0}$. Therefore, r_h satisfies

$$(\nabla r_h, \nabla s)_\Omega = (\mathbf{g}, \nabla s)_\Omega \qquad \forall s \in S_h.$$

Since $(\mathbf{g}, \nabla s)_\Omega = (\nabla \cdot \mathbf{g}, s)_\Omega$, we have $r_h \equiv 0$ provided that $\nabla \cdot \mathbf{g} = 0$. ▮

The solution of (3.6) can be found by employing the following Picard-type iteration: given $(\mathbf{u}_h^{k-1}, \mathbf{b}_h^{k-1}) \in \mathbf{V}_h \times \mathbf{C}_h$, let $(\mathbf{u}_h^k, \mathbf{b}_h^k, p_h^k, r_h^k)$ in $\mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$ be the solution of the linearized Oseen-type problem

$$
\begin{aligned}
A_h(\mathbf{u}_h^k,\mathbf{v})+O_h(\mathbf{u}_h^{k-1},\mathbf{u}_h^k,\mathbf{v})+C(\mathbf{b}_h^{k-1},\mathbf{v},\mathbf{b}_h^k)+B(\mathbf{v},p_h^k) &= (\mathbf{f},\mathbf{v})_\Omega, && (3.7a)\\
B(\mathbf{u}_h^k,q) &= 0, && (3.7b)\\
M(\mathbf{b}_h^k,\mathbf{c})-C(\mathbf{b}_h^{k-1},\mathbf{u}_h^k,\mathbf{c})+D(\mathbf{c},r_h^k) &= (\mathbf{g},\mathbf{c})_\Omega, && (3.7c)\\
D(\mathbf{b}_h^k,s) &= 0, && (3.7d)
\end{aligned}
$$

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

Theorem 3.2 in [44] guarantees the convergence of the iterates $\{(\mathbf{u}_h^k,\mathbf{b}_h^k,p_h^k,r_h^k)\}$, for $k=1,2,\ldots$, to the solution $(\mathbf{u}_h,\mathbf{b}_h,p_h,r_h)$ of (3.6) for any initial guess $(\mathbf{u}_h^0,\mathbf{b}_h^0)\in\mathbf{V}_h\times\mathbf{C}_h$ with exactly divergence-free $\mathbf{u}_h^0$, provided that the small data assumption in (3.5) is satisfied. However, the scheme is only linearly convergent, as we illustrate in Section 3.3.

Remark 4 *A more efficient nonlinear solver such as Newton's method can also be used for solving* (3.6)*; see, e.g., [40, 42, 49]. When upwinding is not incorporated, Newton's method can be straightforwardly applied. However, when upwind terms are included, adapting the nonlinear iteration to our discretization is more delicate, since it requires additional linearization of the convection form* $O_h(\mathbf{w},\mathbf{u},\mathbf{v})$ *in the first argument. This remains an item for future investigation.*

3.2.3 Theoretical results

Our first result is a convergence result. To state it, we suppose the solution $(\mathbf{u},\mathbf{b},p,r)$ of (3.1)–(3.2) possesses the smoothness

$$
\begin{aligned}
&(\mathbf{u},p)\in H^{\sigma+1}(\Omega)^d\times H^{\sigma}(\Omega), && (3.8a)\\
&(\mathbf{b},\nabla\times\mathbf{b},r)\in H^{\tau}(\Omega)^d\times H^{\tau}(\Omega)^d\times H^{\tau+1}(\Omega), && (3.8b)
\end{aligned}
$$

for $\sigma,\tau>\frac{1}{2}$.

Remark 5 *The regularity assumption (3.8b) is minimal in the sense that it is satisfied by the strongest singularities of the Maxwell operator in polyhedral domains;*

cf. [26, 27]. Similarly, the regularity (3.8a) holds true for the strongest singularities of the Stokes operator in polyhedral domains; see [2, 29]. In view of these results, we expect (3.8) to be the minimal smoothness of solutions to the MHD system (3.1)-(3.2) in general Lipschitz polyhedra. However, we do not have a full proof of this conjecture.

To discuss the convergence of the finite element formulation (3.6), we introduce the product norms

$$\|(\mathbf{u},\mathbf{b})\|_{V_h\times C_h} = \left(\nu\|\mathbf{u}\|_{1,h}^2 + \kappa\nu_m\|\mathbf{b}\|_{H(\mathrm{curl};\Omega)}^2\right)^{\frac{1}{2}},$$

$$\|(p,r)\|_{Q\times S} = \left(\frac{1}{\nu}\|p\|_{L^2(\Omega)}^2 + \frac{1}{\kappa\nu_m}\|r\|_{H^1(\Omega)}^2\right)^{\frac{1}{2}},$$

Here, the discrete H^1-norm for the hydrodynamic velocity is defined by

$$\|\mathbf{u}\|_{1,h} = \left(\sum_{K\in\mathscr{T}_h}\|\nabla\mathbf{u}\|_{L^2(K)}^2 + \sum_{F\in\mathscr{F}_h} h_F^{-1}\|[\![\mathbf{u}]\!]\|_{L^2(F)}^2\right)^{\frac{1}{2}},$$

and the curl-norm for the magnetic field is defined by

$$\|\mathbf{b}\|_{H(\mathrm{curl};\Omega)} = \left(\|\mathbf{b}\|_{L^2(\Omega)}^2 + \|\nabla\times\mathbf{b}\|_{L^2(\Omega)}^2\right)^{\frac{1}{2}}.$$

Theorem 6 *Let $(\mathbf{u},\mathbf{b},p,r)$ and $(\mathbf{u}_h,\mathbf{b}_h,p_h,r_h)$ be the solutions of (3.1)–(3.2) and (3.6), respectively, obtained on a sequence of quasi-uniform meshes $\{\mathscr{T}_h\}_{h>0}$ of mesh size h. Assume (3.8) and that $\bar{\kappa}\bar{\nu}^{-2}|\!|\!|(\mathbf{f},\mathbf{g})|\!|\!|$ is sufficiently small. Then we have*

$$\lim_{h\to 0}\|(\mathbf{u}-\mathbf{u}_h,\mathbf{b}-\mathbf{b}_h)\|_{V_h\times C_h} = 0, \qquad \lim_{h\to 0}\|(p-p_h,r-r_h)\|_{Q\times S} = 0.$$

Theorem 6 guarantees that the method (3.6) gives correct solutions provided that the (minimal) smoothness assumption (3.8) is satisfied and the data is sufficiently small; see the proof in [44, Section 4.4]. In particular, it ensures convergence in situations where straightforwardly applied nodal elements for the approximation of **b** are not capable of correctly capturing the singular solution components.

Next, we present a-priori error estimates for the two- and three-dimensional cases. To do so, we introduce the function $\ell(d,h)$ given by

$$\ell(d,h)=\begin{cases} h^{-\varepsilon}, & d=2,\\ h^{-\frac{1}{2}}, & d=3. \end{cases}$$

Here, $\varepsilon>0$ is a fixed number. The function $\ell(d,h)$ will indicate the deterioration of convergence rates for both the two-dimensional and three-dimensional cases. We also denote by $C_d>0$ a generic constant independent of h, ν, κ and ν_m, but dependent on the dimension d. In particular, for $d=2$ it depends on ε and might be unbounded as $\varepsilon\to 0$.

Theorem 7 *Let $\Omega\subset\mathbb{R}^d$ ($d=2$ or 3) be a simply-connected Lipschitz polygon with a connected boundary $\partial\Omega$. Under the same assumption as in Theorem 6, we have the following error estimates:*

$$\begin{aligned}
&\|(\mathbf{u}-\mathbf{u}_h,\mathbf{b}-\mathbf{b}_h)\|_{V_h\times C_h}\\
&\quad\le C_d\ell(d,h)h^{\min\{\sigma,\tau,k\}}\left(\nu^{\frac{1}{2}}\|\mathbf{u}\|_{H^{\sigma+1}(\Omega)}+(\kappa\nu_m)^{\frac{1}{2}}\|\mathbf{b}\|_{H^{\tau}(\Omega)}+(\kappa\nu_m)^{\frac{1}{2}}\|\nabla\times\mathbf{b}\|_{H^{\tau}(\Omega)}\right)\\
&\quad+Ch^{\min\{\sigma,\tau,k\}}\left(\nu^{-\frac{1}{2}}\|p\|_{H^{\sigma}(\Omega)}+(\kappa\nu_m)^{-\frac{1}{2}}\|r\|_{H^{\tau+1}(\Omega)}\right),
\end{aligned}$$

and

$$\begin{aligned}
&\|(p-p_h,r-r_h)\|_{Q\times S}\\
&\quad\le C_d\ell(d,h)^2h^{\min\{\sigma,\tau,k\}}\left(\nu^{\frac{1}{2}}\|\mathbf{u}\|_{H^{\sigma+1}(\Omega)}+(\kappa\nu_m)^{\frac{1}{2}}\|\mathbf{b}\|_{H^{\tau}(\Omega)}+(\kappa\nu_m)^{\frac{1}{2}}\|\nabla\times\mathbf{b}\|_{H^{\tau}(\Omega)}\right)\\
&\quad+C_d\ell(d,h)h^{\min\{\sigma,\tau,k\}}\left(\nu^{-\frac{1}{2}}\|p\|_{H^{\sigma}(\Omega)}+(\kappa\nu_m)^{-\frac{1}{2}}\|r\|_{H^{\tau+1}(\Omega)}\right).
\end{aligned}$$

Here, the constants C is independent of h, ν, κ and ν_m.

We provide the proof of Theorem 7 in [44, Section 4.5]. The convergence rates in Theorem 7 are optimal in the mesh size, up to a loss of $\mathscr{O}(h^{\varepsilon})$ for ε arbitrarily small in two dimensions, and up to a loss of $\mathscr{O}(h^{\frac{1}{2}})$ in three dimensions. In addition, in the two-dimensional case, the constant C_d might become unbounded as ε

tends to zero. However, in our numerical experiments this constant is observed to stay bounded. In fact, we observe optimal rates of convergence in all our tests, for both smooth and non-smooth solutions. Full details are given in Section 3.3. This gap between the theoretical and numerical results indicates that the former could be further improved.

For the linearized variant of the MHD system (3.1), our method converges optimally in the mesh size h, as follows from [58, Remark 3.3]. That is, the estimates of Theorem 7 hold true without any loss, both in two and three dimensions. However, there we make stronger smoothness assumptions on the linearized magnetic field. Therefore, this optimality cannot be straightforwardly carried over to the nonlinear setting.

3.3 Numerical results

In this section we present a series of numerical experiments. Our computations have been carried out using MATLAB, with direct linear solvers; iterative solvers will be discussed in Chapter 4. The primary purpose of our experiments is to confirm optimal convergence rates of our method. We start by considering one problem with a smooth solution and a second one with a singular solution. Then, we consider the numerical approximations of two- and three-dimensional Hartmann channel flow and driven cavity flow problems. Finally, we present results for another benchmark problem: MHD flow over a step in two dimensions.

Throughout this section, the lowest order BDM and Nédélec elements are employed and the interior penalty stabilization parameter is $a_0 = 10$. The Picard iteration described in (3.7) is used to solve the nonlinear systems. For all the examples, we solve a Stokes problem and the Maxwell equations, decoupled, to obtain an initial guess. The tolerance for the Picard iterations is chosen as 1e-5.

We test our method on problems with mixed Dirichlet and Neumann boundary conditions in the hydrodynamic variables, even though the analysis has been carried out solely for the Dirichlet case. Throughout this section, Γ_N denotes the Neumann boundary, and Γ_D the Dirichlet boundary. On Neumann boundaries, we specify the value of $(p\underline{\mathbf{I}} - \nu\nabla\mathbf{u})\mathbf{n}$, where $\underline{\mathbf{I}}$ is the identity matrix.

3.3.1 Example 1: two-dimensional problem with a smooth solution

First, we verify the theoretical results stated in Theorems 6 and 7 for a problem with a smooth analytical solution.

We consider the following two-dimensional problem. We set $\Omega = (-1,1)^2$ with $\Gamma_N = \{(1,y) : y \in (-1,1)\}$, $\Gamma_D = \partial\Omega \backslash \Gamma_N$, $\nu = \kappa = 1$, $\nu_m = 1e4$, and choose the source terms $\mathbf{f}$, $\mathbf{g}$ and the boundary conditions so that the analytical solution is of the form

$$\mathbf{u}(x,y) = (y^2, x^2), \qquad p(x,y) = x,$$
$$\mathbf{b}(x,y) = (1-y^2, 1-x^2), \qquad r(x,y) = (1-x^2)(1-y^2).$$

We construct this example with $r \neq 0$ to show the convergence rate in r_h; later examples will feature a divergence-free $\mathbf{g}$ and a vanishing r; cf. Proposition 3.

DOFs in $\mathbf{u}_h/p_h$	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$	l	$\|p-p_h\|_{L^2(\Omega)}$	l
112/32	3.893e-2	–	8.297e-1	–	1.297	–
416/128	1.016e-2	1.94	4.105e-1	1.01	3.734e-1	1.78
1,600/512	2.707e-3	1.91	2.045e-1	1.01	1.293e-1	1.53
6,272/2,048	7.087e-4	1.93	1.021e-1	1.00	5.475e-2	1.24
24,832/8,192	1.813e-4	1.97	5.104e-2	1.00	2.597e-2	1.08
98,816/32,768	4.578e-5	1.99	2.552e-2	1.00	1.281e-2	1.02

Table 3.1: Example 1. Convergence of $\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$, $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, and $\|p-p_h\|_{L^2(\Omega)}$.

DOFs in $\mathbf{b}_h/r_h$	$\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$	l
56/25	4.720e-1	–	9.431e-1	–
208/81	2.358e-1	1.00	4.714e-1	1.00
800/289	1.179e-1	1.00	2.357e-1	1.00
3,136/1,089	5.893e-2	1.00	1.179e-1	1.00
12,416/4,225	2.946e-2	1.00	5.893e-2	1.00
49,408/16,641	1.473e-2	1.00	2.946e-2	1.00

Table 3.2: Example 1. Convergence of $\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$ and $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$.

DOFs in $\mathbf{b}_h/r_h$	$\|r-r_h\|_{L^2(\Omega)}$	l	$\|\nabla(r-r_h)\|_{L^2(\Omega)}$	l
56/25	1.673e-1	–	9.391e-1	–
208/81	4.433e-2	1.92	4.824e-1	0.96
800/289	1.125e-2	1.98	2.429e-1	0.99
3,136/1,089	2.822e-3	1.99	1.216e-1	1.00
12,416/4,225	7.062e-4	2.00	6.085e-2	1.00
49,408/16,641	1.766e-4	2.00	3.043e-2	1.00

Table 3.3: Example 1. Convergence of $\|r-r_h\|_{L^2(\Omega)}$ and $\|\nabla(r-r_h)\|_{L^2(\Omega)}$.

In Tables 3.1–3.3, we investigate the asymptotic rates of convergence of the errors in the approximations of the hydrodynamic and magnetic variables; here, l denotes the experimental rate of convergence. We observe that $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, $\|p-p_h\|_{L^2(\Omega)}$, $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$ and $\|\nabla(r-r_h)\|_{L^2(\Omega)}$ converge to zero as the mesh is refined, in accordance with Theorem 6. The rate of convergence is $\mathscr{O}(h)$. Notice that we obtain the optimal rate in this numerical experiment, even though Theorem 7 predicts a suboptimal rate with a loss of $\mathscr{O}(h^{\varepsilon})$. Additionally, $\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$ and $\|r-r_h\|_{L^2(\Omega)}$ converge at rate $\mathscr{O}(h^2)$ as h tends to zero, which is also optimal.

In Figure 3.3 we show the convergence history of the Picard iterations for the grid sequence considered in this example. The plot depicts the number of iterations against the differences between consecutive iterates corresponding to the approximated vector coefficients, measured in a normalized discrete 2-norm and labeled as 'Tolerance' in the plot. As expected, convergence is linear and the iteration count is fairly insensitive to the size of the grid. A very similar behavior has been observed in all of our other experiments, in 2D as well as in 3D.

3.3.2 Example 2: two-dimensional problem with a singular solution

In order to verify the capability of the proposed method to capture singularities in two dimensions, we consider a problem in the L-shaped domain $\Omega=(-1,1)^2\setminus([0,1)\times(-1,0])$ with $\Gamma_N=\{(1,y):y\in(0,1)\}$, $\Gamma_D=\partial\Omega\backslash\Gamma_N$, and set $\nu=\kappa=1$, $\nu_m=1\text{e}4$. We choose the forcing terms and the boundary conditions such that the analytic solution is given by the strongest corner singularities for the underlying elliptic operators. In polar coordinates (ρ,θ), the hydrodynamic solution compo-

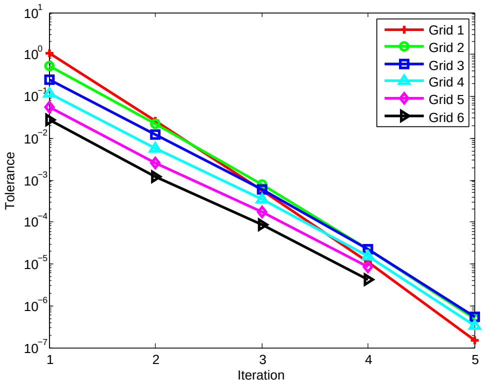


Figure 3.3: Example 1. Convergence history of the Picard iteration for the grid sequence defined in Tables 3.1–3.3.

nents $\mathbf{u}$ and p are then given by

$$\mathbf{u}(\rho,\theta) = \begin{bmatrix} \rho^{\lambda}((1+\lambda)\sin(\theta)\xi(\theta)+\cos(\theta)\xi'(\theta)) \\ \rho^{\lambda}(-(1+\lambda)\cos(\theta)\xi(\theta)+\sin(\theta)\xi'(\theta)) \end{bmatrix},$$

$$p(\rho,\theta) = -\rho^{\lambda-1}((1+\lambda)^2\xi'(\theta)+\xi'''(\theta))/(1-\lambda),$$

where

$$\xi(\theta) = \sin((1+\lambda)\theta)\cos(\lambda\omega)/(1+\lambda)-\cos((1+\lambda)\theta) \\ -\sin((1-\lambda)\theta)\cos(\lambda\omega)/(1-\lambda)+\cos((1-\lambda)\theta),$$

$\omega = \frac{3}{2}\pi$ and $\lambda \approx 0.54448373678246$. The magnetic pair $(\mathbf{b}, r)$ is given by

$$\mathbf{b}(\rho,\theta) = \nabla(\rho^{2/3}\sin(2/3\theta)), \qquad r(\rho,\theta) \equiv 0.$$

For this example, we have that $(\mathbf{u}, p) \in H^{1+\lambda}(\Omega)^2 \times H^{\lambda}(\Omega)$ and $\mathbf{b} \in H^{2/3}(\Omega)^2$. Note that straightforwardly applied nodal elements cannot correctly resolve the magnetic field. In Tables 3.4–3.5, we investigate the asymptotic rates of convergence of the errors in the approximations of the hydrodynamic and magnetic variables. Again, we observe that the discrete solution converges to the exact one as the mesh size h approaches zero, in accordance with Theorem 6. The results show full agreement with the optimal rates for $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$ and $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$. For the pressure, we also see that the rate for $\|p-p_h\|_{L^2(\Omega)}$ is approaching the optimal rate, albeit more slowly. Additionally, we observe the L^2-norm of r is zero because $\mathbf{g}$ is divergence-free, in accordance with Proposition 3.

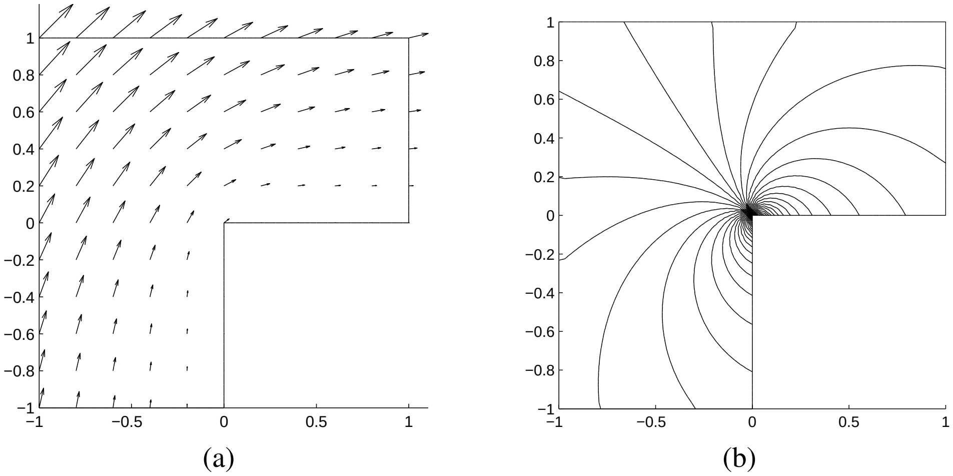
In Figures 3.4–3.5, we show the solution computed on the finest mesh with 24,576 elements; the total number of degrees of freedom employed in the finite element space $\mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$ is 148,481. The results show that our solution captures the strongest corner singularities and are comparable to the results in [58].

DOFs in $\mathbf{u}_h/p_h$	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$	l	$\|p-p_h\|_{L^2(\Omega)}$	l
88/24	2.159e-1	–	2.468	–	15.91	–
320/96	1.781e-1	0.28	1.880	0.39	9.328	0.77
1,216/384	1.204e-1	0.56	1.368	0.46	5.387	0.79
4,736/1,536	6.816e-1	0.82	0.9588	0.51	3.301	0.71
18,688/6,144	3.490e-2	0.97	0.6627	0.53	2.124	0.64
74,240/24,576	1.705e-2	1.03	0.4559	0.54	1.408	0.59

Table 3.4: Example 2. Convergence of $\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$, $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, and $\|p-p_h\|_{L^2(\Omega)}$.

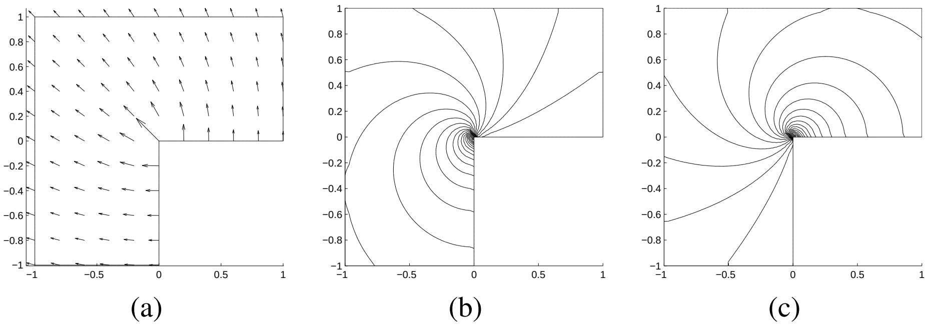
DOFs in $\mathbf{b}_h/r_h$	$\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$	l	$\|r_h\|_{L^2(\Omega)}$
44/21	2.796e-1	–	2.796e-1	–	2.162e-12
160/65	1.814e-1	0.62	1.814e-1	0.62	6.188e-12
608/225	1.169e-1	0.63	1.169e-1	0.63	2.289e-11
2,368/833	7.473e-2	0.65	7.473e-2	0.65	4.260e-11
9,344/3,201	4.754e-2	0.65	4.754e-2	0.65	1.406e-10
37,120/12,545	3.013e-2	0.66	3.013e-2	0.66	3.018e-10

Table 3.5: Example 2. Convergence of $\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$, $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$, and $\|r_h\|_{L^2(\Omega)}$.



(a) (b)

Figure 3.4: Example 2. Numerical approximations of (a) velocity; (b) pressure contours.



(a) (b) (c)

Figure 3.5: Example 2. Numerical approximations of (a) magnetic field; (b) contours of the first component of the magnetic field; (c) contours of the second component of the magnetic field.

3.3.3 Example 3: two-dimensional Hartmann flow

Next, we consider Hartmann channel flow problems in two and three dimensions; cf. [42]. In these examples, we denote by Ha the Hartmann number, which is defined as $\mathrm{Ha} = \sqrt{\frac{\kappa}{\nu\nu_m}}$.

Consider the two-dimensional Hartmann flow problem, which involves a steady unidirectional flow in the channel $\Omega = (0,10) \times (-1,1)$ under the influence of the constant transverse magnetic field $\mathbf{b}_D = (0,1)$. The MHD solution then takes the form:

$$\begin{aligned} \mathbf{u}(x,y) &= (u(y),0), \quad p(x,y) = -Gx + p_0(y), \\ \mathbf{b}(x,y) &= (b(y),1), \quad r(x,y) \equiv 0. \end{aligned} \tag{3.9}$$

We impose the following boundary conditions:

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{on } y = \pm 1, \\ (p\underline{\mathbf{I}} - \nu\nabla\mathbf{u})\mathbf{n} &= p_N\mathbf{n} && \text{on } x = 0 \text{ and } x = 10, \\ \mathbf{n} \times \mathbf{b} &= \mathbf{n} \times \mathbf{b}_D && \text{on } \partial\Omega, \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where

$$\begin{aligned} p_N(x,y) &= p(x,y) \\ &= -Gx - \frac{G^2}{2\kappa}\left(\frac{\sinh(y\mathrm{Ha})}{\sinh(\mathrm{Ha})} - y\right)^2. \end{aligned}$$

The exact solution is given by (3.9) with

$$\begin{aligned} u(y) &= \frac{G}{\nu\mathrm{Ha}\tanh(\mathrm{Ha})}\left(1 - \frac{\cosh(y\mathrm{Ha})}{\cosh(\mathrm{Ha})}\right), \\ b(y) &= \frac{G}{\kappa}\left(\frac{\sinh(y\mathrm{Ha})}{\sinh(\mathrm{Ha})} - y\right), \\ p_0(y) &= -\frac{G^2}{2\kappa}\left(\frac{\sinh(y\mathrm{Ha})}{\sinh(\mathrm{Ha})} - y\right)^2. \end{aligned}$$

We note that $p_0(y)$ and $-\frac{\kappa b(y)^2}{2}$ are the same up to an additive constant.

In Tables 3.6–3.7 and Figures 3.6–3.7, we set $\nu = \kappa = 1$, $\nu_m = 1\text{e}4$, and $G = 10$. We observe that $r_h \equiv 0$, as predicted in Proposition 3, and $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, $\|p-p_h\|_{L^2(\Omega)}$ and $\|\mathbf{b}-\mathbf{b}_h\|_{H(\mathrm{curl};\Omega)}$ converge to zero at the optimal rate $\mathscr{O}(h)$ as the mesh is refined. Moreover, we note that the L^2-norms of the errors in the approximations of $\mathbf{u}$, $\mathbf{b}$ and p tend to zero optimally as well.

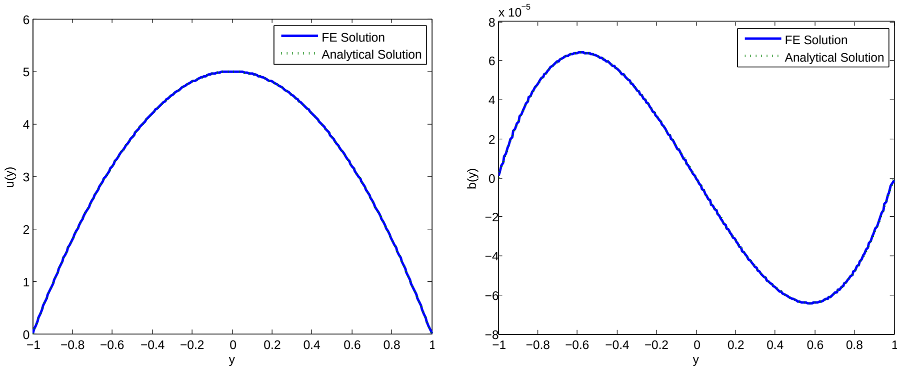
In Figures 3.6–3.7 we show the slices and the vector fields of the solution computed on the mesh with 32,768 elements; the total number of degrees of freedom employed in the finite element space $\mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$ is 197,633. In order to show the directions of vectors, in Figure 3.7(b) and later figures, $\mathbf{b}$ is normalized such that the largest magnitude of each component is 1 in the computational domain. The computed and analytical solutions of the first components in the velocity and magnetic fields are virtually indistinguishable; see Figure 3.6.

DOFs in $\mathbf{u}_h/p_h$	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$	l	$\|p-p_h\|_{L^2(\Omega)}$	l
416/128	2.028e-1	–	3.215	–	13.97	–
1,600/512	5.169e-2	1.97	1.611	1.00	6.986	1.00
6,272/2,048	1.306e-2	1.99	0.8061	1.00	3.493	1.00
24,832/8,192	3.282e-3	1.99	0.4033	1.00	1.747	1.00
98,816/32,768	8.227e-4	2.00	0.2017	1.00	0.8734	1.00

Table 3.6: Example 3. Convergence of $\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$, $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, and $\|p-p_h\|_{L^2(\Omega)}$.

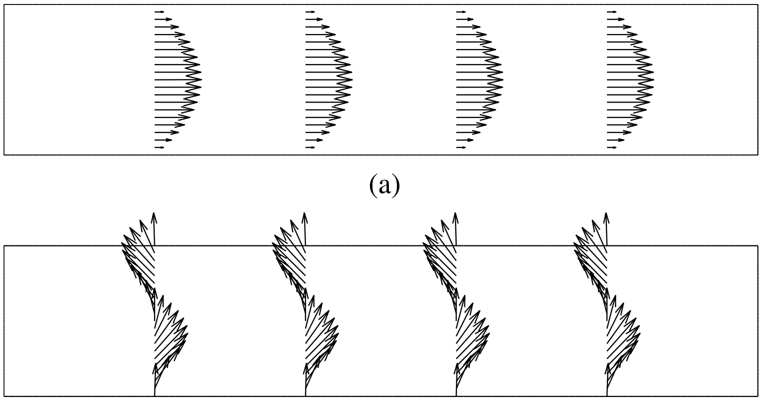
DOFs in $\mathbf{b}_h/r_h$	$\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{b}-\mathbf{b}_h\|_{H(\mathrm{curl};\Omega)}$	l	$\|r_h\|_{L^2(\Omega)}$
208/81	1.679e-4	–	2.259e-4	–	3.868e-12
800/289	8.605e-5	0.96	1.148e-4	0.98	1.746e-11
3,136/1,089	4.328e-5	0.99	5.761e-4	0.99	3.627e-11
12,416/4,225	2.167e-5	1.00	2.883e-5	1.00	9.424e-11
49,408/16,641	1.084e-5	1.00	1.442e-5	1.00	2.401e-10

Table 3.7: Example 3. Convergence of $\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$, $\|\mathbf{b}-\mathbf{b}_h\|_{H(\mathrm{curl};\Omega)}$, and $\|r_h\|_{L^2(\Omega)}$.



(a) (b)

Figure 3.6: Example 3. Slices along $x = 5$, $-1 \leq y \leq 1$: (a) velocity component $u(y)$; (b) magnetic component $b(y)$.



(a)

(b)

Figure 3.7: Example 3. Numerical approximations of (a) velocity; (b) normalized magnetic field.

3.3.4 Example 4: three-dimensional Hartmann flow

In this example, we consider the three-dimensional unidirectional flow in the rectangular duct given by $\Omega = (0, L) \times (-y_0, y_0) \times (-z_0, z_0)$ with $y_0, z_0 \ll L$ under the influence of the constant transverse magnetic field $\mathbf{b}_D = (0,1,0)$. We take $\mathbf{f} = \mathbf{g} = \mathbf{0}$ and consider solutions of the form

$$\mathbf{u}(x,y,z) = (u(y,z),0,0), \qquad p(x,y,z) = -Gx + p_0(y,z),$$
$$\mathbf{b}(x,y,z) = (b(y,z),1,0), \qquad r(x,y,z) \equiv 0.$$

We enforce the boundary conditions

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{for } y = \pm y_0 \text{ and } z = \pm z_0, \\ (p\underline{\mathbf{I}} - \nu\nabla\mathbf{u})\mathbf{n} &= p_N\mathbf{n} && \text{for } x = 0 \text{ and } x = L, \\ \mathbf{n}\times\mathbf{b} &= \mathbf{n}\times\mathbf{b}_D && \text{on } \partial\Omega, \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

with $p_N(x,y,z) = -Gx - \frac{\kappa b(y,z)^2}{2} + 10$. The function $b(y,z)$ is given by the Fourier series

$$b(y,z) = \sum_{n=0}^{\infty} b_n(y)\cos(\lambda_n z),$$

where

$$\begin{aligned} \lambda_n &= \frac{(2n+1)\pi}{2z_0}, \\ b_n(y) &= \frac{\nu}{\kappa}\left(A_n\frac{\lambda_n^2 - p_1^2}{p_1}\sinh(p_1 y) + B_n\frac{\lambda_n^2 - p_2^2}{p_2}\sinh(p_2 y)\right), \\ p_{1,2}^2 &= \lambda_n^2 + \mathrm{Ha}^2/2 \pm \mathrm{Ha}\sqrt{\lambda_n^2 + \mathrm{Ha}^2/4}, \\ A_n &= \frac{-p_1(\lambda_n^2 - p_2^2)}{\Delta_n}u_n(y_0)\sinh(p_2 y_0), \\ B_n &= \frac{p_2(\lambda_n^2 - p_1^2)}{\Delta_n}u_n(y_0)\sinh(p_1 y_0), \\ \Delta_n &= p_2(\lambda_n^2 - p_1^2)\sinh(p_1 y_0)\cosh(p_2 y_0) - p_1(\lambda_n^2 - p_2^2)\sinh(p_2 y_0)\cosh(p_1 y_0), \\ u_n(y_0) &= \frac{-2G}{\nu\lambda_n^3 z_0}\sin(\lambda_n z_0). \end{aligned}$$

DOFs in $\mathbf{u}_h/p_h$	$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$	l	$\|p-p_h\|_{L^2(\Omega)}$	l
360/48	3.959e-1	–	1.829	–	30.89	–
2,592/384	1.320e-1	1.58	0.9561	0.94	8.194	1.91
19,584/3,072	3.609e-2	1.87	0.4903	0.96	2.837	1.53
152,064/24,576	9.590e-3	1.91	0.2484	0.98	1.091	1.38

Table 3.8: Example 4. Convergence of $\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$, $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$, and $\|p-p_h\|_{L^2(\Omega)}$.

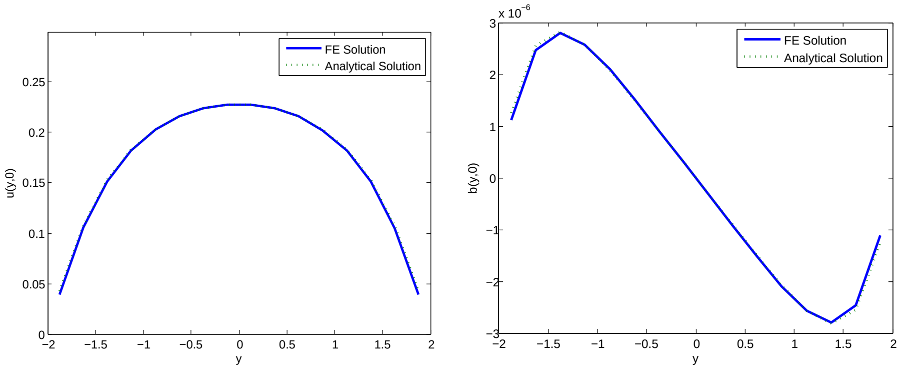
DOFs in $\mathbf{b}_h/r_h$	$\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$	l	$\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$	l	$\|r_h\|_{L^2(\Omega)}$
98/27	1.850e-5	–	3.219e-5	–	9.855e-12
604/125	1.565e-5	0.24	2.579e-5	0.32	1.013e-10
4,184/729	8.592e-6	0.86	1.464e-5	0.82	4.098e-10
31,024/4,913	4.411e-6	0.96	7.543e-6	0.96	1.795e-9

Table 3.9: Example 4. Convergence of $\|\mathbf{b}-\mathbf{b}_h\|_{L^2(\Omega)}$, $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$, and $\|r_h\|_{L^2(\Omega)}$.

The functions $u(y,z)$ and $p_0(y,z)$ can be also expressed by Fourier series; for details; see [40]. In fact, $p_0(y,z)$ and $-\frac{\kappa b(y,z)^2}{2}$ are identical up to an additive constant. Note also that $p(x,y,z) = p_N(x,y,z)$.

In our tests, we set $L = 10$, $y_0 = 2$, $z_0 = 1$, $\nu = \kappa = 1$, $\nu_m = 1e4$ and $G = 0.5$. In Tables 3.8–3.9, we investigate the asymptotic rates of convergence of the errors in the approximations of the hydrodynamic and magnetic variables. Again, we observe that the finite element solution converges to the exact solution as the mesh size h approaches zero, in accordance with Theorem 6. We observe the results show good agreement with the optimal rates for $\|\mathbf{u}-\mathbf{u}_h\|_{1,h}$ and $\|\mathbf{b}-\mathbf{b}_h\|_{H(\text{curl};\Omega)}$. For the pressure, we also see that the rate for $\|p-p_h\|_{L^2(\Omega)}$ is approaching the optimal rate, although more slowly. Additionally, we observe the L^2-norm of r is zero because $\mathbf{g}$ is divergence-free, in accordance with Proposition 3.

In Figures 3.8–3.9 we show the slices and the vector fields of the solution computed on a uniform tetrahedral mesh of 24,576 elements; this results in a total of 212,577 degrees of freedom in the finite element space $\mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$. We observe that the computed and analytical solutions are in good agreement on this relatively coarse mesh; see Figure 3.8.



(a) (b)

Figure 3.8: Example 4. Slices along $x = 5$, $-2 \leq y \leq 2$, and $z = 0$: (a) velocity component $u(y,0)$; (b) magnetic component $b(y,0)$.

3.3.5 Example 5: two-dimensional driven cavity flow

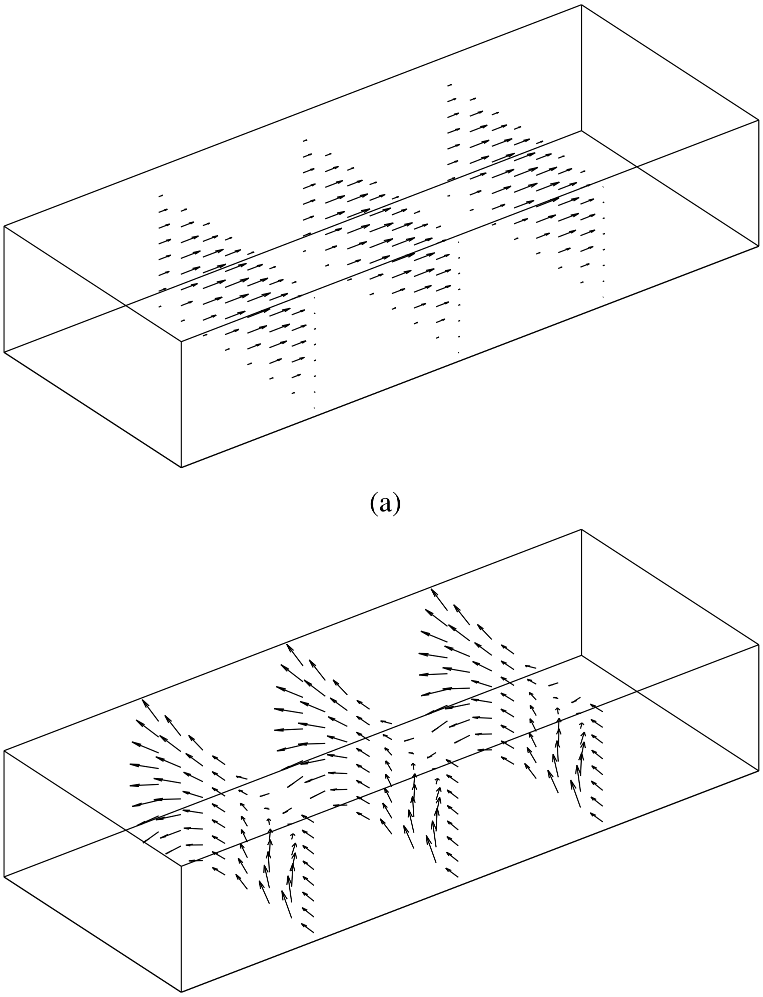
Let us consider a classic test problem used in fluid dynamics, known as driven-cavity flow. It is a model of the flow in a cavity with the lid moving in one direction; cf. [37, Chapter 5.1.3] and [77].

In this example, we consider the two-dimensional domain $\Omega = (-1,1)^2$ with $\Gamma_D = \partial\Omega$, and set the source terms to be zero. The boundary conditions are prescribed as follows:

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{on } x = \pm 1 \text{ and } y = -1, \\ \mathbf{u} &= (1,0) && \text{on } y = 1, \\ \mathbf{n} \times \mathbf{b} &= \mathbf{n} \times \mathbf{b}_D && \text{on } \partial\Omega, \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $\mathbf{b}_D = (1,0)$.

We set $\nu =$ 1e-2, $\nu_m =$ 1e5, $\kappa =$ 1e5, which simulates liquid metal type flows. Figures 3.10–3.11 show the solution computed on a mesh with 8,192 elements

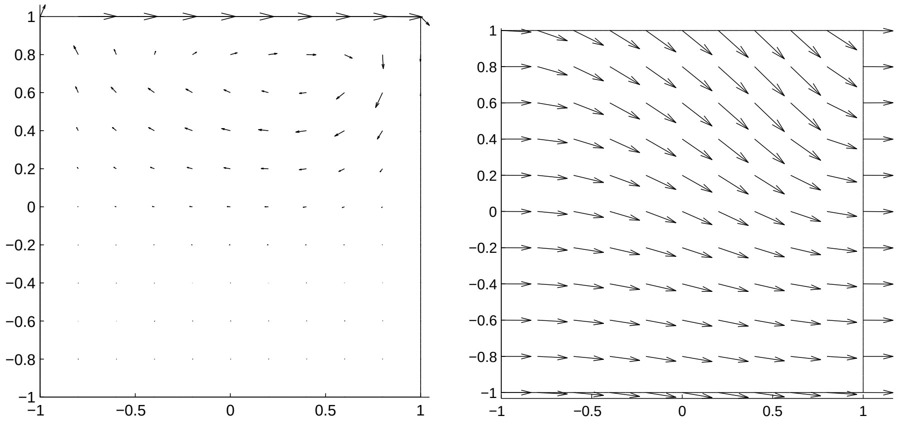


(a)

(b)

Figure 3.9: Example 4. Numerical approximations of (a) velocity; (b) normalized magnetic field.

and 49,665 degrees of freedom. Figure 3.10(a) shows the circulation created by the moving lid; Figure 3.10(b) shows that the magnetic field changes direction due to the coupling effect. Figure 3.11(a) depicts the boundary layer formation in terms of the first component of the velocity. Streamlines for the velocity field are displayed in Figure 3.11(b). The computed solution agrees with the solution in the literature [77].



(a)

(b)

Figure 3.10: Example 5. Numerical approximations of (a) velocity; (b) normalized magnetic field.

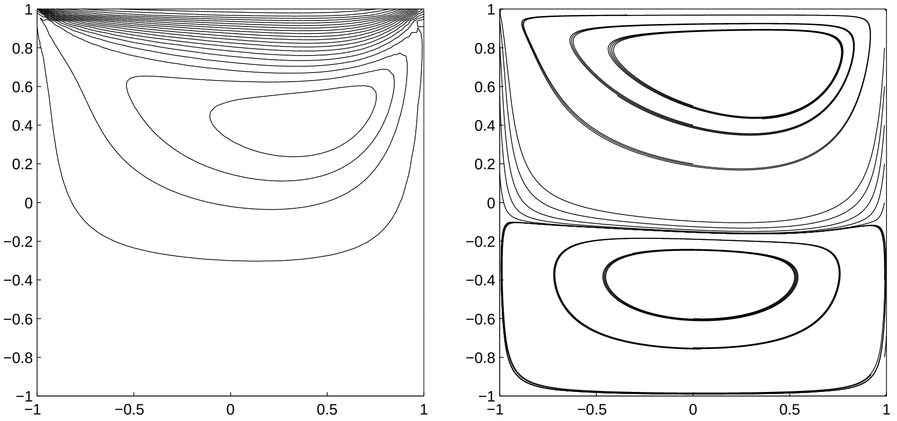
3.3.6 Example 6: three-dimensional driven cavity flow

The problem we consider is the three-dimensional driven cavity flow in the domain $\Omega = (-1,1)^3$ with $\Gamma_D = \partial\Omega$. The source terms are set to be zero. The boundary conditions are prescribed as follows:

$$\begin{aligned} \mathbf{u} &= \mathbf{0} && \text{on } x = \pm 1,\, y = \pm 1 \text{ and } z = -1, \\ \mathbf{u} &= (1,0,0) && \text{on } z = 1, \\ \mathbf{n} \times \mathbf{b} &= \mathbf{n} \times \mathbf{b}_D && \text{on } \partial\Omega, \\ r &= 0 && \text{on } \partial\Omega, \end{aligned}$$

where $\mathbf{b}_D = (1,0,0)$.

We set ν = 1e-2, ν_m = 1e5, κ = 1e5 and obtain Figure 3.12 on a uniform tetrahedral mesh comprising 24,576 elements; this results in a total of 212,577 degrees of freedom. The flow vectors on slices demonstrate a similar behavior to



(a) (b)

Figure 3.11: Example 5. Numerical approximations of (a) contours of first velocity components; (b) streamlines of velocity.

the two-dimensional scenario in Section 3.3.5; see Figure 3.10.

3.3.7 Example 7: two-dimensional MHD flow over a step

The example we present here is another classical problem of a flow over a step under a transverse magnetic field; cf. [41]. The magnetic field tends to damp the vortex of the fluid after the step.

The domain is $\Omega = (-0.25, 0.75) \times (-0.125, 0.125) \setminus (-0.25, 0] \times (-0.125, 0]$, with $\Gamma_N = \{(0.75, y) : y \in (-0.125, 0.125)\}$ and $\Gamma_D = \partial\Omega \setminus \Gamma_N$. We set $\mathbf{f} = \mathbf{g} = \mathbf{0}$, and choose ν = 1e-2, ν_m = 1e5, κ = 2.5e4. The boundary data are given by

$$\begin{aligned}
\mathbf{u} &= \mathbf{0} && \text{on } y = \pm 0.125 \text{ and } \{(x,0) : x \in (-0.25, 0)\},\\
\mathbf{u} &= \mathbf{0} && \text{on } \{(0,y)\colon y \in (-0.125, 0)\},\\
\mathbf{u} &= (-25.6y(y-0.125), 0) && \text{on } x = -0.25,\\
(p\underline{\mathbf{I}} - \nu\nabla\mathbf{u})\mathbf{n} &= p_N\mathbf{n} && \text{on } x = 0.75,\\
\mathbf{n} \times \mathbf{b} &= \mathbf{n} \times \mathbf{b}_D && \text{on } \partial\Omega,\\
r &= 0 && \text{on } \partial\Omega,
\end{aligned}$$

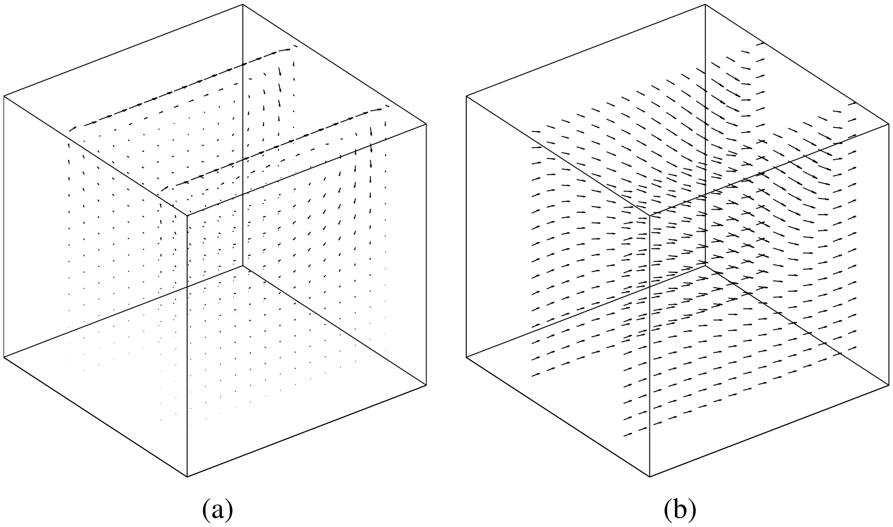


Figure 3.12: Example 6. Numerical approximations of (a) velocity; (b) normalized magnetic field.

where $p_N = 0$ and $\mathbf{b}_D = (0,1)$.

Figures 3.13–3.14 show the solution computed on a mesh with 7,168 elements and 43,649 degrees of freedom. It is evident from Figure 3.13 that the flow field is correctly captured; the magnetic field changes directions due to the coupling effect; the pressure drops behind the step. Figure 3.14 shows the velocity field in terms of stream lines. The recirculation after the step decreases as the coupling coefficient κ increases. We observe that our numerical method reproduces this damping effect without any oscillation in the numerical solution. The computed solutions agree with the solutions in the literature [25, 41].

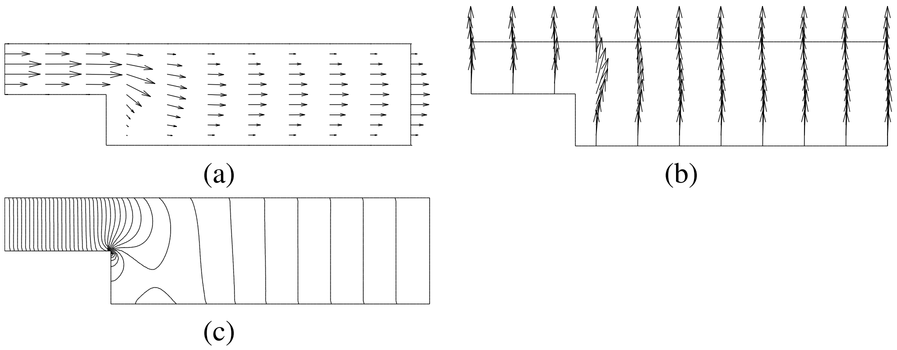
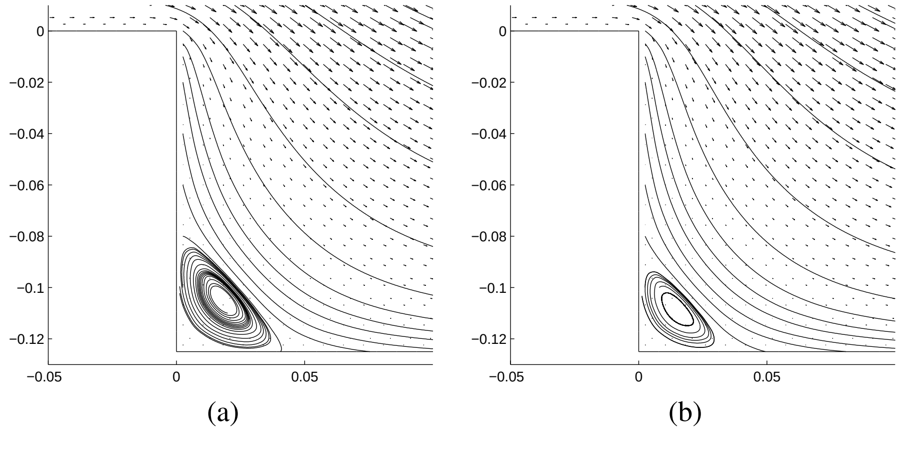

(a) (b)
(c)

Figure 3.13: Example 7. Numerical approximations of (a) velocity; (b) normalized magnetic field; (c) pressure contours.



(a) (b)

Figure 3.14: Example 7. Velocity flow vectors and streamlines zoomed in behind the step for (a) κ=2.5e4; (b) κ=1e5.

Chapter 4

Solution of the discretized incompressible MHD equations

In this chapter, we present preliminary results for preconditioning techniques for the discretized MHD equations in mixed form presented in Chapter 3. The MHD system includes the Maxwell sub-system (discussed in Chapter 2), the Navier-Stokes sub-system (mentioned in Chapter 1) and the coupling terms. A considerable effort has been made to develop efficient and robust solution algorithms for the Maxwell equations and the incompressible Navier-Stokes equations. We present a few preconditioning ideas for the discretized MHD equations, which are based on combining preconditioners for the sub-systems. We provide some eigenvalue analysis for our approach and show that many of the eigenvalues are tightly clustered. Numerical results show that the resulting scheme is rapidly convergent, and is reasonably scalable.

This chapter is structured as follows. First, we discuss related work in Section 4.1. We propose a few ideas of preconditioning in Section 4.2 and compare their performance in the numerical experiments of Section 4.3.

4.1 Background and related work

To transform the Picard iteration (3.7) into matrix form, let $\langle \varphi_j \rangle_{j=1}^{\hat{n}}$ and $\langle \zeta_i \rangle_{i=1}^{\hat{m}}$ be finite element bases for the spaces $\mathbf{V}_h$ and Q_h respectively:

$$\mathbf{V}_h = \operatorname{span}\langle \varphi_j \rangle_{j=1}^{\hat{n}}, \qquad Q_h = \operatorname{span}\langle \zeta_i \rangle_{i=1}^{\hat{m}}. \tag{4.1}$$

Recall from (2.7) that $\langle \psi_j \rangle_{j=1}^{n}$ and $\langle \phi_i \rangle_{i=1}^{m}$ are finite element bases for the spaces $\mathbf{C}_h$ and S_h respectively.

We identify finite element functions $(\mathbf{u}_h^k, \mathbf{b}_h^k, p_h^k, r_h^k) \in \mathbf{V}_h \times \mathbf{C}_h \times Q_h \times S_h$ with their coefficient vectors $u = (u_1, \ldots, u_{\hat{n}}) \in \mathbb{R}^{\hat{n}}$, $b = (b_1, \ldots, b_n) \in \mathbb{R}^n$, $p = (p_1, \ldots, p_{\hat{m}}) \in \mathbb{R}^{\hat{m}}$, and $r = (r_1, \ldots, r_m) \in \mathbb{R}^m$, with respect to the bases (2.7) and (4.1). The solution of (3.7) can now be computed by solving the linear system

$$\mathscr{K} x \equiv \begin{pmatrix} A+O & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix} = \begin{pmatrix} f \\ 0 \\ g \\ 0 \end{pmatrix}. \tag{4.2}$$

Here, A, B, C, D, M and O are matrices associated the forms $A_h(\mathbf{u}_h^k, \mathbf{v})$, $B(\mathbf{u}_h^k, q)$, $C(\mathbf{b}_h^{k-1}, \mathbf{v}, \mathbf{b}_h^k)$, $D(\mathbf{b}_h^k, s)$, $M(\mathbf{b}_h^k, \mathbf{c})$ and $O_h(\mathbf{u}_h^{k-1}, \mathbf{u}_h^k, \mathbf{v})$ defined in Section 3.2. Vectors f and g are associated with the forms $(\mathbf{f}, \mathbf{v})_\Omega$ and $(\mathbf{g}, \mathbf{c})_\Omega$.

The system (4.2) is in a generalized saddle point form. It is unsymmetric due to the convection term O (note also the skew-symmetric term C, which can be easily symmetrized by manipulating signs). We are interested in solving (4.2) iteratively using preconditioned GMRES. To the best of our knowledge, preconditioned approaches for solving (4.2) have not been previously considered in the literature. We denote the coefficient matrix for the Navier-Stokes sub-problem by

$$\mathscr{K}_N = \begin{pmatrix} F & B^T \\ B & 0 \end{pmatrix}, \tag{4.3}$$

where $F = A + O$, and the coefficient matrix for the Maxwell sub-system is

$$\mathscr{K}_M = \begin{pmatrix} M & D^T \\ D & 0 \end{pmatrix}. \tag{4.4}$$

We have already presented preconditioning techniques for the time-harmonic Maxwell equations in Chapter 2. Thus, we discuss some related work for preconditioning the Navier-Stokes system and hence focus on solution algorithms for the linear system associated with the coefficient matrix (4.3).

Consider the *LDU* block factorization of the coefficient matrix,

$$\mathscr{K}_N = \begin{pmatrix} I & 0 \\ BF^{-1} & I \end{pmatrix} \begin{pmatrix} F & 0 \\ 0 & -S \end{pmatrix} \begin{pmatrix} I & F^{-1}B^T \\ 0 & I \end{pmatrix},$$

where

$$S = BF^{-1}B^T$$

is the Schur complement (of F in $\mathscr{K}_N$). We will investigate a preconditioning operator of the form

$$\mathscr{M} \equiv \begin{pmatrix} F & 0 \\ 0 & S \end{pmatrix}. \tag{4.5}$$

The preconditioned linear system $\mathscr{M}^{-1}\mathscr{K}_N$ has exactly three eigenvalues. Thus GMRES will converge in three iterations; see [75]. However, this idealized situation is impractical, since the preconditioning operation requires the action of the inverse of S. Effective approximate block preconditioners are often based on a good approximation to S (e.g., [11, 12, 19, 37, 64]).

Since the linear system is nonsymmetric, we could also consider a block triangular version of (4.5)

$$\mathscr{M} \equiv \begin{pmatrix} F & B^T \\ 0 & -M_S \end{pmatrix}, \tag{4.6}$$

where M_S is an approximation to S. Krylov subspace iteration with the block triangular preconditioner requires approximately half as many as steps as with the block diagonal preconditioner; see [35]. A choice for M_S is the pressure mass matrix K. If F only consists of the discrete Laplacian operator (i.e., the convection term O

is negligible), with the choice $M_S = \frac{1}{\nu}K$, the preconditioner $\mathscr{M}$ makes it possible to obtain mesh-independent convergence rates; see [35, 66]. However, it does not take into account the effects of convection on the Schur complement operator S, and convergence rates deteriorate as the convection increases. Pressure convection-diffusion preconditioners are proposed in [64, 92]. These preconditioners are more faithful approximations to the Schur complement matrix. Numerous theoretical and numerical studies have demonstrated mesh independent convergence on several problems and the overall efficacy of this method. A drawback, however, is that it requires the convection-diffusion operation projected onto the discrete pressure space. As an alternative, the least-squares commutator preconditioner is proposed in [36]. This method automatically generates an approximation to S by solving the normal equation associated with a certain least-squares problem derived from the commutator. A comparison and summary of convergence results for the pressure convection-diffusion preconditioner and the least-squares commutator preconditioner is given in [78].

In this chapter, we investigate efficient preconditioners for (4.2). We consider a few different approaches and propose a preconditioner combining the least-squares commutator preconditioner for the Navier-Stokes sub-system and the block diagonal preconditioner in (2.12) for the Maxwell sub-system. We have some preliminary eigenvalue analysis for this approach. Our numerical results show that this approach performs quite well, though it is not independent of the mesh size.

4.2 The solver

The MHD problem combines the Maxwell sub-system and the Navier-Stokes sub-system with coupling terms. We first discuss ad-hoc preconditioners for these sub-systems. Then, we propose a few ideas of combining them for the discretized MHD problem.

4.2.1 Some ideas for preconditioning

For the Maxwell sub-system, the resulting linear system is associated with the coefficient matrix (4.4). Note this corresponds to the discretized time-harmonic Maxwell equations in (2.2) with $\varepsilon_r = 1$, $\mu_r = \frac{1}{\kappa \nu_m}$ and $k = 0$. We can apply the preconditioner defined in (2.12), which is

$$\mathscr{N} \equiv \begin{pmatrix} M+X & 0 \\ 0 & L \end{pmatrix}. \tag{4.7}$$

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

$$L_{i,j} = \int_\Omega \nabla\phi_j \cdot \nabla\phi_i \, d\mathbf{x},$$

and $X = (X_{i,j})_{i,j=1}^{n} \in \mathbb{R}^{n\times n}$ is the mass matrix on $\mathbf{C}_h$, defined as

$$X_{i,j} = \int_\Omega \psi_j \cdot \psi_i \, d\mathbf{x}.$$

For the Navier-Stokes sub-system with the coefficient matrix (4.3), we apply the least-squares commutator preconditioner [36, 37]. This approach is based on the assumption that

$$BF^{-1}B^T \approx (BQ^{-1}B^T)(BQ^{-1}FQ^{-1}B^T)^{-1}(BQ^{-1}B^T),$$

where $Q = (Q_{i,j})_{i,j=1}^{\hat{n}} \in \mathbb{R}^{\hat{n}\times\hat{n}}$ is the mass matrix on $\mathbf{V}_h$ and is defined as

$$Q_{i,j} = \int_\Omega \varphi_j \cdot \varphi_i \, d\mathbf{x}.$$

In the lowest order case, the mass matrix Q is diagonal; for higher order elements, it is block-diagonal. In the latter case, if necessary Q can be replaced by $\hat{Q} = \mathrm{diag}(Q)$ to enhance sparsity; see [37, Remark 6.4]. This would give the following approximation to the Schur complement:

$$M_S = (B\hat{Q}^{-1}B^T)(B\hat{Q}^{-1}F\hat{Q}^{-1}B^T)^{-1}(B\hat{Q}^{-1}B^T). \tag{4.8}$$

The least-squares commutator preconditioner has the form in (4.6) with (4.8) as the (2,2) block. Applying this preconditioner involves the inverse

$$M_S^{-1} = (B\hat{Q}^{-1}B^T)^{-1}(B\hat{Q}^{-1}F\hat{Q}^{-1}B^T)(B\hat{Q}^{-1}B^T)^{-1},$$

and so, it involves two Poisson solves, for which many efficient solution methods are available.

Next, we discuss three different approaches to combining the preconditioners for the sub-systems.

(1) *Combining preconditioners for the sub-systems without coupling.* First we consider applying the ad-hoc preconditioners for the sub-systems directly. We treat the coupling term C explicitly in the Picard iteration in (3.7). We obtain the following Picard-type linearization

$$\begin{aligned} A_h(\mathbf{u}_h^k,\mathbf{v})+O_h(\mathbf{u}_h^{k-1},\mathbf{u}_h^k,\mathbf{v})+B(\mathbf{v},p_h^k) &= (\mathbf{f},\mathbf{v})_\Omega - C(\mathbf{b}_h^{k-1},\mathbf{v},\mathbf{b}_h^{k-1}),\\ B(\mathbf{u}_h^k,q) &= 0,\\ M(\mathbf{b}_h^k,\mathbf{c})+D(\mathbf{c},r_h^k) &= (\mathbf{g},\mathbf{c})_\Omega + C(\mathbf{b}_h^{k-1},\mathbf{u}_h^{k-1},\mathbf{c}),\\ D(\mathbf{b}_h^k,s) &= 0. \end{aligned} \tag{4.9}$$

The coefficient matrix associated with this iteration is

$$\mathscr{K}_E = \left(\begin{array}{cc|cc} F & B^T & 0 & 0 \\ B & 0 & 0 & 0 \\ \hline 0 & 0 & M & D^T \\ 0 & 0 & D & 0 \end{array}\right). \tag{4.10}$$

The coefficient matrix (4.10) is still unsymmetric, but the two sub-systems are decoupled. Let us refer to (4.10) as a 2×2 block matrix. The (1, 1) block corresponds to the Navier-Stokes sub-system and the (2, 2) block corresponds to the Maxwell sub-system. We can apply GMRES preconditioned with (4.6) to the (1,1) block and apply MINRES preconditioned with (4.7) to the (2, 2) block.

If we also treat the convection term O explicitly and move it to the right-hand-side, we obtain the following Picard-type iteration

$$\begin{aligned}
A_h(\mathbf{u}_h^k,\mathbf{v})+B(\mathbf{v},p_h^k) &= (\mathbf{f},\mathbf{v})_\Omega - O_h(\mathbf{u}_h^{k-1},\mathbf{u}_h^{k-1},\mathbf{v}) - C(\mathbf{b}_h^{k-1},\mathbf{v},\mathbf{b}_h^{k-1}),\\
B(\mathbf{u}_h^k,q) &= 0,\\
M(\mathbf{b}_h^k,\mathbf{c})+D(\mathbf{c},r_h^k) &= (\mathbf{g},\mathbf{c})_\Omega + C(\mathbf{b}_h^{k-1},\mathbf{u}_h^{k-1},\mathbf{c}),\\
D(\mathbf{b}_h^k,s) &= 0.
\end{aligned} \tag{4.11}$$

The coefficient matrix is symmetric:

$$\mathscr{K}_S = \left(\begin{array}{cc|cc} A & B^T & 0 & 0\\ B & 0 & 0 & 0\\ \hline 0 & 0 & M & D^T\\ 0 & 0 & D & 0 \end{array}\right).$$

We can use a symmetric solve. Similar to the previous case, the system is decoupled. We now apply MINRES to both of the sub-systems. For the Navier-Stokes sub-system, we use the preconditioner

$$\begin{pmatrix} A & 0\\ 0 & M_S \end{pmatrix}.$$

For the Maxwell sub-system, we apply MINRES preconditioned with (4.7).

We expect to see fast convergence for the last two approaches. However, because the convection and coupling terms are treated explicitly, the Picard-type linearization may require more iterations to converge or may not converge, if the convection and/or coupling is strong.

(2) *Combining preconditioners for the sub-systems with coupling.* In order to develop a more faithful approximation to the discretized MHD system in (4.2), we need to incorporate effects of the coupling terms in the preconditioner. We

consider the following preconditioner

$$\mathscr{P}_C = \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -M_S & 0 & 0 \\ -C & 0 & M+X & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{4.12}$$

We expect this preconditioner to perform better when the coupling and/or convection is strong. In order to apply the preconditioning approach, we need to solve linear systems associated with $\mathscr{P}_C$. We propose to use GMRES with the following inner preconditioner

$$\mathscr{P}_I = \begin{pmatrix} F & B^T & 0 & 0 \\ 0 & -M_S & 0 & 0 \\ 0 & 0 & M+X & 0 \\ 0 & 0 & 0 & L \end{pmatrix}. \tag{4.13}$$

We provide some preliminary eigenvalue analysis for this preconditioning approach in Section 4.2.2.

(3) *Applying the least-squares preconditioner to the entire MHD system.* The system (4.2) can be written as

$$\left(\begin{array}{cc|cc} F & C^T & B^T & 0 \\ -C & M & 0 & D^T \\ \hline B & 0 & 0 & 0 \\ 0 & D & 0 & 0 \end{array}\right) \begin{pmatrix} u \\ b \\ p \\ r \end{pmatrix} = \begin{pmatrix} f \\ g \\ 0 \\ 0 \end{pmatrix}. \tag{4.14}$$

Rewrite the coefficient matrix in (4.14) as a 2×2 block matrix as the following

$$\begin{pmatrix} \mathscr{F} & \mathscr{B}^T \\ \mathscr{B} & 0 \end{pmatrix}, \tag{4.15}$$

where

$$\mathscr{F} = \begin{pmatrix} F & C^T \\ -C & M \end{pmatrix},$$

and

$$\mathscr{B} = \begin{pmatrix} B & 0 \\ 0 & D \end{pmatrix}.$$

Note that (4.15) has the same form as (4.3). Our finite element formulation is inf-sup stable; see [44, Section 3.3]. The operator $\mathscr{F}$ has a component F, which is a convection-diffusion type operator and the size of F is bigger than M. We assume $\mathscr{F}$ preserves some properties of the convection-diffusion operator. We therefore consider applying the least-squares commutator preconditioner directly to the entire discretized MHD system. The preconditioner is

$$\mathscr{P}_A = \begin{pmatrix} \mathscr{F} & \mathscr{B} \\ 0 & -\mathscr{M}_S \end{pmatrix}.$$

Here $\mathscr{M}_S$ is defined as

$$\mathscr{M}_S = (\mathscr{B}\hat{\mathscr{Q}}^{-1}\mathscr{B}^T)^{-1}(\mathscr{B}\hat{\mathscr{Q}}^{-1}\mathscr{F}\hat{\mathscr{Q}}^{-1}\mathscr{B}^T)(\mathscr{B}\hat{\mathscr{Q}}^{-1}\mathscr{B}^T)^{-1},$$

where

$$\hat{\mathscr{Q}} = \mathrm{diag}\begin{pmatrix} Q & 0 \\ 0 & X \end{pmatrix}.$$

For the three approaches discussed above, if the coupling and convection are not very strong, we can treat them explicitly. Then the Navier-Stokes and Maxwell sub-systems are decoupled, and we apply the ad-hoc preconditioners for them separately. If the coupling and convection are strong, we apply $\mathscr{P}_C$ as a preconditioner for GMRES. As later shown in the numerical examples, the approach is not completely independent of the mesh size, but shows reasonably good performance. For the approach of applying the least-squares commutator preconditioner directly to the discretized MHD system, it is not as good as the other approaches.

4.2.2 Preliminary eigenvalue analysis

Eigenvalue analysis of the Maxwell system is given in Theorem 2. It is generally difficult to derive convergence analysis for nonsymmetric systems. As far as we know, there is no complete analysis of the convergence characteristics of the least-squares preconditioner. Thus convergence analysis for $\mathscr{P}_A$ is also a very difficult problem.

In this section, we provide preliminary eigenvalue analysis for the preconditioner $\mathscr{P}_C$ in (4.12). Note that the (3,3) block $M+X$ is spectrally equivalent to $M+D^TL^{-1}D$; see [43, Theorem 3.3]. In the analysis, we consider an ideal version of (4.12)

$$\hat{\mathscr{P}}_C = \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -M_S & 0 & 0 \\ -C & 0 & M+D^TL^{-1}D & 0 \\ 0 & 0 & 0 & L \end{pmatrix}.$$

Theorem 8 *The matrix $\hat{\mathscr{P}}_C^{-1}\mathscr{K}$ has an eigenvalue $\lambda = 1$ with algebraic multiplicity of at least $n+\hat{n}$ and an eigenvalue $\lambda = -1$ with algebraic multiplicity of at least m.*

Proof The corresponding eigenvalue problem is

$$\begin{pmatrix} F & B^T & C^T & 0 \\ B & 0 & 0 & 0 \\ -C & 0 & M & D^T \\ 0 & 0 & D & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix} = \lambda \begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -M_S & 0 & 0 \\ -C & 0 & M+D^TL^{-1}D & 0 \\ 0 & 0 & 0 & L \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix}.$$

The four block rows can be written as

$$(1-\lambda)(Fu+B^Tp+C^Tb) = 0, \tag{4.16}$$

$$Bu = -\lambda M_S p, \tag{4.17}$$

$$(\lambda-1)Cu+(1-\lambda)Mb-\lambda D^TL^{-1}Db+D^Tr = 0, \tag{4.18}$$

$$Db = \lambda Lr. \tag{4.19}$$

If $\lambda = 1$, (4.16) is satisfied automatically. Equation (4.17) shows

$$p = -M_S^{-1}Bu.$$

Equation (4.19) leads to

$$r = L^{-1}Db.$$

If this holds, (4.18) is satisfied.

Therefore, $(u, -M_S^{-1}Bu, b, L^{-1}Db)$ is an eigenvector. There exist $\hat{n}$ linearly independent such u's and n linearly independent such b's. There are at least $\hat{n}+n$ linearly independent non-zero vectors satisfying the eigenvalue problem when $\lambda = 1$. Therefore $\lambda = 1$ is an eigenvalue with algebraic multiplicity of at least $n+\hat{n}$.

If $\lambda = -1$, (4.19) leads to

$$r = -L^{-1}Db.$$

Substituting it into (4.18), we obtain

$$Cu = Mb.$$

If b is a discrete gradient in $\mathbf{C}_h$, $Mb = 0$ and $C^T b = 0$. If we take $u = 0$, then $Cu = 0$. The requirement $Cu = Mb$ is satisfied.

If $u = 0$ and b is a discrete gradient, (4.16) becomes

$$B^T p = 0.$$

Since B has full row rank, this implies $p = 0$.

Therefore, if b is a discrete gradient, $(0, 0, b, -L^{-1}Db)$ is an eigenvector corresponding to $\lambda = -1$. There are m discrete gradients in $\mathbf{C}_h$. Therefore $\lambda = -1$ is an eigenvalue with at least algebraic multiplicity m. ▮

Remark 9 *In our experiments, we have observed the eigenvalue $\lambda = 1$ has algebraic multiplicity of exactly $n+\hat{n}$ and the eigenvalue $\lambda = -1$ has algebraic multiplicity of exactly m. Proving this may be difficult due to complicated generalized eigenvalue problems that other eigenvalues satisfy.*

Theorem 8 shows that $\hat{\mathscr{P}}_C^{-1}\mathscr{K}$ has some tightly clustered eigenvalues and GMRES should converge quickly. We expect a similar behavior for $\mathscr{P}_C^{-1}\mathscr{K}$.

Next, we show some preliminary eigenvalue analysis for the inner GMRES iterations.

Theorem 10 *The matrix $\mathscr{P}_I^{-1}\mathscr{P}_C$ has an eigenvalue $\lambda = 1$ with algebraic multiplicity of at least $\hat{m}+\hat{n}+3m-n$.*

Proof The corresponding eigenvalue problem is

$$\begin{pmatrix} F & B^T & C^T & 0 \\ 0 & -M_S & 0 & 0 \\ -C & 0 & M+X & 0 \\ 0 & 0 & 0 & L \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix} = \lambda \begin{pmatrix} F & B^T & 0 & 0 \\ 0 & -M_S & 0 & 0 \\ 0 & 0 & M+X & 0 \\ 0 & 0 & 0 & L \end{pmatrix} \begin{pmatrix} u \\ p \\ b \\ r \end{pmatrix}.$$

The four block rows can be written as

$$\begin{aligned} (1-\lambda)Fu+(1-\lambda)B^T p+C^T b &= 0, && (4.20) \\ (1-\lambda)M_S p &= 0, && (4.21) \\ -Cu+(1-\lambda)(M+X)b &= 0, && (4.22) \\ (1-\lambda)Lr &= 0. && (4.23) \end{aligned}$$

If $\lambda = 1$, (4.21) and (4.23) are satisfied automatically.

Equation (4.20) becomes

$$C^T b = 0.$$

If b is a discrete gradient, this is satisfied.

Equation (4.22) becomes

$$Cu = 0.$$

If u is in the null space of C, we have $Cu = 0$.

Therefore, (u, p, b, r) is an eigenvector associated with eigenvalue $\lambda = 1$, if u is in the null space of C and b is a discrete gradient. The matrix C is $n \times \hat{n}$, so we have

$$\begin{aligned} \text{rank}(C^T) + \text{nullity}(C^T) &= n, && (4.24) \\ \text{rank}(C) + \text{nullity}(C) &= \hat{n}. && (4.25) \end{aligned}$$

There are m distinct discrete gradients in C_h. The nullity of C^T is at least m. According to (4.24), the rank of C^T is at most $n-m$. Matrix C has the same rank as C^T. According to (4.25), the nullity of C is at least $\hat{n}-n+m$. We have $\hat{m}$ linearly independent p in Q_h and m linearly independent r in S_h. So, there are at least

$\hat{m}+\hat{n}+3m-n$ linearly independent eigenvectors. Therefore, the eigenvalue $\lambda = 1$ has algebraic multiplicity of at least $\hat{m}+\hat{n}+3m-n$. ∎

Theorem 10 shows that many eigenvalues of $\mathscr{P}_I^{-1}\mathscr{P}_C$ are tightly clustered. We expect GMRES to converge within a few iterations.

4.3 Numerical results

In this section we present a series of numerical experiments. The primary purpose of our experiments is to compare the performance of various preconditioners presented in Section 4.2.1. We use inner-outer Krylov iterations for each Picard iteration. In all experiments, which are carried out using MATLAB, the relative residual of the outer iteration is set to 1e-6 and the relative residual of the inner iteration is set to 1e-8. The tolerance for the Picard iterations is chosen as 1e-5. For outer iterations, we show iteration counts in the last Picard iteration. For inner iterations, we show iterations counts for the last inner iteration. In the numerical experiments, we denote by (i), (ii), and (iii) the three Picard-type of linearization schemes, where (i) denotes the standard Picard linearization in (3.7), (ii) and (iii) denote the Picard-type linearizations in (4.9) and (4.11), respectively.

4.3.1 Example 1: two-dimensional problem with a smooth solution

In this example, we apply our solvers for linear systems arising from Section 3.3.1. We assume $\kappa = 1$, $\nu_m = 1\text{e}4$.

Table 4.1 shows the number of iterations required for different Picard-type linearization schemes. The symbol nc indicates the linearization does not converge. We run the experiments with different values of ν. Note that ν only shows up in front of the diffusion operator A and reducing it, both the convection and the coupling get relatively stronger.

Table 4.1 shows that when the convection and coupling get stronger, linearization (iii) in (4.11) does not always converge to the solution, while the other two schemes converge. Scheme (ii) in (4.9) behaves like the standard Picard linearization. The numbers of steps required to converge by these two schemes are the same in this example.

	$\nu = 1$			$\nu = 0.1$			$\nu = 0.01$		
DOFs	(i)	(ii)	(iii)	(i)	(ii)	(iii)	(i)	(ii)	(iii)
169	5	5	5	8	8	nc	13	13	nc
721	5	5	5	9	9	nc	15	15	nc
2,977	5	5	5	9	9	nc	15	15	nc
12,097	5	5	5	8	8	nc	14	14	nc

Table 4.1: Example 1. Picard-type iteration counts for various values of ν. We denote by (i) the standard Picard linearization in (3.7), by (ii) and (iii) the Picard-type linearizations in (4.9) and (4.11), respectively.

If we treat the coupling terms explicitly, the linear system is decoupled. We solve the Navier-Stokes and Maxwell sub-systems separately. Table 4.2 shows the iteration counts for the two sub-problems. We solve the Navier-Stokes sub-system with GMRES and use the least-squares commutator preconditioner. We denote by its_1 the GMRES iterations. The Maxwell system is solved with MINRES with the block diagonal preconditioner in (4.7). We denote by its_2 the MINRES iterations. When the mesh is refined, more GMRES iterations are required, while the number of MINRES iterations remains the same. This shows our preconditioner for the Maxwell sub-system is independent of the mesh size, but the least-squares commutator preconditioner is not (yet it is not very sensitive to a change of the mesh size). When the convection and coupling get stronger, typically more iterations are required. The fast convergence of the Maxwell problem can be attributed to the fact that the eigenvalues appear in two extremely tight clusters.

	$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
DOFs	its_1	its_2	its_1	its_2	its_1	its_2
169	28	2	27	2	24	2
721	51	2	48	2	57	2
2,977	58	2	58	2	85	2
12,097	66	2	65	2	99	2

Table 4.2: Example 1. Iteration counts for various values of ν. The coupling terms are treated explicitly.

When both the coupling and convection terms are treated explicitly, we show the number of MINRES iterations required for the two sub-problems. We only show results for $\nu = 1$, because in the other cases, this linearization scheme does not converge. When the mesh is refined, we observe that the number of iteration counts for the Navier-Stokes problem increases, while that for the Maxwell system remains the same.

	$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
DOFs	its_1	its_2	its_1	its_2	its_1	its_2
169	38	2	nc	nc	nc	nc
721	42	2	nc	nc	nc	nc
2,977	44	2	nc	nc	nc	nc
12,097	54	2	nc	nc	nc	nc

Table 4.3: Example 1. Iteration counts for various values of ν. The coupling and convection terms are treated explicitly.

Table 4.4 shows the inner and outer GMRES iteration counts, using the outer preconditioner $\mathscr{P}_C$ in (4.12) and the inner preconditioner $\mathscr{P}_I$ in (4.13). Here, its denotes the number of outer GMRES iterations, and its_i denotes the number of inner GMRES iterations. When the mesh is refined, the outer GMRES solve requires more iterations and the iteration counts are comparable to the iteration counts for the Navier-Stokes sub-system in Table 4.2. The preconditioner $\mathscr{P}_C$ is not independent of the mesh size, but it is not very sensitive to the change. When ν decreases, more outer iterations are required. The inner iterations show mesh-size independent behavior and they are not sensitive to a change of ν, either.

The real eigenvalues of the preconditioned linear system $\hat{\mathscr{P}}_C^{-1}\mathscr{K}$ are shown in Figure 4.1. In this experiment, we take $\nu = 1$ and use a mesh with 721 degrees of freedom, which corresponds to $n = 176$, $m = 49$, $\hat{n} = 368$ and $\hat{m} = 128$. According to Theorem 8, we expect to see $\lambda = 1$ with algebraic multiplicity of at least $n+\hat{n} = 544$ and $\lambda = -1$ with algebraic multiplicity of at least $m = 49$. In our experiment, we obtain $\lambda = 1$ with algebraic multiplicity of exactly 544, and $\lambda = -1$ with algebraic multiplicity of exactly 49. For the inner preconditioner, we expect to see $\lambda = 1$ with algebraic multiplicity of at least $\hat{m}+\hat{n}+3m-n = 467$ of the pre-

	$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
DOFs	its	its_i	its	its_i	its	its_i
169	29	3	28	4	26	5
721	54	3	52	3	61	5
2,977	62	3	63	3	94	4
12,097	75	3	70	3	110	4

Table 4.4: Example 1. Inner and outer iteration counts for various values of ν, using $\mathscr{P}_C$.

conditioned linear system $\mathscr{P}_I^{-1}\mathscr{P}_C$. Numerically, we obtain 469 real eigenvalues and they are all 1.

Table 4.5 shows iteration counts for GMRES preconditioned with $\mathscr{P}_A$. The number of iteration counts grows up very quickly, as we refine the mesh. This preconditioning approach is more sensitive to a change of the mesh size, compared with the previous approaches.

DOFs	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$
169	37	35	33
721	98	97	106
2,977	nc	283	310
12,097	nc	nc	nc

Table 4.5: Example 1. Iteration counts for various values of ν, using $\mathscr{P}_A$.

4.3.2 Example 2: two-dimensional problem with a singular solution

In this example, we apply our solvers for linear systems arising from Section 3.3.2. We assume $\kappa = 1$, $\nu_m = 1\text{e}4$.

Table 4.6 shows the number of iterations required for different Picard-type linearization schemes. Again, we observe that scheme (iii), which is the Picard-type linearization in (4.11), converges more slowly than the other two linearization schemes. If the coupling and convection get stronger, this linearization does not converge. Scheme (ii) given in (4.9) shows a similar behavior to the standard Picard linearization. However, because the convection term is treated explicitly in

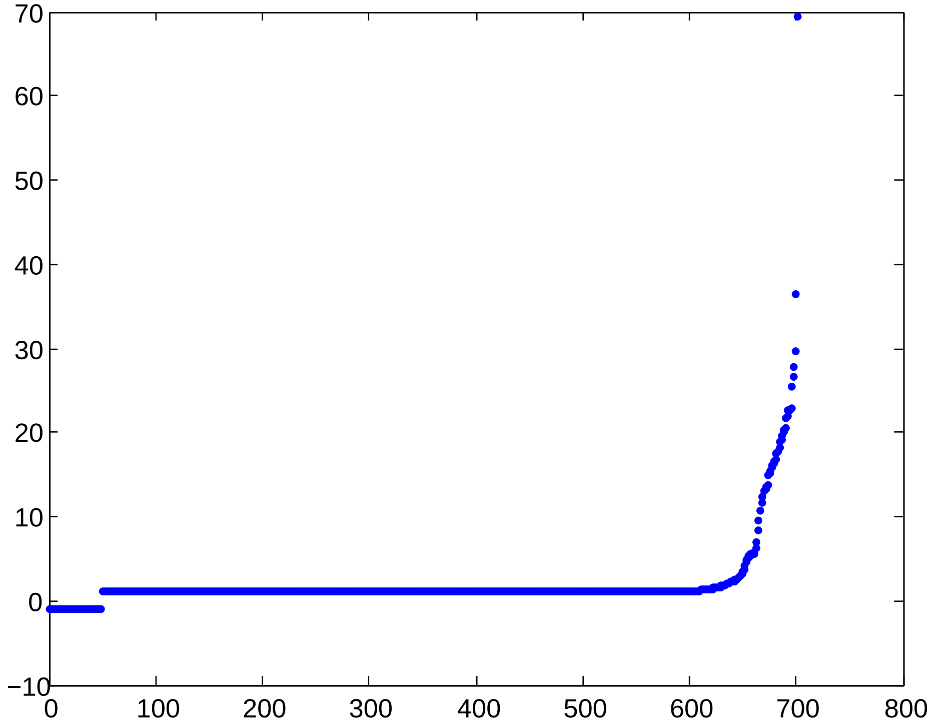


Figure 4.1: Example 1. Plot of the eigenvalues of the preconditioned matrix $\hat{\mathscr{P}}_C^{-1}\mathscr{K}$, DOFs $= 721$.

this scheme, it will fail to converge if convection is strong. Indeed, for this example, taking $\nu = 0.01$, $\nu_m = 1e4$ and $\kappa = 1e6$, scheme (ii) fails to converge, while the standard Picard linearization still converges.

Table 4.7 shows the iteration counts for the two sub-problems, when the coupling terms are treated explicitly. Here, we also find that the least-squares commutator preconditioner is not completely independent of the mesh size, but it is not very sensitive to a change of the mesh size. For the Maxwell sub-problem, our scheme is not sensitive to a change of the mesh size.

Table 4.8 shows the inner and outer GMRES iteration counts, using the outer preconditioner $\mathscr{P}_C$ in (4.12) and the inner preconditioner $\mathscr{P}_I$ in (4.13). The outer

	$\nu = 1$			$\nu = 0.1$			$\nu = 0.01$		
DOFs	(i)	(ii)	(iii)	(i)	(ii)	(iii)	(i)	(ii)	(iii)
117	4	4	7	6	6	nc	8	8	nc
521	5	5	9	7	7	nc	10	10	nc
2,193	4	4	9	8	8	nc	10	10	nc
8,893	4	4	9	7	7	nc	11	11	nc

Table 4.6: Example 2. Picard-type iteration counts for various values of ν. We denote by (i) the standard Picard linearization in (3.7), by (ii) and (iii) the Picard-type linearizations in (4.9) and (4.11), respectively.

	$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
DOFs	its_1	its_2	its_1	its_2	its_1	its_2
117	22	1	20	1	18	1
521	46	1	50	1	40	1
2,193	56	1	84	1	83	1
8,893	63	1	88	1	99	1

Table 4.7: Example 2. Iteration counts for various values of ν. The coupling terms are treated explicitly.

iterations demonstrate a similar behavior to the iterations of the Navier-Stokes subsystem in Table 4.7. The inner iterations are not sensitive to a change of the mesh size and ν.

	$\nu = 1$		$\nu = 0.1$		$\nu = 0.01$	
DOFs	its	its_i	its	its_i	its	its_i
117	22	3	20	3	18	3
521	46	3	50	3	41	3
2,193	56	3	84	3	89	3
8,893	68	3	106	3	138	3

Table 4.8: Example 2. Inner and outer iteration counts for various values of ν, using $\mathscr{P}_C$.

Table 4.9 shows iteration counts for GMRES preconditioned with $\mathscr{P}_A$. Here, we also find that the iteration does not scale very well when we refine the mesh.

DOFs	$\nu = 1$	$\nu = 0.1$	$\nu = 0.01$
117	27	25	23
521	79	83	73
2,193	217	245	244
8,893	nc	nc	nc

Table 4.9: Example 2. Iteration counts for various values of ν, using $\mathscr{P}_A$.

4.3.3 Discussion

We have considered three different approaches to combining the preconditioners for the sub-systems. We choose the preconditioner based on how strong the convection and coupling are. If they are not very strong, we take the first approach, where preconditioners for the sub-systems are combined together directly without coupling. If the coupling terms are small, we can treat them explicitly. As shown in Tables 4.1 and 4.6, the linearization scheme (ii) requires the same iterations as the standard Picard linearization (i). If both of the coupling and convection terms are negligible, we can treat both of them explicitly as in scheme (iii). When the mesh is refined, schemes (ii) and (iii) are not completely mesh-independent, but they are not very sensitive to a change of the mesh size. However, if the coupling and convection are strong, these schemes will converge more slowly compared with the standard Picard scheme, or may not converge. We propose to combine the preconditioners for the sub-systems with the coupling terms as in $\mathscr{P}_C$. As shown in Tables 4.4 and 4.8, this approach performs qualitatively similar to the least-squares commutator preconditioner for the Navier-Stoke sub-problem shown in Tables 4.2 and 4.7. The third approach, where the least-squares commutator is applied to the entire discretized MHD system, does not work as well as the other two approaches, as shown in Tables 4.5 and 4.9.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this thesis, we have studied the numerical solutions of the time-harmonic Maxwell equations and incompressible MHD equations. For the Maxwell equations, we have proposed and analyzed an efficient solver based on inner-outer iterations. We have demonstrated very good scalability in our numerical experiments. For the MHD equations, we have designed a new finite element discretization. This discretization yields exactly divergence-free velocity approximations and captures the strongest magnetic singularities. We have also investigated preconditioning techniques for the discretized MHD equations. We have proposed a preconditioner based on efficient preconditioners of the Maxwell and Navier-Stokes sub-systems. Our numerical results clearly indicate optimal convergence rates in two and three dimensions and show that our preconditioning approach performs reasonably well.

In the following subsections, we summarize the results in the thesis and draw some conclusions.

5.1.1 The Maxwell problem

We have developed and implemented a fully scalable parallel iterative solver for the time-harmonic Maxwell equations with variable coefficients, on unstructured complicated domains with heterogeneous coefficients. Our code for preconditioned iterative solvers is specialized, dealing with the specific features of the discretization, such as the use of edge elements. We use state-of-the-art software packages (PETSc [7], BoomerAMG [51], Hypre [38], and METIS [62]) to optimize the performance of our solvers. We have also developed our own mesher for structured meshes, and we use TetGen [91] for unstructured and locally refined meshes.

The preconditioned solvers are based on inner-outer iterations. For outer iterations we extend and generalize [43] to the variable coefficient case. The inner iterations are based on the auxiliary spaces technique developed in [56].

We have shown that the preconditioned matrix is well conditioned and its eigenvalues are tightly clustered; this is key to the effectiveness of the proposed approach. As our extensive numerical experiments indicate, the inner and outer iterations are scalable in terms of iteration counts and computation times, and the solver is minimally sensitive to changes in the mesh size. Highly varying coefficients result in higher iteration counts, but those barely change as the mesh is refined.

5.1.2 The MHD problem

We have introduced a new mixed finite element method for the numerical discretization of a stationary incompressible magnetohydrodynamics problem, with divergence-conforming BDM elements and curl-conforming Nédélec elements for the velocity and magnetic fields, respectively. The approximation of the velocity field is exactly mass conservative. The discrete formulation is well-posed under a standard small data assumption, and the approximations are convergent under minimal regularity assumptions.

We have presented an extensive set of numerical tests in Section 3.3. The numerical experiments show optimal convergence in two and three dimensions, and indicate that the constant C_d in Theorem 7 stays bounded in all cases, even though this is not guaranteed by the analysis. Altogether, the computed results are in excellent agreement with results in the literature, and the method correctly resolves the strongest magnetic singularities in non-convex domains.

Based on the theoretical results in [82], we expect the same good performance of our discretization and solution techniques to carry over to the dynamic problem, provided that the nonlinear terms are treated (semi)implicitly. We also mention the issue of higher order elements. Here, too, we do not expect any deviation from our current computational results. In particular, we expect to see optimal convergence rates for smooth solutions.

We have done a preliminary study in efficient solvers for the discretized MHD equations based on Krylov methods. We have investigated a few preconditioning ideas. If the coupling and convection are not very strong, we can treat them explicitly. Then the Navier-Stokes and Maxwell sub-systems are decoupled, and we apply preconditioners for them separately. If the coupling and convection are strong, we have developed a preconditioner $\mathscr{P}_C$ for GMRES based on combining the preconditioners for the sub-systems and the coupling terms. We have shown that many of the eigenvalues are well-clustered. We have also investigated a preconditioner for the inner iterations. Our numerical results show that this approach scales reasonably well, though is not fully scalable.

5.2 Future work

The scope of our work can be broadened in a number of directions:

1. Our results for the solution of the Maxwell equations apply to low wave numbers; dealing with high wave numbers is a primary challenge of much interest. If the wave number is high, the inf-sup constant for our mixed formulation is large [32], which results in numerical difficulties. Therefore, solving problems with high wave numbers exhibits several complications that need to be dealt with using a possibly different methodology.

2. In our parallel solver for the Maxwell equations, the meshing part is sequential and each processor stores the entire mesh. A dense mesh may require high memory resources. Integrating our solver with parallel distributed mesh generators could relieve the memory consumption and improve the performance of our solvers. Parallel mesh generation methods require the partitioning of the mesh in a way that accommodates load balancing. However, this goal may not always accommodate other important factors, such as the quality of the mesh. As a result, the overall computational time may increase. Optimizing these issues is a challenging and involved task.

3. Our preconditioners for the MHD equations perform reasonably well, but are not fully scalable. Deriving effective and scalable preconditioners remains an item for future research. While there are efficient solution techniques for the Navier-Stokes equations as well as for the curl-curl operator, the primary challenge here is how to deal with the skew-symmetric coupling terms, especially when coupling is strong.

4. While for the discretized Maxwell problem we have a parallelized and efficient code, we do not have such a code yet for the discretized MHD equations. When the mesh is refined, the size of the resulting linear system increases very quickly and efficient parallel implementations are crucial for obtaining a solution within a reasonable computational time.

5. Another item for future work is the derivation of a non-linear solver for the MHD equations, which potentially converges more rapidly than the Picard iteration used in our experiments. As we have pointed out in Remark 4 of Section 3.2.2, developing the Newton iteration for our discretization is somewhat delicate and is the subject of ongoing investigation.

6. For the preconditioning techniques for the discretized MHD problems, we provide some preliminary eigenvalue analysis. The complete eigenvalue analysis is very challenging, because it involves complicated generalized eigenvalue problems including the coupling terms. A complete analysis may require a better understanding of the differential properties of the coupling terms.

Bibliography

[1] 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.

[2] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault. Vector potentials in three-dimensional non-smooth domains. *Mathematical Methods in The Applied Sciences*, 21:823–864, 1998.

[3] F. Armero and J. 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*, 131:41–90, 1996.

[4] D. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19:742–760, 1982.

[5] D. Arnold, F. Brezzi, B. Cockburn, and L. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39:1749–1779, 2001.

[6] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficienct management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhauser Press, 1997.

[7] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc users manual. Technical Report ANL-95/11 - Revision 2.3.2, Argonne National Laboratory, 2006.

[8] S. Balay, K. Buschelman, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2009.

[9] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II Web page. http://www.dealii.org/, 2010.

[10] M. Benzi. Preconditioning techniques for large linear systems: a survey. *Journal of Computational Physics*, 182(2):418–477, 2002.

[11] M. Benzi and G. H. Golub. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):20–41, 2005.

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

[13] P. Bochev, C. Garasi, J. Hu, A. Robinson, and R. Tuminaro. An improved algebraic multigrid method for solving Maxwell's equations. *SIAM Journal on Scientific Computing*, 25:623–642, 2003.

[14] A. Bonito and J.-L. Guermond. Approximation of the eigenvalue problem for the time harmonic Maxwell system by continuous Lagrange finite elements. Technical Report 2009-121, Institute for Applied Mathematics and Computational Science, Texas A&M University, 2009.

[15] F. Brezzi and M. Fortin. Mixed and Hybrid Finite Element Methods. In *Springer Series in Computational Mathematics*, volume 15. Springer-Verlag, New York, 1991.

[16] P. Castillo, R. Rieben, and D. White. FEMSTER: An object-oriented class library of high-order discrete differential forms. *ACM Transactions on Mathematical Software*, 31:425–457, 2005.

[17] Z. Chen, Q. Du, and J. Zou. Finite element methods with matching and nonmatching meshes for Maxwell equations with discontinuous coefficients. *SIAM Journal on Numerical Analysis*, 37(5):1542–1570, 2000.

[18] A. Chorin and J. Marsden. *A Mathematical Introduction to Fluid Mechanics*. Springer-Verlag, 1990.

[19] E. Chow and Y. Saad. Approximate inverse techniques for block-partitioned matrices. *SIAM Journal on Scientific Computing*, 18(6):1657–1675, 1997.

[20] B. Cockburn and C.-W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16: 173–261, 2001.

[21] B. Cockburn, G. Karniadakis, and C.-W. Shu, editors. *Discontinuous Galerkin Methods. Theory, Computation and Applications*, volume 11 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, 2000.

[22] B. Cockburn, G. Kanschat, and D. Schötzau. Local discontinuous Galerkin methods for the Oseen equations. *Mathematics of Computation*, 73: 569–593, 2004.

[23] B. Cockburn, G. Kanschat, and D. Schötzau. A locally conservative LDG method for the incompressible Navier-Stokes equations. *Mathematics of Computation*, 74:1067–1095, 2005.

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

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

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

[27] M. Costabel and M. Dauge. Weighted regularization of Maxwell equations in polyhedral domains. *Numerische Mathematik*, 93:239–277, 2002.

[28] T. G. Cowling. *Magnetohydrodynamics*. Adam Hilger, England, 1976.

[29] M. Dauge. Stationary Stokes and Navier-Stokes systems on two- or three-dimensional domains with corners. Part I: Linearized equations. *SIAM Journal on Mathematical Analysis*, 20:74–97, 1989.

[30] P. A. Davidson. *An Introduction to Magnetohydrodynamics*. Cambridge University Press, 2001.

[31] C. Dawson. Foreword for the special issue on discontinuous Galerkin methods. *Computer Methods in Applied Mechanics and Engineering*, 195:3, 2006.

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

[33] J. W. Demmel. *Applied Numerical Linear Algebra.* SIAM, 1997.

[34] M. Discacciati. Numerical approximation of a steady MHD problem. In U. Langer, M. Discacciati, D. E. Keys, O. B. Widlund, and W. Zulehner, editors, *Domain Decomposition Methods in Science and Engineering XVII*, volume 60 of *Lecture Notes in Computational Science and Engineering*, pages 313–320. Springer-Verlag, 2008.

[35] H. Elman and D. Silvester. Fast nonsymmetric iterations and preconditioning for Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 17(1):33–46, 1996.

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

[37] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, 2006.

[38] R. Falgout, A. Cleary, J. Jones, E. Chow, V. Henson, C. Baldwin, P. Brown, P. Vassilevski, and U. M. Yang. Hypre Web page. http://acts.nersc.gov/hypre/, 2010.

[39] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala. ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[40] J.-F. Gerbeau. *Problèmes mathématiques et numériques posés par la modélisation de l'électrolyse de l'aluminium.* PhD thesis, École Nationale des Ponts et Chaussées, 1998.

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

[42] J.-F. Gerbeau, C. L. Bris, and T. Lelièvre. *Mathematical Methods for the Magnetohydrodynamics of Liquid Metals*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2006.

[43] 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.

[44] C. Greif, D. 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*, 199:2840–2855, 2010.

[45] M. Griebel and P. Oswald. On the abstract theory of additive and multiplicative Schwarz algorithms. *Numerische Mathematik*, 70(2): 163–180, 1995.

[46] W. Gropp, E. Lusk, D. Ashton, P. Balaji, D. Buntinas, R. Butler, A. Chan, J. Krishna, G. Mercier, R. Ross, R. Thakur, and B. Toonen. MPI Web page. http://www-unix.mcs.anl.gov/mpi/, 2007.

[47] J.-L. Guermond and P. Minev. Mixed finite element approximation of an MHD problem involving conducting and insulating regions: the 3D case. *Numerical Methods for Partial Differential Equations*, 19:709–731, 2003.

[48] J.-L. Guermond, R. Laguerre, J. Léorat, and C. Nore. Nonlinear magnetohydrodynamics in axisymmetric heterogeneous domains using a Fourier/finite element technique and an interior penalty method. *Journal of Computational Physics*, 228:2739–2757, 2009.

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

[50] U. Hasler, A. Schneebeli, and D. Schötzau. Mixed finite element approximation of incompressible MHD problems based on weighted regularization. *Applied Numerical Mathematics*, 51:19–45, 2004.

[51] V. E. Henson and U. M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2000.

[52] M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Transactions on Mathematical Software*, 31(3):397–423, 2005.

[53] M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards*, 49 (6):409–436, 1952.

[54] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM Journal on Numerical Analysis*, 36(1):204–225, 1998.

[55] R. Hiptmair. Finite elements in computational electromagnetism. *Acta Numerica*, 11:237–339, 2002.

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

[57] P. Houston, I. Perugia, and D. Schötzau. Mixed discontinuous Galerkin approximation of the Maxwell operator: Non-stabilized formulation. *Journal of Scientific Computing*, 22:315–346, 2005.

[58] P. Houston, D. Schötzau, and X. Wei. A mixed DG method for linearized incompressible magnetohydrodynamics. *Journal of Scientific Computing*, 40:281–314, 2009.

[59] J. Jackson. *Classical Electrodynamics*. Jon Wiley & Sons, Inc., 2nd edition, 1975.

[60] J. Jin. *The Finite Element Method in Electromagnetics*. Jon Wiley & Sons, Inc., 1993.

[61] J. Jones and B. Lee. A multigrid method for variable coefficient Maxwell's equations. *SIAM Journal on Scientific Computing*, 27(5):1689–1708, 2006.

[62] G. Karypis. METIS Web page. http://glaros.dtc.umn.edu/gkhome/views/metis/, 2010.

[63] G. Karypis and V. Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48(1):96–129, 1998.

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

[65] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers*, 22(3):237–254, 2006.

[66] A. Klawonn and G. Starke. Block triangular preconditioners for nonsymmetric saddle point problems: field-of-values analysis. *Numerische Mathematik*, 81:577–594, 1999.

[67] T. Kolev and P. Vassilevski. Parallel auxiliary space AMG for H(curl) problems. *Journal of Computational Mathematics*, 27:604–623, 2009.

[68] P. LeSaint and P. Raviart. On a finite element method for solving the neutron transport equation. In C. de Boor, editor, *Mathematical Aspects of Finite Elements in Partial Differential Equations*, pages 89–123. Academic Press, New York, 1974.

[69] D. Li. Parallel numerical solution of the time-harmonic Maxwell equations. *Lecture Notes in Computer Science on High Performance Computing and Applications*, 5938:224–229, 2010.

[70] D. Li, C. Greif, and D. Schötzau. Parallel numerical solution of the time-harmonic Maxwell equations in mixed form. *Numerical Linear Algebra with Applications*, Submitted (19 pages), 2010.

[71] X. S. Li and J. W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. *ACM Transactions on Mathematical Software*, 29(2):110–140, 2003.

[72] J. C. Maxwell. *A treatise on electricity and magnetism.* Clarendon Press, 1873.

[73] P. Monk. *Finite Element Methods for Maxwell's Equations*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2003.

[74] U. Müller and L. Bühler. *Magnetofluiddynamics in Channels and Containers*. Springer-Verlag, Berlin, 2001.

[75] 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, 1999.

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

[77] A. I. Nesliturk, S. H. Aydin, and M. Tezer-Sezgin. Two-level finite element method with a stabilizing subgrid for the incompressible MHD equations. *International Journal or Numerical Methods in Fluids*, 58:551–572, 2008.

[78] M. A. Olshanskii and Y. V. Vassilevski. Pressure Schur complement preconditioners for the discrete Oseen problem. *SIAM Journal on Scientific Computing*, 29:2686–2704, 2007.

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

[80] I. Perugia, D. Schötzau, and P. Monk. Stabilized interior penalty methods for the time-harmonic Maxwell equations. *Computer Methods in Applied Mechanics and Engineering*, 191:4675–4697, 2002.

[81] W. P. Petersen and P. Arbenz. *Introduction to Parallel Computing: [A Practial Guide with Examples in C].* Oxford University Press, 2004.

[82] A. Prohl. Convergent finite element discretizations of the nonstationary incompressible magnetohydrodynamic system. *Mathematical Modelling and Numerical Analysis*, 42:1065–1087, 2008.

[83] S. Reitzinger and J. Schöberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numerical Linear Algebra with Applications*, 9(3):223–238, 2002.

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

[85] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.

[86] 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.

[87] O. Schenk and K. Gärtner. PARDISO Web page. http://www.pardiso-project.org, 2010.

[88] J. Schöberl. NGSolve Web page. http://www.hpfem.jku.at/ngsolve/index.html, 2010.

[89] D. Schötzau. Mixed finite element methods for incompressible magneto-hydrodynamics. *Numerische Mathematik*, 96:771–800, 2004.

[90] L. Scott, T. Clark, and B. Bagheri. *Scientific Parallel Computing*. Princeton University Press, 2005.

[91] H. Si. TetGen Web page. http://tetgen.berlios.de/, 2010.

[92] D. Silvester, H. Elman, D. Kay, and A. Wathen. Efficient preconditioning of the linearized Navier-Stokes equations for incompressible flow. *Journal of Computational and Applied Mathematics*, 128(1-2):261–279, 2001.

[93] L. N. Trefethen and D. Bau. *Numerical Linear Algebra*. SIAM, 1997.

[94] L. Vardapetyan and L. Demkowicz. *hp*-adaptive finite elements in electromagnetics. *Computer Methods in Applied Mechanics and Engineering*, 169:331–344, 2 1999.

[95] X. Wei. *Finite element methods for incompressible magnetohydrodynamics*. PhD thesis, The University of British Columbia. To appear.

[96] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing*, 56(3):215–235, 1996.