Nested Iteration and First-Order Systems Least Squares for Incompressible Resistive Magnetohydrodynamics

by

James Haley Adler

B.A., Cornell University, 2004

M.S., University of Colorado, Boulder, 2006

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Applied Mathematics

2009

This thesis entitled:
Nested Iteration and First-Order Systems Least Squares for Incompressible Resistive Magnetohydrodynamics
written by James Haley Adler
has been approved for the Department of Applied Mathematics

Prof. Tom Manteuffel

Prof. Steve McCormick

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Adler, James Haley (Ph.D., Applied Mathematics)

Nested Iteration and First-Order Systems Least Squares for Incompressible Resistive Magnetohydrodynamics

Thesis directed by Prof. Tom Manteuffel

Magnetohydrodynamics (MHD) is a fluid theory that describes Plasma Physics by treating the plasma as a fluid of charged particles. Hence, the equations that describe the plasma form a nonlinear system that couples Navier-Stokes with Maxwell's equations. This thesis develops a nested-iteration-Newton-FOSLS-AMG approach to solve this type of system. For full efficiency, most of the work is done on coarse grids, including most of the linearizations. We show that at most one Newton step and a few V-cycles are all that is needed on the finest grid. Here, we describe how the FOSLS method can be applied to incompressible resistive MHD and how it can be used to solve these MHD problems efficiently in a full multigrid approach. An algorithm is developed that uses the a posteriori error estimates of the FOSLS formulation to determine how well the system is being solved and what needs to be done to get the most accuracy per computational cost. In addition, various aspects of the algorithm are analyzed, including a timestepping analysis to confirm stability of the numerical scheme as well as the benefits of an efficiency based adaptive mesh refinement method. A 3D steady state and a reduced 2D time-dependent test problem are studied. The latter equations can simulate a "large aspect-ratio" tokamak. The goal is to resolve as much physics from the test problems with the least amount of computational work. This thesis shows that this is achieved in a few dozen work units (a work unit is the cost of one fine grid residual evaluation).

Dedication

To my grandparents:

Charles and Evelyn Haley

Sam and Lillian Adler

Acknowledgements

I'd first like to thank my advisors, Tom Manteuffel and Steve McCormick for all their support and advice in the past few years. From them, I have learned how to be a better researcher, professional, and academic. They taught me the importance of enjoying what you do and the excitement that comes from studying mathematics. Without them, I would not have learned so much, not have biked up so many hills, skiied so many runs, nor learned how to look for a pun at every possible moment. I'd also like to thank John Ruge and Marian Brezina for all their help and insight. Without John and Marian explaining what I was doing wrong or fixing all my bugs, this thesis would not be possible. I greatly appreciate their patience in responding to thousands of emails and the hundreds of hours they spent sitting down with me to get things to work. I'd like to thank the rest of my committee, Bobby Philip, Scott Parker, and Srinath Vadlamani for sharing all their knowledge on the physics of my problem and showing me what I should be looking for. In addition, I want to thank the rest of the Grandview Gang while I was in Boulder for making every meeting that much more enjoyable. To Eunjung Lee for teaching me everything about FOSLS, to Josh Nolting and Geoff Sanders for their senior advice and help getting my project done, and to Christian Ketelsen, Minho Park, and Lei Tang for sharing all the bad jokes. I'd like to thank all my friends in Boulder and family in New York for supporting me and all the teachers and mentors that I've known, from my parents, to my high school and college teachers, as well as my research mentors at Lawrence Livermore National Lab. Finally, this thesis would be not be possible without all the support from the Department of Applied Mathematics and funding from the Department of Energy, Lawrence Livermore National Laboratory, and the National Science Foundation.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

In many physical applications, complex systems of partial differential equations (PDEs) must be solved. These equations can be time dependent and nonlinear and they can involve several dependent variables in many dimensions. In some of these applications, analytic solutions are not available and robust numerical simulations are needed to solve the problems that arise. This thesis presents one such application, Magnetohydrodynamics (MHD), which consists of a set of nonlinear time-dependent PDEs. These equations contain many unknowns and need to be solved on computational domains that contain millions to billions of degrees of freedom. Specifically, in simulations of fusion, physicists study the effects of plasma instabilities in large fusion reactors such as tokamaks [47]. This thesis develops a first-order system least squares (FOSLS) [14, 15] formulation on the MHD equations. The overall goal is to develop a numerical scheme that approximates the solution of the MHD equations accurately, resolving the desired properties, but with the least amount of computational cost. The equations are first linearized and then discretized by the FOSLS finite element method. This approach has been shown to work well on similar types of PDEs [4, 5, 14, 15, 19, 20, 28, 43, 49].

The FOSLS functional yields a sharp a posteriori local error estimate, which is used to make the algorithm more robust and, under the right conditions, produces algebraic systems that are solved easily by multilevel iterative solvers. Our

choice here is algebraic multigrid (AMG) [8, 12, 38, 45], which, when applied to the FOSLS discretization, has been shown to be an optimal ($O(n)$) solver [12, 14, 15]. In general, this might not be the case, but it has been shown, specifically in [50], that when the FOSLS functional is H^1 equivalent, one can find an optimal multigrid solver for the algebraic systems produced. It also has been shown numerically to be true for various applications [4, 5, 14, 15, 19, 20, 28, 43, 49]. The key feature to our algorithm is nested iteration. We first solve the system on a very coarse grid where computation is cheap. Next, we interpolate this approximate solution to a finer grid and use it as a starting guess for solving the problem at that level. Thus, by the time the desired grid is reached, we expect the solution on the previous grid to be close to the solution on the current grid and, therefore, only a minimal amount of computation is needed to find that fine grid approximation to the solution. This thesis demonstrates that the Newton-FOSLS-AMG solver in the context of nested iteration yields solutions to the MHD equations in a small number of work units that is bounded independently of grid size, while using a limited amount of computational resources. Here, we define a work unit to be the cost of one computation of the residual on the finest grid (i.e., one matrix-vector multiplication on the finest grid). The FOSLS methodology is what makes this possible.

In this thesis, we analyze various aspects of the model as well as the algorithm. First, in Chapter 2, we describe the incompressible resistive MHD equations. We explain where the model comes from and describe the specific set of equations that are used. Chapter 3 describes the discretization methods. Here, we briefly discuss the FOSLS methodology and describe how it produces the algebraic systems that we solve. This chapter also discusses the implications of the nonlinearities in the system and how we need to linearize the problem. Next, Chapter 4 analyzes the MHD equations and how they are manipulated to be used in a

FOSLS setting. We discuss various formulations and scalings in order to put the system in an appropriate first-order system. The goal of this chapter is to show that by rearranging the equations to be amenable to FOSLS, we still preserve the physical attributes of the problem as well as guarantee a solution to the discretized problem via H^1 ellipticity. In Chapter 5, we discuss the solution algorithm. This chapter describes the nested iteration strategy that we use to approximate the solution of the MHD system. We show the methodology behind solving a linear system as well as a nonlinear system in this nested iteration-FOSLS way. In this chapter, we discuss how we move up through successively finer grids and how well we solve the problem at each of these levels. In addition, we take some time to study the implications of timestepping. In this context, the FOSLS formulations yields a set of differential algebraic equations (DAEs). We study the stability and convergence of the time-stepping schemes used. Finally, Chapter 6 gives numerical results of our entire algorithm on a 3D problem as well as a reduced 2D problem that simulates certain instabilities in a plasma in a tokamak fusion reactor. The results confirm that the FOSLS discretization resolves solutions to the MHD equations in a small number of work units, independently of the discretization parameters.

Chapter 2

Background of MHD

The study of Plasma Physics involves the movement of charged particles. Plasmas are treated by a hierarchy of models [36]. Ideally, each particle would be tracked and all the forces acting on it would be computed, which of course would involve an enormous amount of computation and is, therefore, impractical. Instead, in kinetic theory, an ensemble average of the particles is taken. The plasma is treated as a collection of particles of different type with a distribution function, f_s, for each particle of type s. Most commonly, one has two species, positive ions and negative elections. When no collisions are included, the resulting model is of advection type, known as the Vlasov Equation. When collisions are taken into account, the model is a Boltzman Transport equation,

$$\frac{\partial f_s}{\partial t} + \mathbf{v} \cdot \nabla f_s + \mathbf{a}(\mathbf{x}, \mathbf{v}) \cdot \nabla_v f_s = \mathcal{C}, \tag{2.1}$$

where f_s is the distribution of particles of type s, $\mathbf{v}$ is the velocity of the charged fluid, $\mathbf{a}(\mathbf{x}, \mathbf{v})$ is an acceleration or force, and $\mathcal{C}$ is the collision term if it exists. In this context, the acceleration force, $\mathbf{a}(\mathbf{x}, \mathbf{v})$, commonly used is the Lorentz force [36]. This force involves the electric field, $\mathbf{E}$, the magnetic field, $\mathbf{B}$, and an electron's mass, m, and charge q:

$$\mathbf{a}(\mathbf{x}, \mathbf{v}) = \frac{q}{m}(\mathbf{E} + \frac{\mathbf{v}}{c} \times \mathbf{B}). \tag{2.2}$$

The transport equation is a very accurate description of a plasma if all types of particles and collisions are included. However, it is a 7-dimensional hyperbolic conservative equation and can be difficult to solve efficiently. This type of problem has been solved, though, using particle-in-cell and conservative finite volume methods [1, 23, 24]. These methods require a large amount of computation.

Another approach is to take velocity moments of these equations, which yields an infinite hierarchy of equations in just 4 dimensions (space and time). A truncation, or "closure," of this hierarchy is used to get a two-fluid theory for the plasma. This is, in effect, averaging the quantities of motion, and the system is treated as a fluid of positive ions and negative electrons. In more detail, if one were to take the zeroth velocity moment of equation (2.1),

$$\int (2.1) d\mathbf{v}, \tag{2.3}$$

one obtains a continuity equation for the density of each type of particle,

$$\frac{\partial n_s(\mathbf{x}, t)}{\partial t} + \nabla \cdot (n_s \mathbf{V}_s(\mathbf{x}, t)) = 0. \tag{2.4}$$

Here, n_s is the particle density in space alone, $n_s = \int f_s d\mathbf{v}$, and $\mathbf{V}_s = \frac{1}{n_s} \int \mathbf{v} f_s d\mathbf{v}$ is the fluid velocity of species, s. In an incompressible plasma, n_s is constant in time and space and we get the incompressibility constraint, $\nabla \cdot \mathbf{V}_s = 0$.

Taking the the first velocity moment of the transport equation,

$$\int (2.1) \mathbf{v} d\mathbf{v}, \tag{2.5}$$

yields a momentum equation for each type of particle equivalent to Navier-Stokes for each species, s. Taking a second velocity moment would yield an energy equation. A more detailed derivation of these fluid equations can be found in [18, 31, 34, 36]. It should be noted that for every velocity moment of the transport

equation taken, one obtains a higher velocity moment in the resulting equation. For example, the continuity equation obtains a first-order velocity moment, $\mathbf{V}_s$. The momentum equation obtains a second-order velocity moment and the energy equation a third-order moment. Therefore, a closure is needed to truncate the system of equations. In the context of this thesis, the closure occurs at the second velocity moment, which we incorporate into the pressure term in the momentum equation. Using this fluid theory with the pressure closure misses certain velocity-dependent behavior such as Landau damping, which describes the exchange of energy between waves and particles [36]. However, for overall effects of plasmas, such as instabilities in fusion reaction devices, the fluid theory is sufficient and much easier to solve.

Finally, when these fluids are averaged together, we arrive at the magneto-hydrodynamic (MHD) equations. These equations treat the plasma as one fluid of charged particles. Hence, the equations that describe the plasma are Navier-Stokes coupled with Maxwell's equations. These can be derived directly from kinetic theory by the velocity moment method described above [36]. A form of MHD, known as Extended MHD, introduces new terms that allow the fluid to be treated, once again, with a two-fluid theory. We do not address extended MHD in this thesis, since we are still able to study many instabilities that arise in tokamak physics without it.

2.1 Incompressible, Resistive MHD

As stated above, the system is a coupling of the incompressible Navier-Stokes equations and Maxwell's equations. The primitive variables are defined to be the fluid velocity, $\mathbf{u}$, the fluid pressure, p, the magnetic field, $\mathbf{B}$, the current density, $\mathbf{j}$, and the electric field, $\mathbf{E}$.

With ρ, the mass density, $-q$, the charge of an electron, and ν, the viscosity, the Navier-Stokes part of the coupled equations is

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{\mathbf{j} \times \mathbf{B}}{\rho} + \frac{q\mathbf{E}}{\rho} - \frac{\nabla p}{\rho} + \nu \nabla^{\mathbf{2}} \mathbf{u}, \tag{2.6}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{2.7}$$

With μ_0, the permeability of free space, ρ_c, the charge density, and c, the speed of light, the Maxwell's part is

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}, \tag{2.8}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{j} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t}, \tag{2.9}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.10}$$

$$\nabla \cdot \mathbf{E} = \rho_c. \tag{2.11}$$

In most practical applications of MHD, only low-frequency behavior is studied and, thus, the $\frac{q}{\rho}\mathbf{E}$ term can be ignored in equation (2.6) as well as the displacement current, $\frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t}$, in equation (2.9) [36, 48]. In addition, a resistive form of Ohm's law,

$$\mathbf{j} = \sigma(\mathbf{E} + \mathbf{u} \times \mathbf{B}), \tag{2.12}$$

is used to eliminate the electric field, $\mathbf{E}$, from the equations. In general, there are a variety of forms for Ohm's Law, which take different physics into account. These other forms result in a different set of MHD equations, including Ideal MHD, Hall MHD, or Extended MHD. The resistive form that is used in this thesis allows for conductivity to be present in the system resulting in resistive instabilities that can be simulated.

After a non-dimensionalization, using the Alfvén speed, v_A, the Alfvén time, τ_A, and a characteristic length scale, L, the following equations for incompressible resistive MHD are obtained (i.e., Navier-Stokes coupled with Maxwell's equations) [36, 48]:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \mathbf{j} \times \mathbf{B} + \nabla p - \frac{1}{R_e} \nabla^2 \mathbf{u} = \mathbf{f}, \tag{2.13}$$

$$\frac{\partial \mathbf{B}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u} + \frac{1}{S_L} (\nabla \times \mathbf{j}) = \mathbf{g}, \tag{2.14}$$

$$\nabla \times \mathbf{B} = \mathbf{j}, \tag{2.15}$$

$$\nabla \cdot \mathbf{B} = 0, \tag{2.16}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2.17}$$

$$\nabla \cdot \mathbf{j} = 0. \tag{2.18}$$

Here, $R_e = \frac{v_A L}{\nu}$ is the fluid Reynolds Number and is a ratio of the fluid speed to the viscosity of the fluid. Similarly, $S_L = \sigma \mu_0 v_A L$ is the Lundquist Number and is a ratio of the fluid speed to the resistivity of the system, since the conductivity is inversely proportional to the resistivity. Both numbers are assumed to be constant with respect to time and space and are adjusted for different types of physical behavior.

The above set of equations, (2.13)-(2.18), form a coupled system of nonlinear second-order partial differential equations. The next few chapters of this thesis discuss how we solve these equations in order to study a few test problems arising in tokamak plasma physics. First, we describe the FOSLS formulation and finite element discretization.

Chapter 3

Discretization Methods

3.1 FOSLS

To approximate a solution to the nonlinear system of equations, a first-order system least squares (FOSLS) finite element discretization is used [14, 15]. The system is first put into a differential first-order system of equations, denoted by $\mathcal{L}(u) = f$. This system is then linearized using a standard Newton method. In this section, we first consider a generic linear first-order system of equations, denoted by $Lu = f$, in order to explain the FOSLS methodology. Here, L is a mapping from an appropriate Hilbert space, $\mathcal{V}$, to an L^2 product space. In the context of this thesis, we choose $\mathcal{V}$ to be an H^1 product space with appropriate boundary conditions. The boundary conditions are chosen to satisfy the physical constraints of the problem as well as the assumptions needed for the FOSLS framework.

This linear problem is now recast as the minimization of a functional constructed by taking the L^2 norm of the residual of each equation in the system. This is written as

$$u_* = \arg\min_{u \in \mathcal{V}} G(u; f) := \arg\min_{u \in \mathcal{V}} ||Lu - f||_0^2, \tag{3.1}$$

where u_* is the solution in an appropriate Hilbert space $\mathcal{V}$.

Thus, u_* satisfies $G'(u_*)[v] = 0$, which is the Fréchet derivative of G in the direction $v \in \mathcal{V}$. The Fréchet derivative is obtained by the following:

$$G'(u_*)[v] = \lim_{\alpha \to 0} \frac{G(u + \alpha v; f) - G(u; f)}{\alpha}. \tag{3.2}$$

.

This results in the weak form of the problem:

find $u_* \in \mathcal{V}$ such that

$$< Lu_*, Lv > = < f, Lv > \qquad \forall v \in \mathcal{V}, \tag{3.3}$$

where $< \cdot, \cdot >$ is the usual L^2 inner product on the product space, $(L^2)^k$, for k equations in the linear system. If one assumes the following properties of the bilinear form, $< Lu, Lv >$:

$\exists$ constants, c_1 and c_2, such that

$$\text{continuity} \qquad < Lu, Lv > \ \leq \ c_2 ||u||_{\mathcal{V}} ||v||_{\mathcal{V}} \qquad \forall\, u, v \in \mathcal{V}, \tag{3.4}$$

$$\text{coercivity} \qquad < Lu, Lu > \ \geq \ c_1 ||u||^2_{\mathcal{V}} \qquad \forall\, u \in \mathcal{V}, \tag{3.5}$$

then, by the Riesz Representation Theorem [29], this bilinear form is an inner product on $\mathcal{V}$. In addition, these properties imply the existence of a unique solution, $u_* \in \mathcal{V}$, for the weak problem (3.3). Here, c_1 and c_2 depend only on the operator, L, and the domain of the problem. They are independent of u and v.

Next, we approximate u_* by restricting (3.1) to a finite-dimensional space, $\mathcal{V}^h \subseteq \mathcal{V}$, which leads to (3.3) restricted to $\mathcal{V}^h$. Since $\mathcal{V}^h$ is a subspace of $\mathcal{V}$, the discrete problem is also well-posed. Choosing an appropriate basis, $\mathcal{V}^h = span\{\Phi_j\}$,

and restricting (3.3) to this basis, yields an algebraic system of equations involving the matrix, A, with elements

$$(A)_{ij} =< L\Phi_j, L\Phi_i > .$$

Again, from [50], it has been shown that, in the context of a symmetric positive definite (SPD) H^1-equivalent bilinear form restricted to a finite element subspace, a multilevel technique exists that yields optimal convergence to the linear system. In particular, AMG has been shown to work well on a wide range of such problems [4, 5, 7, 21, 8, 12, 14, 15, 19, 38, 43, 45, 49].

In addition, note that the functional yields an a posteriori error measure. Define the error of an element in the finite element subspace, u^h, to be $e^h = u^h - u_*$. If $u^h \in \mathcal{V}^h$, then, with (3.1) restricted to $\mathcal{V}^h$,

$$G(u^h; f) = ||Lu^h - f||_0^2 = ||Lu^h - Lu_*||_0^2 = ||Le^h||_0^2. \tag{3.6}$$

Now, with (3.4) and (3.5) restricted to $\mathcal{V}^h$, we know that $||Le^h||_0^2$ is equivalent to $||e^h||_{\mathcal{V}}^2$. Thus, the functional value is equivalent to the error measured in the Hilbert space norm. If (3.4) and (3.5) hold, then the bilinear form is an inner product, which induces a norm. In general, this is a semi-norm on a subdomain. The local semi-norm is what we use for a local a posteriori error estimator. This property of FOSLS helps make it possible to efficiently solve complex systems. At each step in the solution algorithm, a local measure of the functional is available. This allows judgements to be made based on estimates of the increase of accuracy that results from an increase in computational cost. More is said about this in Chapter 5. In general, the constants, c_2 and c_1, depend on properties of the PDE as well as on boundary conditions and the computational domain. Addressing these issues would require extensive analysis and has been studied in other problems for Navier-Stokes and Maxwell's equations such as in [4, 5, 7, 21, 28].

3.2 Linearization (Newton-FOSLS)

Since the MHD system is nonlinear, we first linearize it, and then put it into a FOSLS weak form. By taking a Newton step, we approximate the nonlinear system by

$$\tilde{\mathcal{L}}(u_0 + \hat{u}) = \mathcal{L}(u_0) + \mathcal{L}'(u_0)[\hat{u}], \tag{3.7}$$

where u_0 is the current approximation and $\mathcal{L}'(u_0)[\hat{u}]$ is the Fréchet derivative in the direction $\hat{u}$. The linearized functional around u_0 then becomes

$$G(u_0 + \hat{u}; f) = ||\mathcal{L}'(u_0)[\hat{u}] - (f - \mathcal{L}(u_0))||_0^2. \tag{3.8}$$

Minimization of the linearized functional yields $\hat{u}_*$ that satisfies the weak form:

find $\hat{u}_* \in \mathcal{V}$ such that,

$$< \mathcal{L}'(u_0)[\hat{u}_*], \mathcal{L}'(u_0)[\hat{v}] >=< (f - \mathcal{L}(u_0)), \mathcal{L}'(u_0)[\hat{v}] > \qquad \forall\ \hat{v} \in \mathcal{V}. \tag{3.9}$$

Once $\hat{u}$ is found, it is added to the previous iterate to get the next approximation:

$$u_1 = u_0 + \hat{u}. \tag{3.10}$$

This constitutes one Newton iteration. In general, Newton's method for solving $\mathcal{L}(u) = f$ would produce the following abstract iteration:

$$u_{n+1} = u_n - (\mathcal{L}'(u_n))^{-1}(f - \mathcal{L}(u_n)). \tag{3.11}$$

However, to obtain a discrete form of (3.11), we first apply the FOSLS approach to the linearized system. We restrict u to $\mathcal{V}^h \subset \mathcal{V}$ in (3.9) and, if the linearized functional retains continuity and coercivity, (3.4) and (3.5), it also retains the desirable properties as described for the linearized system. In addition, the nonlinear functional, that is, the L^2 norm of the residual of the nonlinear system, denoted by

$\mathcal{G}(u; f)$, can be computed as well as the linearized functional, that is, the L^2 norm of the linearized system, equation (3.8). This allows the two functional values to be compared after an approximate solution to the linearized problem is computed and helps determine if the Newton iterations are converging as expected.

In general, we want to minimize the full nonlinear functional,

$$\mathcal{G}(u; f) = ||\mathcal{L}(u) - f||_0^2. \tag{3.12}$$

Linearizing the functional and minimizing this directly involves finding the gradient of the functional and setting this to zero. Computing the Fréchet derivative of $\mathcal{G}(u; f)$ yields

$$\mathcal{G}'(u)[v] = 2 < \mathcal{L}(u) - f, \mathcal{L}'(u)[v] > . \tag{3.13}$$

.

Rearranging this inner product, we write it as $< (\mathcal{L}'(u))^*(\mathcal{L}(u) - f), v >$, where $(\mathcal{L}'(u))^*$ is the adjoint operator to the linearized system of equations. Thus, we define the gradient to be

$$\nabla\mathcal{G} = (\mathcal{L}'(u))^*(\mathcal{L}(u) - f). \tag{3.14}$$

Restricting (3.14) to our finite element space, $\mathcal{V}$, the minimum of the functional, $\mathcal{G}^h(u; f)$, is obtained by finding a u^h such that $(\mathcal{L}'(u^h))^*$ is perpendicular to $\mathcal{L}(u^h) - f$. This approach, which we call FOSLS-Newton, can be more complicated. However, by linearizing the operator first and then applying FOSLS, which we call Newton-FOSLS, produces a similar result. Here, we minimize the linearized functional, $G(u_0 + \hat{u}; f)$, and get the FOSLS weak form as in (3.9). Again, we must restrict to our finite element space. Minimzing (3.8) and restricting to $\mathcal{V}^h$ is equivalent to finding the update

$$\hat{u}^h = ((\mathcal{L}'(u^h))^* \mathcal{L}'(u^h))^{-1} (\mathcal{L}'(u^h)^* \mathcal{L}'(u^h)(f - \mathcal{L}(u^h)). \tag{3.15}$$

Notice that if our Newton iterations have converged, then we have reached a fixed point and the update should be zero. This is equivalent to having $(\mathcal{L}'(u^h)^* \mathcal{L}'(u^h)(f - \mathcal{L}(u^h)) = 0$, or to $(\mathcal{L}'(u^h))^*$ being perpendicular to $\mathcal{L}(u^h) - f$. This is the exact result obtained from applying FOSLS-Newton directly. Thus, as the approximation approaches the solution, the two iterates from the two methods tend to be the same. FOSLS-Newton may be more robust in some applications, but the Newton-FOSLS approach is simpler and has been successful in a number of applications [19, 20]. We confine our presentation to it in this thesis.

In the next chapter, we show how the MHD equations can be rewritten to make them amenable to solution by the Newton-FOSLS approach.

Chapter 4

FOSLS Formulation

4.1 A Current-Vorticity Formulation for Reduced Resistive MHD

The first task is to write the MHD equations as a first-order system, which we do based on a vorticity-velocity-pressure formulation for the Navier-Stokes part of the equations [28]. This formulation has fewer dependent variables than the grad-velocity-pressure formulation described in [4, 5] and is chosen because the test problems we examine provide explicit vorticity boundary conditions. These boundary conditions make the vorticity form of the Navier-Stokes part of the functional H^1 elliptic along regular branches of the solution. It has been shown in [6] that the vorticity formulation for Stokes alone is coercive with the proper boundary conditions, while it is not coercive for velocity boundary conditions. From prior work on Maxwell's equations, it is also known that a current-magnetic field formulation is H^1 elliptic [7]. Thus, we believe that this combined formulation, which we denote the current-vorticity formulation, is a good choice for the full MHD system. For the rest of this thesis, we assume explicit vorticity and current boundary conditions on rectangular domains.

Vorticity, $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, is introduced and, due to incompressibility ($\nabla \cdot \mathbf{u} = 0$), it is noted that

$$\nabla^2 \mathbf{u} = -\nabla \times \nabla \times \mathbf{u} = -\nabla \times \boldsymbol{\omega}.$$

Also,

$$\mathbf{u}\cdot\nabla\mathbf{u} = \frac{1}{2}\nabla|\mathbf{u}|^2 - \mathbf{u}\times(\nabla\times\mathbf{u}) = \frac{1}{2}\nabla|\mathbf{u}|^2 - \mathbf{u}\times\boldsymbol{\omega}.$$

Define the total pressure as

$$\tilde{p} = p + \frac{1}{2}|\mathbf{u}|^2.$$

Then, the incompressible, resistive MHD equations in 3D can be rewritten as

$$\begin{aligned}
\frac{\partial\mathbf{u}}{\partial t} - \mathbf{u}\times\boldsymbol{\omega} - \mathbf{j}\times\mathbf{B} + \nabla(\tilde{p}) + \frac{1}{R_e}\nabla\times\boldsymbol{\omega} &= \mathbf{f}, && (4.1)\\
\frac{\partial\mathbf{B}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{B} - \mathbf{B}\cdot\nabla\mathbf{u} + \frac{1}{S_L}\nabla\times\mathbf{j} &= \mathbf{g}, && (4.2)\\
\nabla\times\mathbf{B} - \mathbf{j} &= \mathbf{0}, && (4.3)\\
\boldsymbol{\omega} - \nabla\times\mathbf{u} &= 0, && (4.4)\\
\nabla\cdot\boldsymbol{\omega} &= 0, && (4.5)\\
\nabla\cdot\mathbf{B} &= 0, && (4.6)\\
\nabla\cdot\mathbf{u} &= 0, && (4.7)\\
\nabla\cdot\mathbf{j} &= 0. && (4.8)
\end{aligned}$$

4.1.1 Linearized equations

Given an approximate solution, u^n, we write the true solution as $u^n + \hat{u}$, plug this into the system of equations, and discard any quadratic terms in $\hat{u}$. This yields the linearized system

$$\begin{aligned}
\frac{\partial\hat{\mathbf{u}}}{\partial t} - \hat{\mathbf{u}}\times\boldsymbol{\omega}^n - \mathbf{u}^n\times\hat{\boldsymbol{\omega}} - \mathbf{j}^n\times\hat{\mathbf{B}} - \hat{\mathbf{j}}\times\mathbf{B}^n + \nabla(\tilde{p}) + \frac{1}{R_e}\nabla\times\hat{\boldsymbol{\omega}} &= \mathbf{f}^n, && (4.9)\\
\frac{\partial\hat{\mathbf{B}}}{\partial t} - \mathbf{B}^n\cdot\nabla\hat{\mathbf{u}} + \hat{\mathbf{u}}\cdot\nabla\mathbf{B}^n - \hat{\mathbf{B}}\cdot\nabla\mathbf{u}^n + \mathbf{u}^n\cdot\nabla\hat{\mathbf{B}} + \frac{1}{S_L}\nabla\times\hat{\mathbf{j}} &= \mathbf{g}^n, && (4.10)
\end{aligned}$$

$$\nabla \times \hat{\mathbf{B}} - \hat{\mathbf{j}} = -\nabla \times \mathbf{B}^n + \mathbf{j}^n, \tag{4.11}$$

$$\hat{\boldsymbol{\omega}} - \nabla \times \hat{\mathbf{u}} = -\boldsymbol{\omega}^n + \nabla \times \mathbf{u}^n, \tag{4.12}$$

$$\nabla \cdot \hat{\boldsymbol{\omega}} = -\nabla \cdot \boldsymbol{\omega}^n, \tag{4.13}$$

$$\nabla \cdot \hat{\mathbf{B}} = -\nabla \cdot \mathbf{B}^n, \tag{4.14}$$

$$\nabla \cdot \hat{\mathbf{u}} = -\nabla \cdot \mathbf{u}^n, \tag{4.15}$$

$$\nabla \cdot \hat{\mathbf{j}} = -\nabla \cdot \mathbf{j}^n, \tag{4.16}$$

where $\mathbf{f}^n$ and $\mathbf{g}^n$ have been modified appropriately with the linearized residuals. The first-order operator can then be written as

$$Lu = \left(\begin{array}{cc|cc|cc|cc}
\nabla\times & -\nabla & -I & 0 & 0 & 0 & 0 & 0 \\
\nabla\cdot & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
\frac{1}{dt}I + \boldsymbol{\omega}^n\times & 0 & \frac{1}{R_e}\nabla\times - \mathbf{u}^n\times & \nabla & -\mathbf{j}^n\times & 0 & \mathbf{B}^n\times & 0 \\
0 & 0 & \nabla\cdot & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & \nabla\times & -\nabla & -I & 0 \\
0 & 0 & 0 & 0 & \nabla\cdot & 0 & 0 & 0 \\
\hline
-\mathbf{B}^n\cdot\nabla + \nabla\mathbf{B}^n\cdot & 0 & 0 & 0 & \frac{1}{dt}I - \nabla\mathbf{u}^n\cdot + \mathbf{u}^n\cdot\nabla & 0 & \frac{1}{S_L}\nabla\times & -\nabla \\
0 & 0 & 0 & 0 & 0 & 0 & \nabla\cdot & 0
\end{array}\right)
\left(\begin{array}{c}
\hat{\mathbf{u}} \\ \alpha \\ \hat{\boldsymbol{\omega}} \\ \tilde{p} \\ \hat{\mathbf{B}} \\ \beta \\ \hat{\mathbf{j}} \\ \gamma
\end{array}\right). \tag{4.17}$$

where α, β, and γ are slack variables that are introduced here to make the system square and are chosen with appropriate boundary conditions, which, along with the form of the right-hand side, ensure that they are equal to zero. Notice that the system is composed of a sequence of curl-grad-div systems along the main diagonal. Any curl-grad-div system with two boundary conditions for the 4 by 4 block can be shown to be H^1 equivalent. Thus, by choosing the boundary conditions on the slack variables correctly and with the known vorticity and current boundary conditions, we obtain an H^1 equivalent system for the diagonal blocks. Also, most of the off-diagonal-block terms are either zero or zeroth-order, that

is, they do not involve derivatives. If all of the off-diagonal-block terms were zeroth-order and if we could show that the infinity norm of each previous iterate (i.e., $||\mathbf{u}^n||_\infty$) is bounded, then, because of the two boundary conditions in each block, it would again be straightforward to establish H^1 equivalence of the FOSLS functional. However, the off-diagonal-block derivatives that appear in the last block-row complicate the analysis and H^1 equivalence needs to be shown. This is done later in this chapter.

4.1.2 Scalings

In this section, we describe a scaling of the system, (4.1)-(4.8), to make it more amenable to solution by iterative solvers, specifically Algebraic Multigrid (AMG). The scaling we describe does not change the finite element approximation, but rather the character of the algebraic system of equations. We can either rescale the specific variables or each equation. The former is called a right scaling of the system, whereas the latter is called a left scaling. We choose to do the scaling with a constant across the domain. Therefore, scaling on the right does not change the choice of function from the finite element space. However, scaling on the left essentially changes the norm in which we are minimizing, which does change the choice of function from the finite element space. Our goal is to find the scaling that gives the best AMG convergence. We show below that this turns out to be a right scaling and, thus, we need not worry about the effects of a left scaling.

Since the MHD system is a coupled system of Navier-Stokes and Maxwell's, it is broken up into the two pieces and scalings are analyzed for each. Note that the nonlinear cross-coupling terms, $\mathbf{j} \times \mathbf{B}$ and $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$, are the terms that describe the stiffest MHD waves and are, therefore, important to the physics. These two terms are complimentary forces of the magnetic field interacting with the fluid. Thus, we must always consider the contribution of these two terms in

any analysis of the system. For this section, we will be testing how well AMG performs on the linear system. We would like to compare scalings of each block of the system with and without these terms added. In the end, these terms will be considered, but the goal of this section is to determine what effect they have on the linear solvers.

For now, consider a 2D system since this is easier to analyze and no slack variables are needed. See Chapter 6 for a derivation of the 2D equations and note that, in this context, ω and j become scalar variables. The hat notation is now dropped.

Maxwell's equations uncoupled are

$$\begin{pmatrix} -dy & dx & -1 \\ dx & dy & 0 \\ \frac{1}{dt} & 0 & \frac{1}{S_L}dy \\ 0 & \frac{1}{dt} & -\frac{1}{S_L}dx \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ j \end{pmatrix}. \tag{4.18}$$

Navier-Stokes equations uncoupled are

$$\begin{pmatrix} -dy & dx & -1 & 0 \\ dx & dy & 0 & 0 \\ \frac{1}{dt} & -\omega^n & \frac{1}{R_e}dy - u_2^n & dx \\ \omega^n & \frac{1}{dt} & -\frac{1}{R_e}dx + u_1^n & dy \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \omega \\ p \end{pmatrix}. \tag{4.19}$$

The full MHD system, (4.17), has the added coupling, $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$, in the last two equations of the Maxwell block, (4.18), and $-\mathbf{j} \times \mathbf{B}$ in the last two equations of the Navier-Stokes block, (4.19). In the FOSLS setting, we are, in effect, solving a weak form of L^*L. Therefore, we would like to understand the structure of this L^*L system. This is called the "formal normal" and, ideally, this is a simple block-diagonal system, such as a set of uncoupled Poisson equations.

The goal is to find a scaling that achieves a nice structure to the formal normal and gives the best AMG convergence.

To study the components, the discrete matrix, A, is created from the Newton-FOSLS discretization on a bi-quadratic tensor product mesh for each part of the system separately. The linearization is done around a known equilibrium state and all right-hand sides are zero. One Backward-Euler timestep is taken and the convergence of AMG on this mock system is observed. The iterations are stopped when the linear system residual is reduced by a factor of 10^{-6}. As an effective convergence factor, $\left(\frac{F_{final}}{F_{initial}}\right)^{\frac{1}{\text{V-cycles}}}$ is computed, where F_{final} is the final and $F_{initial}$ is the initial FOSLS functional value. This allows one to see how the error is being reduced and at what cost. We compute the reduction in the error and raise this to one over the work. In this case, the work is the number of AMG V-cycles needed to reduce the error by the prescribed tolerance. Using this, we test AMG convergence on the Maxwell block alone, the Navier-Stokes block alone, and on hybrid systems where we introduce pieces of the MHD coupling back into each block. For all the following tests, $R_e = S_L = 1000$ and the timestep $dt = 1$.

4.1.2.1 Maxwell's

For the Maxwell system, consider the following scalings. We first scale the columns of the Maxwell block so that $\mathbf{B} \to \dfrac{1}{\sqrt{S_L}}\mathbf{B}$ and $\mathbf{j} \to \sqrt{S_L}\mathbf{j}$:

$$L_1^M = \begin{pmatrix} -\frac{1}{\sqrt{S_L}}dy & \frac{1}{\sqrt{S_L}}dx & -\sqrt{S_L} \\ \frac{1}{\sqrt{S_L}}dx & \frac{1}{\sqrt{S_L}}dy & 0 \\ \frac{1}{\sqrt{S_L}} & 0 & \frac{1}{\sqrt{S_L}}dy \\ 0 & \frac{1}{\sqrt{S_L}} & -\frac{1}{\sqrt{S_L}}dx \end{pmatrix}. \qquad \text{(Scaling 1)}$$

This leads to the formal normal

$$L_1^{M*}L_1^M = \begin{pmatrix} \frac{1}{S_L} - \frac{1}{S_L}\Delta & 0 & \frac{1}{S_L}dy - dy \\ 0 & \frac{1}{S_L} - \frac{1}{S_L}\Delta & dx - \frac{1}{S_L}dx \\ dy - \frac{1}{S_L}dy & \frac{1}{S_L}dx - dx & S_L - \frac{1}{S_L}\Delta \end{pmatrix}.$$

This system, however, can easily be made diagonal by scaling the last two rows with $\sqrt{S_L}$. Let

$$L_2^M = \begin{pmatrix} -\frac{1}{\sqrt{S_L}}dy & \frac{1}{\sqrt{S_L}}dx & -\sqrt{S_L} \\ \frac{1}{\sqrt{S_L}}dx & \frac{1}{\sqrt{S_L}}dy & 0 \\ 1 & 0 & dy \\ 0 & 1 & -dx \end{pmatrix}, \qquad \text{(Scaling 2)}$$

which yields the formal normal

$$\begin{pmatrix} 1 - \frac{1}{S_L}\Delta & 0 & 0 \\ 0 & 1 - \frac{1}{S_L}\Delta & 0 \\ 0 & 0 & S_L - \Delta \end{pmatrix}.$$

Equations	V-cycs	Avg Residual Reduction Factor	Effective Conv. Factor
Scaling 1	60	0.84	0.826
Scaling 2	5	0.04	0.127

Table 4.1: Maxwell system with no nonlinear coupling. $S_L = 1000$

Table 4.1 shows that the latter scaling produces good results as expected, whereas the first one does not do as well.

Now, if the nonlinear cross-coupling term, $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$, is introduced, we can see how this affects AMG on the Maxwell system. A fixed equilibrium state is used for the values of $\mathbf{u}^n$. The Maxwell block now looks like

$$\begin{pmatrix} -dy & dx & -1 \\ dx & dy & 0 \\ \frac{1}{dt} + \mathbf{u}^n \cdot \nabla - u_{1x}^n & -u_{1y}^n & \frac{1}{S_L} dy \\ -u_{2x}^n & \frac{1}{dt} + \mathbf{u}^n \cdot \nabla - u_{2y}^n & -\frac{1}{S_L} dx \end{pmatrix} \begin{pmatrix} B_1 \\ B_2 \\ j \end{pmatrix}. \tag{4.20}$$

All scalings lead to a complicated formal normal, which is no longer diagonal. In addition, the size of the finite element approximations to the linearized solution, $\mathbf{u}^n$, will now affect the structure of the formal normal and how well AMG performs. See table 4.2.

Equations	V-cycs	Avg Residual Reduction Factor	Effective Conv. Factor
Scaling 1	72	0.86	0.895
Scaling 2	17	0.5	0.694

Table 4.2: Maxwell system with nonlinear coupling added. $S_L = 1000$

While AMG convergence deteriorated slightly for Scaling 2, AMG still performs better than for Scaling 1. However, it is interesting to note that the introduction of the coupling term does not affect the performance of Scaling 1 very much. Since this latter system has the appropriate coupling present in the full MHD system, this systems is closer to the one we would like to solve. This indicates that Scaling 1 might be the appropriate choice.

4.1.2.2 Navier-Stokes

Next, consider Navier-Stokes. First, we scale the columns such that $\mathbf{u} \to \frac{1}{\sqrt{R_e}}\mathbf{u}$, $\boldsymbol{\omega} \to \sqrt{R_e}\boldsymbol{\omega}$, and $p \to -\sqrt{R_e}p$. We obtain

$$L_1^N = \begin{pmatrix} -\frac{1}{\sqrt{R_e}}dy & \frac{1}{\sqrt{R_e}}dx & -\sqrt{R_e} & 0 \\ \frac{1}{\sqrt{R_e}}dx & \frac{1}{\sqrt{R_e}}dy & 0 & 0 \\ \frac{1}{\sqrt{R_e}} & -\omega^n & \frac{1}{\sqrt{R_e}}dy - u_2^n & -\sqrt{R_e}dx \\ \omega^n & \frac{1}{\sqrt{R_e}} & -\frac{1}{\sqrt{R_e}}dx + u_1^n & -\sqrt{R_e}dy \end{pmatrix}. \qquad \text{(Scaling 1)}$$

This leads to the following formal normal. For simplicity, we set $\mathbf{u}^n = \omega^n = 0$ to get

$$L_1^{N*}L_1^N = \begin{pmatrix} \frac{1}{R_e}(1-\Delta) & 0 & \frac{1}{R_e}dy - dy & -dx \\ 0 & \frac{1}{R_e}(1-\Delta) & dx - \frac{1}{R_e}dx & -dy \\ dy - \frac{1}{R_e}dy & \frac{1}{R_e}dx - dx & R_e - \frac{1}{R_e}\Delta & 0 \\ dx & dy & 0 & -R_e\Delta \end{pmatrix}.$$

Introducing the linearized solutions back in adds the following terms to the formal normal:

$$\begin{pmatrix} (\omega^n)^2 & 0 & \frac{1}{\sqrt{R_e}}(u_2^n + \omega^n dx) + u_1^n\omega^n & -\sqrt{R_e}\omega^n dy \\ 0 & (\omega^n)^2 & \frac{1}{\sqrt{R_e}}(u_1^n - \omega^n dy) + u_2^n\omega^n & \sqrt{R_e}\omega^n dx \\ \frac{1}{\sqrt{R_e}}(u_2^n - \omega^n dx) + u_1^n\omega^n & \frac{1}{\sqrt{R_e}}(u_1^n + \omega^n dy) + u_2^n\omega^n & (\mathbf{u}^n)^2 & \sqrt{R_e}(u_2^n dx - u_1^n dy) \\ \sqrt{R_e}\omega^n dy & -\sqrt{R_e}\omega^n dx & \sqrt{R_e}(-u_2^n dx + u_1^n dy) & 0 \end{pmatrix}.$$

A different scaling leads to a simpler formal normal. Here, we let $\mathbf{u} \to R_e^{-1/4}\mathbf{u}$, $\boldsymbol{\omega} \to R_e^{1/4}\boldsymbol{\omega}$, and $p \to -R_e^{-3/4}p$ and then scale the last two rows with $\sqrt{R_e}$. This gives

$$L = \begin{pmatrix} -\frac{1}{R_e^{1/4}}dy & \frac{1}{R_e^{1/4}}dx & -R_e^{1/4} & 0 \\ \frac{1}{R_e^{1/4}}dx & \frac{1}{R_e^{1/4}}dy & 0 & 0 \\ R_e^{1/4} & -\sqrt{R_e}\omega^n & \frac{1}{R_e^{1/4}}dy - \sqrt{R_e}u_2^n & -\frac{1}{R_e^{1/4}}dx \\ \sqrt{R_e}\omega^n & R_e^{1/4} & -\frac{1}{R_e^{1/4}}dx + \sqrt{R_e}u_1^n & -\frac{1}{R_e^{1/4}}dy \end{pmatrix}. \qquad \text{(Scaling 2)}$$

Again, with $\mathbf{u}^n = \omega^n = 0$, we have the formal normal

$$L^*L = \begin{pmatrix} \sqrt{R_e} - \frac{1}{\sqrt{R_e}}\Delta & 0 & 0 & -dx \\ 0 & \sqrt{R_e} - \frac{1}{\sqrt{R_e}}\Delta & 0 & -dy \\ 0 & 0 & \sqrt{R_e} - \frac{1}{\sqrt{R_e}}\Delta & 0 \\ dx & dy & 0 & -\frac{1}{\sqrt{R_e}}\Delta \end{pmatrix}.$$

However, with the nonlinear terms not equal to zero we have the added terms

$$R_e^{\frac{1}{4}} \begin{pmatrix} R_e^{\frac{3}{4}}(\omega^n)^2 & 0 & -R_e^{\frac{1}{2}}u_2^n + \omega^n(R_e^{\frac{3}{4}}u_1^n - dx) & -\omega^n dy \\ 0 & R_e^{\frac{3}{4}}(\omega^n)^2 & R_e^{\frac{1}{2}}u_1^n + \omega^n(R_e^{\frac{3}{4}}u_2^n - dy) & \omega^n dx \\ -R_e^{\frac{1}{2}}u_2^n + \omega^n(R_e^{\frac{3}{4}}u_1^n + dx) & R_e^{\frac{1}{2}}u_1^n + \omega^n(R_e^{\frac{3}{4}}u_2^n + dy) & R_e^{\frac{3}{4}}(\mathbf{u}^n)^2 & (u_2^n dx - u_1^n dy) \\ \omega^n dy & -\omega^n dx & -(u_2^n dx - u_1^n dy) & 0 \end{pmatrix}.$$

Therefore, the performance of AMG will depend on the size of these nonlinear terms. We can see that if they are large, then probably neither scalings will do any better than the other. This is confirmed below.

First, with $\mathbf{u}^n$ and ω^n not equal to zero, we see the results in table 4.3. Then, for the results in table 4.4, we have the $\mathbf{u} \times \boldsymbol{\omega}$ term removed completely.

Equations	V-cycs	Avg Residual Reduction Factor	Effective Conv. Factor
Scaling 1	34	0.74	0.71
Scaling 2	40	0.77	0.83

Table 4.3: Navier-Stokes system with nonlinear coupling added. $R_e = 1000$

Equations	V-cycs	Avg Residual Reduction Factor	Effective Conv. Factor
Scaling 1	34	0.74	0.71
Scaling 2	41	0.78	0.81

Table 4.4: Navier-Stokes system with no nonlinear coupling. $R_e = 1000$

Both scalings are fairly comparable here. Again, it is interesting to note that the introduction of the nonlinear term does not affect the performance of Scaling 1. This leads us to believe that this scaling should handle the coupled system better.

4.1.2.3 Coupled System

Based on the above results, it would appear that combining Scaling 2 for Maxwell's and Scaling 1 for Navier-Stokes should yield the best convergence for the whole system with no coupling. We show all possible combinations for the full MHD system.

First, with the $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$ term removed from the full system, as this is probably the most problematic term, we obtain the results in table 4.5. As expected from above, Scaling 2 for Maxwell's and Scaling 1 for Navier-Stokes works the best. However, with the more problematic $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$ term added back in, table 4.6 shows that Scaling 1 for both blocks works best.

Max Scaling	N/S Scaling	V-cycs	Avg Conv. Factor	Effective Conv. Factor
1	1	75	0.89	0.883
1	2	75	0.90	0.914
2	1	35	0.74	0.726
2	2	75	0.89	0.890

Table 4.5: Maxwell and Navier-Stokes system combined with no coupling. $R_e = S_L = 1000$.

Max Scaling	N/S Scaling	V-cycs	Avg Conv. Factor	Effective Conv. Factor
1	1	51	0.82	0.807
1	2	75	0.90	0.911
2	1	62	0.85	0.852
2	2	75	0.90	0.916

Table 4.6: Fully coupled system. $R_e = S_L = 1000$.

Thus, after trying various scalings, it was determined that what worked

best for the individual Maxwell and Navier-Stokes systems was not the best in the end for the fully coupled system. Scaling 1 produces more consistent results when the nonlinear cross-coupling is added. Since these coupling terms are what drives the physics, specifically at the fastest time scales, this is the scaling we choose. Finally, Scaling 1 was obtained only by a change of variables and not a scaling of the equations. As described above, this does not affect the finite element solution.

4.1.3 Current-Vorticity Model

Our final formulation in 3D is obtained by letting $\mathbf{u} \to \frac{1}{\sqrt{R_e}}\mathbf{u}$, $\boldsymbol{\omega} \to \sqrt{R_e}\boldsymbol{\omega}$, $p \to -\sqrt{R_e}p$, $\mathbf{B} \to \frac{1}{\sqrt{S_L}}\mathbf{B}$, and $\mathbf{j} \to \sqrt{S_L}\mathbf{j}$:

$$\frac{1}{\sqrt{R_e}}\nabla \times \mathbf{u} - \sqrt{R_e}\boldsymbol{\omega} = 0, \tag{4.21}$$

$$\frac{1}{\sqrt{R_e}}\nabla \cdot \mathbf{u} = 0, \tag{4.22}$$

$$\sqrt{R_e}\nabla \cdot \boldsymbol{\omega} = 0, \tag{4.23}$$

$$\frac{1}{\sqrt{R_e}}\frac{\partial \mathbf{u}}{\partial t} - \mathbf{u} \times \boldsymbol{\omega} - \mathbf{j} \times \mathbf{B} - \sqrt{R_e}\nabla p + \frac{1}{\sqrt{R_e}}\nabla \times \boldsymbol{\omega} = \mathbf{f}, \tag{4.24}$$

$$\frac{1}{\sqrt{S_L}}\nabla \times \mathbf{B} - \sqrt{S_L}\mathbf{j} = 0, \tag{4.25}$$

$$\frac{1}{\sqrt{S_L}}\nabla \cdot \mathbf{B} = 0, \tag{4.26}$$

$$\sqrt{S_L}\nabla \cdot \mathbf{j} = 0, \tag{4.27}$$

$$\frac{1}{\sqrt{S_L}}\frac{\partial \mathbf{B}}{\partial t} + \frac{1}{\sqrt{R_e S_L}}(\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}) + \frac{1}{\sqrt{S_L}}\nabla \times \mathbf{j} = \mathbf{g}. \tag{4.28}$$

Here, we have 16 equations and 13 unknowns. The slack variables make up the other three unknowns in the square system, (4.21)-(4.28). Since they are zero, they are not included in any computations.

This formulation we denote as the current-vorticity formulation and it consists of two block matrices of a div/curl system coupled together with the nonlinear terms. Due to this nice symmetry and the specific way in which the nonlinear coupling occurs, we can now prove H^1 equivalence of this first-order system.

4.2 H^1 Ellipticity

To prove H^1 equivalence, we first need to define some notation and prove some lemmas. Here, we also make assumptions on the computational domain and the boundary conditions of the system. Since all of the test problems that are considered in this thesis are posed on simple rectangular domains with standard boundary conditions, we are able to use many standard results. In addition, in the context of this thesis, we assume explicit vorticity and current boundary conditions. This makes proving ellipticity of a curl-grad-div block trivial. We use this fact throughout the following proof.

4.2.1 Preliminaries and Definitions

First, we take the L^2 norm of the residual of our linearized system in 3D, equations (4.17). Again, slack variables, α, β, and γ, are introduced to make the system square and the analysis easier to undertand. The following proof is for the unscaled system. Applying the proof on the scaled 2D system, a linearized version of equations (4.21)-(4.28) would give similar results. The only difference would be in the coercivity constants obtained. The results from section 4.1 show that the scaling should, in fact, improve these constants, although they still naturally depend on the problem parameters, such as the Reynolds and Lundquist numbers. With this in mind, the full functional of the linearized system on the error is defined as

$$G = ||\nabla \times \mathbf{u} - \boldsymbol{\omega} - \nabla\gamma||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2 + ||\nabla \cdot \boldsymbol{\omega}||_0^2 +$$

$$||\frac{1}{\Delta t}\mathbf{u} - \mathbf{u}^n \times \boldsymbol{\omega} + \boldsymbol{\omega}^n \times \mathbf{u} - \mathbf{j}^n \times \mathbf{B} + \mathbf{B}^n \times \mathbf{j} - \nabla p + \frac{1}{R_e}\nabla \times \boldsymbol{\omega}||_0^2$$
$$+||\nabla \times \mathbf{B} - \mathbf{j} - \nabla\alpha||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2 +$$
$$||\frac{1}{\Delta t}\mathbf{B} + (\mathbf{u}^n \cdot \nabla\mathbf{B} + \nabla\mathbf{B}^n \cdot \mathbf{u} - \mathbf{B}^n \cdot \nabla\mathbf{u} - \nabla\mathbf{u}^n \cdot \mathbf{B}) + \frac{1}{S_L}\nabla \times \mathbf{j} - \nabla\beta||_0^2. \quad (4.29)$$

All variables with a superscript n are known values in the finite element space, come from the linearization of the system. Here, the right-hand sides, f and g, have been dropped so that now $\mathcal{U} = (\mathbf{u}, \gamma, \omega, p, \mathbf{B}, \alpha, \mathbf{j}, \beta)^T$ represents the error.

We next break the system into its components, specifically the Maxwell block and the linearized Navier-Stokes block. For each block of these systems, we can prove H^1 ellipticity. We then look at the system as a whole and prove ellipticity with the internal couplings added to the system. We define the fluid variables to be $\mathbf{u}_{FL}$ and the electromagnetism variables to be $\mathbf{B}_{EM}$, with the following block matrices for each system.

The full Maxwell block is

$\mathcal{L}^0_{max}\mathbf{B}_{EM} =$

$$\left(\begin{array}{cc|cc} \nabla\times & -\nabla & I & 0 \\ \nabla\cdot & 0 & 0 & 0 \\ \hline (\frac{1}{\Delta t} - \nabla\mathbf{u}^n)I + \mathbf{u}^n \cdot \nabla & 0 & \frac{1}{S_L}\nabla\times & -\nabla \\ 0 & 0 & \nabla\cdot & 0 \end{array}\right) \left(\begin{array}{c} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{array}\right) \quad (4.30)$$

$$= \left(\begin{array}{c|c} A_0 & I_{1,1} \\ \hline D & A_1 \end{array}\right) \left(\begin{array}{c} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{array}\right).$$

Next, if we remove the zeroth-order terms in the upper triangular block we have

$\mathcal{L}^{(1)}_{max}\mathbf{B}_{EM} =$

$$\left(\begin{array}{c|c} A_0 & 0 \\ \hline D & A_1 \end{array}\right)\begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix}. \tag{4.31}$$

Finally, removing the lower triangular coupling, we get

$\mathcal{L}^{(2)}_{max}\mathbf{B}_{EM} =$

$$\left(\begin{array}{c|c} A_0 & 0 \\ \hline 0 & A_1 \end{array}\right)\begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix}. \tag{4.32}$$

Next, we define the Navier-Stokes block as

$\mathcal{L}^{(0)}_{ns}\mathbf{u}_{FL} =$

$$\left(\begin{array}{cc|cc} \nabla\times & -\nabla & I & 0 \\ \nabla\cdot & 0 & 0 & 0 \\ \hline (\frac{1}{\Delta t} + \boldsymbol{\omega}^n\times)I & 0 & \frac{1}{R_e}\nabla\times -(\mathbf{u}^n\times)I & -\nabla \\ 0 & 0 & \nabla\cdot & 0 \end{array}\right)\begin{pmatrix} \mathbf{u} \\ \gamma \\ \boldsymbol{\omega} \\ p \end{pmatrix}. \tag{4.33}$$

Then the combined full MHD system is

$\mathcal{L}^{(0)}\mathcal{U} =$

$$\left(\begin{array}{cccc|cccc}
 & & & & -\mathbf{j}^n\times & 0 & \mathbf{B}^n\times & 0 \\
 & & & & 0 & 0 & 0 & 0 \\
 & & \mathcal{L}_{ns}^{(0)} & & & & & \\
 & & & & 0 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & & & & \\
0 & 0 & 0 & 0 & & & & \\
 & & & & & \mathcal{L}_{max}^{(0)} & & \\
\nabla\mathbf{B}^n\cdot -\mathbf{B}^n\cdot\nabla & 0 & 0 & 0 & & & & \\
0 & 0 & 0 & 0 & & & &
\end{array}\right)\left(\begin{array}{c}\mathbf{u}\\ \gamma\\ \boldsymbol{\omega}\\ p\\ \mathbf{B}\\ \alpha\\ \mathbf{j}\\ \beta\end{array}\right). \tag{4.34}$$

$$= \left(\begin{array}{c|c} A_2 & K \\ \hline T & A_3 \end{array}\right)\left(\begin{array}{c}\mathbf{u}_{FL}\\ \mathbf{B}_{EM}\end{array}\right). \tag{4.35}$$

Again, we define the submatrices as follows. First, we remove the zeroth-order coupling in the upper triangular portion:

$\mathcal{L}^{(1)} =$

$$= \left(\begin{array}{c|c} A_2 & 0 \\ \hline T & A_3 \end{array}\right). \tag{4.36}$$

Next, we remove the lower triangular portion from this to get

$\mathcal{L}^{(2)} =$

$$= \left(\begin{array}{c|c} A_2 & 0 \\ \hline 0 & A_3 \end{array}\right). \tag{4.37}$$

Now, for all the above systems, we have the corresponding functionals

$$G_j^i = ||\mathcal{L}_j^{(i)}\mathbf{v}||_0^2 \quad i = 0, 1, 2 \quad j = \max, \mathrm{ns}. \tag{4.38}$$

In this way, we can rewrite the full functional, (4.29), as

$$G = ||\mathcal{L}^{(0)} \begin{pmatrix} \mathbf{u}_{FL} \\ \mathbf{B}_{EM} \end{pmatrix} ||_0^2. \tag{4.39}$$

We also need the following definitions:

$$L_0^2(\Omega) = \{p \in L^2(\Omega) : \int_\Omega p dx = 0\},$$

$$H_0^1(\Omega) = \{p \in L^2(\Omega) : \nabla p \in (L^2(\Omega))^3 \text{ and } p = 0 \text{ on } \partial\Omega\},$$

$$H(div, \Omega) = \{\mathbf{v} \in (L^2(\Omega))^3 : \nabla \cdot \mathbf{v} \in L^2(\Omega)\},$$

$$H(curl, \Omega) = \{\mathbf{v} \in (L^2(\Omega))^3 : \nabla \times \mathbf{v} \in (L^2(\Omega))^3\}.$$

$$H_0(div, \Omega) = \{\mathbf{v} \in (L^2(\Omega))^3 : \nabla \cdot \mathbf{v} \in L^2(\Omega) \text{ and } n \cdot \mathbf{v} = 0 \text{ on } \partial\Omega\},$$

$$H_0(curl, \Omega) = \{\mathbf{v} \in (L^2(\Omega))^3 : \nabla \times \mathbf{v} \in (L^2(\Omega))^3 \text{ and } n \times \mathbf{v} = 0 \text{ on } \partial\Omega\}.$$

Finally, to prove H^1 ellipticity, we use the following lemmas, some of which are well known results (cf. [14, 15, 26]).

Lemma 1 (Poincaré-Friedrichs Inequality)**.** *For any $p \in L_0^2(\Omega)$, there exists a positive constant, C_1, such that*

$$||p||_0^2 \leq C_1 ||\nabla p||_0^2. \tag{4.40}$$

Lemma 2. *(cf. [26]) Assume that the domain, Ω, has a connected boundary and that it is a bounded convex polyhedron or has a $C^{1,1}$ boundary, which is also simply connected. Then, for any $\mathbf{v} \in H_0(div; \Omega) \cap H(curl; \Omega)$ or $\in H(div; \Omega) \cap H_0(curl; \Omega)$, there exists a positive constant, C_2, such that*

$$||v||_1^2 \leq C_2(||\nabla \cdot \mathbf{v}||_0^2 + ||\nabla \times \mathbf{v}||_0^2). \tag{4.41}$$

Equivalently, this says

$$H_0(div; \Omega) \cap H(curl; \Omega) \subseteq H^1,$$

$$H(div; \Omega) \cap H_0(curl; \Omega) \subseteq H^1.$$

Lemma 3. *Let Ω be as in Lemma 2.*

a. *For $q \in L_0^2(\Omega) \cap H^1$ and $\mathbf{w} \in H(div;\Omega) \cap H_0(curl;\Omega)$, there exists a positive constant, C_{3a}, such that*

$$||q||_1^2 + ||\mathbf{w}||_1^2 \leq C_{3a}(||\nabla \times \mathbf{w} - \nabla q||_0^2 + ||\nabla \cdot \mathbf{w}||_0^2). \tag{4.42}$$

b. *For $\alpha \in H_0^1$ and $\mathbf{v} \in H_0(div;\Omega) \cap H(curl;\Omega)$, there exists a positive constant, C_{3b}, such that*

$$||\alpha||_1^2 + ||\mathbf{v}||_1^2 \leq C_{3b}(||\nabla \times \mathbf{v} - \nabla \alpha||_0^2 + ||\nabla \cdot \mathbf{v}||_0^2). \tag{4.43}$$

Proof. Due to the boundary conditions for both a and b, we have

$$||\nabla \times \mathbf{w} - \nabla q||_0^2 = ||\nabla \times \mathbf{w}||_0^2 + ||\nabla q||_0^2, \tag{4.44}$$

$$||\nabla \times \mathbf{v} - \nabla \alpha||_0^2 = ||\nabla \times \mathbf{v}||_0^2 + ||\nabla \alpha||_0^2, \tag{4.45}$$

respectively.

a. Equations (4.44) and (4.45) imply

$$||\nabla q||_0^2 + ||\nabla \times \mathbf{w}||_0^2 + ||\nabla \cdot \mathbf{w}||_0^2 = ||\nabla \times \mathbf{w} - \nabla q||_0^2 + ||\nabla \cdot \mathbf{w}||_0^2.$$

By Lemma 1, $||q||_1^2 \leq C_1||\nabla q||_0^2$ and, by Lemma 2,

$$||\mathbf{w}||_1^2 \leq C_2(||\nabla \cdot \mathbf{w}||_0^2 + ||\nabla \times \mathbf{w}||_0^2).$$

This yields

$$\begin{aligned} ||q||_1^2 + ||\mathbf{w}||_1^2 &\leq C_1||\nabla q||_0^2 + C_2||\nabla \times \mathbf{w}||_0^2 + C_2||\nabla \cdot \mathbf{w}||_0^2 \\ &\leq C_{3a}||\nabla \times \mathbf{w} - \nabla q||_0^2 + ||\nabla \cdot \mathbf{w}||_0^2. \end{aligned}$$

b. A similar argument works to give

$$||\alpha||_1^2 + ||\mathbf{v}||_1^2 \leq C_{3b}(||\nabla \times \mathbf{v} - \nabla \alpha||_0^2 + ||\nabla \cdot \mathbf{v}||_0^2).$$

□

Lemma 4 (Standard Compactness Argument (cf. [26])). *Let $\mathcal{L} : \mathcal{V} \to \mathcal{W}$ be a continuous bounded linear operator that is one-to-one. Assume $\mathcal{V}$ and $\mathcal{W}$ are Hilbert spaces such that $\mathcal{V}$ is compactly embedded in $\mathcal{W}$. Continuity of $\mathcal{L}$ means that there exists a positive constant, c_4, such that*

$$||\mathcal{L}u||_{\mathcal{W}} \leq c_4||u||_{\mathcal{V}} \ \ \forall u \in \mathcal{V}.$$

Also, assume there exists a positive constant, C_4, such that

$$||u||_{\mathcal{V}} \leq C_4(||\mathcal{L}u||_{\mathcal{W}} + ||u||_{\mathcal{W}}) \ \ \forall u \in \mathcal{V}.$$

Then, there exists a positive constant, $\hat{C}_4$, such that

$$||u||_{\mathcal{V}} \leq \hat{C}_4||\mathcal{L}u||_{\mathcal{W}} \ \ \forall u \in \mathcal{V}. \tag{4.46}$$

Proof. Assume that (4.46) is not true. Then, $\exists$ a sequence, $\{u_j\} \in \mathcal{V}$, such that $||u_j||_{\mathcal{V}} = 1$ and $||\mathcal{L}u_j||_{\mathcal{W}} < \frac{1}{j}$, $j = 1, 2, ...$

Since $\mathcal{V}$ is compactly embedded in $\mathcal{W}$, $\exists$ a subsequence, $\{u_{j_l}\}$, such that the sequence converges to some $u^* \in \mathcal{W}$. In other words,

$$\lim_{l \to \infty} ||u_{j_l} - u^*||_{\mathcal{W}} = 0.$$

Consider two such elements of the subsequence, $\{u_{j_k}\}$ and $\{u_{j_m}\}$; then, for k and m sufficiently large,

$$||u_{j_k} - u_{j_m}||_{\mathcal{V}} \leq C_4(||\mathcal{L}(u_{j_k} - u_{j_m})||_{\mathcal{W}} + ||u_{j_k} - u_{j_m}||_{\mathcal{W}})$$

$$\leq c(||\mathcal{L}u_{j_k}||_{\mathcal{W}} + ||\mathcal{L}u_{j_m}||_{\mathcal{W}} + ||u_{j_k} - u^*||_{\mathcal{W}} + ||u_{j_m} - u^*||_{\mathcal{W}})$$

$$\leq c(\frac{1}{k} + \frac{1}{m} + \epsilon/2 + \epsilon/2)$$

$$\to 0.$$

Thus, $\{u_{j_l}\}$ is Cauchy in $\mathcal{V}$, which implies that it has a limit in $\mathcal{V}$. This, in turn, implies that u^* is in $\mathcal{V}$ and $\lim_{l\to\infty} ||u_{j_l} - u^*||_{\mathcal{V}} = 0$. Then, due to the continuity of $\mathcal{L}$, we have

$$\begin{aligned} ||\mathcal{L}u^*||_{\mathcal{W}} &\leq ||\mathcal{L}(u^* - u_{jl})||_{\mathcal{W}} + ||\mathcal{L}u_{jl}||_{\mathcal{W}} \\ &\leq c_4||u^* - u_{jl}||_{\mathcal{V}} + \frac{1}{l} \\ &\to 0, \end{aligned}$$

which implies

$$\mathcal{L}u^* = 0.$$

Since $\mathcal{L}$ is injective, we see that $u^* = 0$. But $||u^*||_{\mathcal{V}} = 1$ since it is the limit of a sequence with such properties, and we have a contradiction. Therefore, there exists a positive constant, $\hat{C}_4$, such that

$$||u||_{\mathcal{V}} \leq \hat{C}_4||\mathcal{L}u||_{\mathcal{W}}.$$

□

Lemma 5. *Let $\mathcal{L}$ be a 2×2 lower triangular block matrix such that*

$$\mathcal{L}\mathcal{U} = \begin{pmatrix} A_1 & 0 \\ T & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

Let A_1 and A_2 be invertible and let there exist a positive constant, C_5, such that,

$$||T\mathbf{u}_1||_0^2 \leq C_5||A_1\mathbf{u}_1||_0^2.$$

Then,

$$|| \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} ||_0^2 \leq (1 + C_5)|| \begin{pmatrix} A_1 & 0 \\ T & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} ||_0^2.$$

Proof. Note that

$$\begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} A_1^{-1} & 0 \\ -A_2^{-1}TA_1^{-1} & A_2^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}.$$

Then,

$$\begin{aligned} ||\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}||_0^2 &= ||\begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix} \begin{pmatrix} A_1^{-1} & 0 \\ -A_2^{-1}TA_1^{-1} & A_2^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 \\ &= ||\begin{pmatrix} I & 0 \\ -TA_1^{-1} & I \end{pmatrix} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 \\ &\leq ||\mathbf{f}||_0^2 + ||\mathbf{g}||_0^2 + ||TA_1^{-1}\mathbf{f}||_0^2. \end{aligned}$$

Then, by assumption, this is equal to

$$\begin{aligned} &||\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 + ||T\mathbf{u}_1||_0^2 \\ &\leq ||\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 + C_5||A_1\mathbf{u}_1||_0^2 \\ &= ||\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 + C_5||\mathbf{f}||_0^2 \\ &\leq (1+C_5)||\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}||_0^2 \\ &= (1+C_5)||\begin{pmatrix} A_1 & 0 \\ T & A_2 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}||_0^2. \end{aligned}$$

□

With the above definitions and lemmas, we can now prove H^1 equivalence for the Maxwell and Navier-Stokes blocks separately, and then for the coupled system.

4.2.2 Main Theorem

Theorem 6. *Let the functional, G, be defined as in equation (4.29) and let $\mathcal{L}$ be the linearized first order MHD operator as in equation (4.34). Let the domain, Ω, be a bounded convex polyhedron with connected boundary or a simply connected region with a $C^{1,1}$ boundary. Denote $\mathcal{U} = (\mathbf{u}, \gamma, \boldsymbol{\omega}, p, \mathbf{B}, \alpha, \mathbf{j}, \beta)^T$ and let*

$$\mathcal{U} \in \mathcal{V} := [H_0(div;\Omega) \cap H(curl;\Omega)] \otimes [H_0^1(\Omega)] \otimes [H(div;\Omega) \cap H_0(curl;\Omega)]$$

$$\otimes [L_0^2(\Omega) \cap H^1(\Omega)] \otimes [H_0(div;\Omega) \cap H(curl;\Omega)] \otimes [H_0^1(\Omega)]$$

$$\otimes [H(div;\Omega) \cap H_0(curl;\Omega)] \otimes [H_0^1(\Omega)].$$

Assume $\mathbf{u}^n$, $\boldsymbol{\omega}^n$, $\mathbf{B}^n$, and $\mathbf{j}^n$ are known elements of our finite space and, therefore, $||\mathbf{u}^n||_\infty$, $||\nabla \mathbf{u}^n||_\infty$, $||\mathbf{B}^n||_\infty$, $||\nabla \mathbf{B}^n||_\infty$, $||\boldsymbol{\omega}||_\infty$, and $||\mathbf{j}||_\infty < \infty$. Then, there exists positive constants, c_1 and c_2, that yield the following bounds:

a.

$$G \leq c_2(||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2), \quad (4.47)$$

b.

$$c_1(||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2) \leq G. \quad (4.48)$$

These bounds mean that the functional, (4.29), is equivalent to the H^1 norm of the error.

4.2.3 Proof of Upper Bound

Here, we show that for every $\mathcal{U} \in \mathcal{V}$, denoted above, there exists a constant, $c_2 > 0$, such that

$$G \leq c_2||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2. \tag{4.49}$$

Proof. This is easily established by repeated use of Cauchy-Schwarz and the triangle inequality. Again,

$$G = ||\nabla \times \mathbf{u} - \boldsymbol{\omega} - \nabla\gamma||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2 + ||\nabla \cdot \boldsymbol{\omega}||_0^2 +$$
$$||\frac{1}{\Delta t}\mathbf{u} - \mathbf{u}^n \times \boldsymbol{\omega} + \boldsymbol{\omega}^n \times \mathbf{u} - \mathbf{j}^n \times \mathbf{B} + \mathbf{B}^n \times \mathbf{j} - \nabla p + \frac{1}{R_e}\nabla \times \boldsymbol{\omega}||_0^2$$
$$+||\nabla \times \mathbf{B} - \mathbf{j} - \nabla\alpha||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2 +$$
$$||\frac{1}{\Delta t}\mathbf{B} + (\mathbf{u}^n \cdot \nabla\mathbf{B} + \nabla\mathbf{B}^n \cdot \mathbf{u} - \mathbf{B}^n \cdot \nabla\mathbf{u} - \nabla\mathbf{u}^n \cdot \mathbf{B}) + \frac{1}{S_L}\nabla \times \mathbf{j} - \nabla\beta||_0^2.$$

Via some triangle inequalities, we have

$$G \leq (\frac{1}{\Delta t^2} + ||\boldsymbol{\omega}^n||_\infty^2 + ||\nabla\mathbf{B}^n||_\infty^2)||\mathbf{u}||_0^2 + ||\nabla \times \mathbf{u}||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2$$
$$+||\mathbf{B}^n||_\infty^2||\nabla\mathbf{u}||_0^2 + ||\nabla\gamma||_0^2 + (1 + ||\mathbf{u}^n||_\infty^2)||\boldsymbol{\omega}||_0^2 + \frac{1}{R_e^2}||\nabla \times \boldsymbol{\omega}||_0^2$$
$$+||\nabla \cdot \boldsymbol{\omega}||_0^2 + ||\nabla p||_0^2 + (||\mathbf{j}^n||_\infty^2 + \frac{1}{\Delta t^2} + ||\nabla\mathbf{u}^n||_\infty^2)||\mathbf{B}||_0^2$$
$$+||\nabla \times \mathbf{B}||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2 + (||\mathbf{u}^n||_\infty^2)||\nabla\mathbf{B}||_0^2 + ||\nabla\alpha||_0^2 + (1 + ||\mathbf{B}^n||_\infty^2)||\mathbf{j}||_0^2$$
$$+\frac{1}{S_L^2}||\nabla \times \mathbf{j}||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2 + ||\nabla\beta||_0^2.$$

If $||\mathbf{u}^n||_\infty^2$, $||\boldsymbol{\omega}^n||_\infty^2$, $||\mathbf{B}^n||_\infty^2$, $||\mathbf{j}^n||_\infty^2$, $||\nabla\mathbf{u}^n||_\infty^2$, and $||\nabla\mathbf{B}^n||_\infty^2$ are all bounded, then there exists some constant, $c_2 > 0$, such that

$$G \leq c_2(||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2). \tag{4.50}$$

Thus, we have the upper bound on the MHD functional. □

4.2.4 Proof of Lower Bound (Coercivity)

Proof. Here, we prove the lower bound on G. In other words, there exists a positive constant, c_1, such that $G \geq c_1||\mathcal{U}||_1^2$. We do this for each block system, Maxwell's and Navier-Stokes, and then for the entire system.

Maxwell's

First consider coercivity of just the Maxwell's portion. We want to show that there exists a positive constant, c_{max}, such that

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_{max} G_{max}^{(0)}. \tag{4.51}$$

But first we show

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_1 G_{max}^{(2)}. \tag{4.52}$$

Note that

$$G_{max}^{(2)} = ||\nabla \times \mathbf{B} - \nabla\alpha||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2 + ||\frac{1}{S_L}\nabla \times \mathbf{j} - \nabla\beta||_0^2.$$

Then, by Lemma 3, we know there exists positive constants, c_2 and c_3, such that

$$||\alpha||_1^2 + ||\mathbf{B}||_1^2 \leq c_2(||\nabla \times \mathbf{B} - \nabla\alpha||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2)$$

and

$$||\beta||_1^2 + ||\mathbf{j}||_1^2 \leq c_3(||\nabla \times \mathbf{j} - \nabla\beta||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2).$$

Thus, there exists a constant, $c_4 = c_2 + c_3$, such that

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2$$

$$\leq c_4(||\nabla \times \mathbf{B} - \nabla\alpha||_0^2 + ||\nabla \cdot \mathbf{B}||_0^2 + ||\frac{1}{S_L}\nabla \times \mathbf{j} - \nabla\beta||_0^2 + ||\nabla \cdot \mathbf{j}||_0^2) = c_4 G_{max}^{(2)}.$$

Next, look at $\mathcal{L}^{(1)}_{max}$:

$$\mathcal{L}^{(1)}_{max}\mathbf{B}_{EM} = \begin{pmatrix} A_0 & 0 \\ D & A_1 \end{pmatrix} \begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix}.$$

This implies

$$G^{(1)}_{max} = || \begin{pmatrix} A_0 & 0 \\ D & A_1 \end{pmatrix} \begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix} ||^2_0.$$

Computing $||D \begin{pmatrix} \mathbf{B} \\ \alpha \end{pmatrix} ||^2_0$, we see that there exists a positive constant, c_5, such that

$$||D \begin{pmatrix} \mathbf{B} \\ \alpha \end{pmatrix} ||^2_0 = ||(\frac{1}{\Delta t} - \nabla \mathbf{u}^n)\mathbf{B} - \mathbf{u}^n \cdot \nabla \mathbf{B}||^2_0$$

$$\leq c_5(||\mathbf{B}||^2_0 + ||\nabla \mathbf{u}^n||^2_\infty ||\mathbf{B}||^2_0 + ||\mathbf{u}^n||^2_\infty ||\nabla \mathbf{B}||^2_0).$$

If there exists positive constants, c_6 and c_7, such that $||\nabla \mathbf{u}^n||_\infty \leq c_6$ and $||\mathbf{u}^n||_\infty \leq c_7$, then there exists a positive constant, c_8, that gives

$$||D\mathbf{B}||^2_0 \leq (1 + c_6)||\mathbf{B}||^2_0 + c_7||\nabla \mathbf{B}||^2_0 \leq c_8||\mathbf{B}||^2_1.$$

Then, Lemma 2 yields that there exists a positive constant, c_9, such that

$$||D\mathbf{B}||^2_0 \leq c_9(||\nabla \times \mathbf{B}||^2_0 + ||\nabla \cdot \mathbf{B}||^2_0)$$

$$\leq c_9||A_0\mathbf{B}||^2_0.$$

Thus, by Lemma 5, there exists a positive constant, c_{10}, such that

$$G^{(2)}_{max} = || \begin{pmatrix} A_0 & 0 \\ 0 & A_1 \end{pmatrix} \begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix} ||_0^2 \leq (1+c_9)|| \begin{pmatrix} A_0 & 0 \\ D & A_1 \end{pmatrix} \begin{pmatrix} \mathbf{B} \\ \alpha \\ \mathbf{j} \\ \beta \end{pmatrix} ||_0^2 = c_{10} G^{(1)}_{max}.$$

With this result, there exists a positive constant, c_{11}, which depends on c_4 and c_{10}, such that

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_4 G^{(2)}_{max} \leq c_{11} G^{(1)}_{max}. \tag{4.53}$$

Next, from equation (4.30), we see that

$$G^{(1)}_{max} \leq (G^{(0)}_{max} + ||\mathbf{j}||_0^2).$$

Hence, we now have

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_{11} G^{(1)}_{max} \leq c_{11}(G^{(0)}_{max} + ||\mathbf{j}||_0^2)$$

and then, by Lemma 4, there exists a positive constant, c_{max}, such that

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_{max} G^{(0)}_{max}. \tag{4.54}$$

Thus, we have shown coecivity for the Maxwell block.

Navier-Stokes

Next, we must do the same for linearized Navier-Stokes. Note that the constants, c_j, have been reset and are not the same as for the Maxwell proof. We want to show that there exists a positive constant, c_{ns}, such that

$$||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}^2||_1 + ||p||_1^2 \leq c_{ns} G_{ns}.$$

First, we show that there exists a positive constant, c_1, such that

$$||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}^2||_1 + ||p||_1^2 \leq c_1 \hat{G}_{ns},$$

where we define a reduced version of the Navier-Stokes functional to be

$$\hat{G}_{ns} = ||\nabla \times \mathbf{u} - \nabla\gamma||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2 + ||\nabla \cdot \boldsymbol{\omega}||_0^2 + || - \nabla p + \frac{1}{R_e}\nabla \times \boldsymbol{\omega}||_0^2.$$

This is related to the full Navier-Stokes functional from equation, (4.38), as follows:

$$\hat{G}_{ns} \leq c_2(G_{ns} + ||\mathbf{u}||_0^2 + ||\boldsymbol{\omega}||_0^2 + ||\mathbf{u}^n \times \boldsymbol{\omega}||_0^2 + ||\boldsymbol{\omega}^n \times \mathbf{u}||_0^2), \tag{4.55}$$

for some positive constant, c_2.

Again, by Lemma 3, we know there exists positive constants, c_3 and c_4, such that

$$||\gamma||_1^2 + ||\mathbf{u}||_1^2 \leq c_3(||\nabla \times \mathbf{u} - \nabla\gamma||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2)$$

and

$$||p||_1^2 + ||\boldsymbol{\omega}||_1^2 \leq c_4(||\nabla \times \boldsymbol{\omega} - \nabla p||_0^2 + ||\nabla \cdot \boldsymbol{\omega}||_0^2).$$

This implies that there exists a positive constant, $c_5 = c_3 + c_4$, such that

$$||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\boldsymbol{\omega}||_1^2 + ||p||_1^2$$

$$\leq c_5(||\nabla \times \mathbf{u} - \nabla\gamma||_0^2 + ||\nabla \cdot \mathbf{u}||_0^2 + ||\frac{1}{R_e}\nabla \times \boldsymbol{\omega} - \nabla p||_0^2 + ||\nabla \cdot \boldsymbol{\omega}||_0^2) = c_5 \hat{G}_{ns},$$

which says that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 \leq c_5 \hat{G}_{ns}. \tag{4.56}$$

Again, using (4.55), there exists a positive constant, c_6, such that

$$\hat{G}_{ns} \leq c_2(G_{ns} + ||\mathbf{u}||_0^2 + ||\mathbf{u}^n \times \boldsymbol{\omega}||_0^2 + ||\boldsymbol{\omega}^n \times \mathbf{u}||_0^2)$$

$$\leq c_6(G_{ns} + ||\mathbf{u}||_0^2(1 + ||\boldsymbol{\omega}^n||_\infty^2) + ||\mathbf{u}^n||_\infty^2||\boldsymbol{\omega}||_0^2).$$

Then, if $||\boldsymbol{\omega}^n||_\infty^2$ and $||\mathbf{u}^n||_\infty^2$ are bounded, there exists a positive constant, c_7, such that

$$\hat{G}_{ns} \leq c_7(G_{ns} + ||\mathbf{u}||_0^2 + ||\boldsymbol{\omega}||_0^2).$$

By a standard compactness argument, Lemma 4, we conclude that there exists a constant, $c_{ns} > 0$, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 \leq c_6\hat{G}_{ns} \leq c_{ns}G_{ns}. \tag{4.57}$$

Thus, we have coercivity of the linearized Navier-Stokes block.

Coupled System

Now consider the full system, where again the constants, c_j, have been reset and are not the same as the ones used above. From above, there exists positive constants, c_{ns} and c_{max}, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 \leq c_{ns}G_{ns}, \tag{4.58}$$

$$||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_{max}G_{max}^{(0)}. \tag{4.59}$$

Since

$$G_{ns} + G_{max}^{(0)} = G^{(2)},$$

we know there exists a positive constant, $c_1 = c_{ns} + c_{max}$, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_1G^{(2)}. \tag{4.60}$$

We use the same strategy here as in the Maxwell case. With the definition from equations (4.34)-(4.37), we have

$$\mathcal{L}^{(1)} = \begin{pmatrix} A_2 & 0 \\ T & A_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{FL} \\ \mathbf{B}_{EM} \end{pmatrix}.$$

Now, define

$$G^{(1)} = || \begin{pmatrix} A_2 & 0 \\ T & A_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{FL} \\ \mathbf{B}_{EM} \end{pmatrix} ||_0^2.$$

Next, one can see that

$$||T\mathbf{u}_{FL}||_0^2 = ||\nabla \mathbf{B}^n \cdot \mathbf{u} - \mathbf{B}^n \cdot \nabla \mathbf{u}||_0^2$$

and, thus, there exists a positive constant, c_2, such that

$$||T\mathbf{u}_{FL}||_0^2 \leq c_2(||\nabla \mathbf{B}^n||_\infty^2 ||\mathbf{u}||_0^2 + ||\mathbf{B}^n||_\infty^2 ||\nabla \mathbf{u}||_0^2).$$

If $||\nabla \mathbf{B}^n||_\infty \leq c_3$ and $||\mathbf{B}^n||_\infty \leq c_4$, for positive constants, c_3 and c_4, then there exists another constant, $c_5 > 0$, such that

$$||T\mathbf{u}_{FL}||_0^2 \leq c_5 ||\mathbf{u}||_1^2.$$

From the coercivity result for the Navier-Stokes block, we know that there exists a positive constant, c_6, such that

$$||T\mathbf{u}_{FL}||_0^2 \leq c_6 G_{ns} = c_6 ||A_2 \mathbf{u}_{FL}||_0^2.$$

Thus, Lemma 5 yields

$$G^{(2)} = || \begin{pmatrix} A_2 & 0 \\ 0 & A_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{FL} \\ \mathbf{B}_{EM} \end{pmatrix} ||_0^2$$

$$\leq (1 + c_6) || \begin{pmatrix} A_2 & 0 \\ T & A_3 \end{pmatrix} \begin{pmatrix} \mathbf{u}_{FL} \\ \mathbf{B}_{EM} \end{pmatrix} ||_0^2 = (1 + c_6) G^{(1)}.$$

Using the bound in equation, (4.60), we see that there exists another positive constant, c_7, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_1 G^{(2)} \leq c_7 G^{(1)}. \quad (4.61)$$

Finally, one can see from equation (4.29) that

$$G^{(1)} \leq (G + ||\mathbf{B}^n \times \mathbf{j}||_0^2 + ||\mathbf{j}^n \times \mathbf{B}||_0^2$$

and, thus, there exists a positive constant, c_8, such that

$$G^{(1)} \leq c_8(G + ||\mathbf{B}^n||_\infty^2 ||\mathbf{j}||_0^2 + ||\mathbf{j}^n||_\infty^2 ||\mathbf{B}||_0^2).$$

We assume that $||\mathbf{B}^n||_\infty^2$ and $||\mathbf{j}^n||_\infty^2$ are bounded and, hence, there exists a positive constant, c_9, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_9(G + ||\mathbf{B}||_0^2 + ||\mathbf{j}||_0^2).$$

Again with the standard compactness argument, Lemma 4, we have that there exists a positive constant, c_{mhd}, such that

$$||\boldsymbol{\omega}||_1^2 + ||p||_1^2 + ||\mathbf{u}||_1^2 + ||\gamma||_1^2 + ||\mathbf{B}||_1^2 + ||\alpha||_1^2 + ||\mathbf{j}||_1^2 + ||\beta||_1^2 \leq c_{mhd} G. \quad (4.62)$$

Finally, we have coercivity of the full incompressible, resistive MHD system.

□

4.2.5 Equivalence

With the above upper and lower bounds, equations (4.62) and (4.50), and with $\mathcal{U}$ as defined in Theorem 6, we have

$$c_1 ||\mathcal{U}||_1^2 \leq G \leq c_2 ||\mathcal{U}||_1^2. \quad (4.63)$$

Here, we have replaced c_{mhd} with c_1. We showed that the current-pressure-vorticity formulation defined above is an H^1 elliptic system. This assumes we have vorticity and current boundary conditions on convex polyhedral domains with connected boundaries. This is the case for all the test problems shown in

this thesis. In addition, with the scaling described above, we chose a first-order system in which the linear systems arising from a discrete form of the FOSLS formulation can be solved via Algebraic Multigrid with reasonable convergence factors. The scaling may change the size of the constants, c_1 and c_2, but equivalence is still obtained. Next, we develop an efficient algorithm for solving this first-order system.

Chapter 5

Solution Methods

Now that a FOSLS formulation has been established, we describe the strategy for solving the algebraic systems that result from its discretization. The main component of this algorithm is a nested iteration approach. Most of the work is done on coarse grids where computation is much cheaper. The approximation to the solution on these coarse grids is interpolated up to successively finer grids, where linearization and multigrid cycles are applied, and the process is then continued to finer grids until a desired error tolerance or resolution is reached.

First, an initial guess is used on a very coarse grid. Using this initial guess, the system is linearized as described above. The least-squares functional is then formed and the minimization results in an algebraic system, which is denoted by $Ax = b$. Several AMG V-cycles are used to solve the algebraic system. If necessary, several more linearization steps are done and the process is repeated. After the Newton steps have converged and the algebraic system is solved "well-enough," the solution is interpolated to a finer grid where it is again linearized, FOSLized, and solved with AMG [43].

In this chapter, we describe an approach that attempts to minimize the number of linearization steps and AMG V-cycles required to obtain an approximation that is of the same accuracy as the discretization error on the current grid. A strategy has been developed for this, parts of which are described in [49]. We go

into more detail here. In addition, we discuss how refinement is done between grids using both uniform refinement and an adaptive mesh refinement (AMR) scheme. Also, since the MHD equations are time dependent, we discuss the implications of using standard time-stepping schemes with this nested iteration algorithm. We compare Backward Euler, Crank-Nicholson, and a second-order backwards differencing formula (BDF-2) and determine that BDF-2 yields superior accuracy and stability.

5.1 Nested Iteration Strategies

The goal for this section is to find out how many Newton steps and linear iterations are needed at each refinement level to solve a problem with the most accuracy-per-computational-cost. Results below show that only a few Newton steps and V-cycles are needed to get a good approximation to the solution on the finest grid. Under natural hypothesis, which are in force here, this was proved in [19, 20]. In addition, we discuss the heuristic that it is not necessary to solve the discrete problem on these coarse grids below the level of discretization error. In fact, at some point, doing more work on the coarser grid is not as efficient as just moving to a finer grid. It is at this point that the FOSLS methodology proves to be important. Using the a posteriori error estimate of the functional, we determine how well both the linear and nonlinear systems are being solved. The linearized and nonlinear functional merely need to be computed. The squared functionals are given in equations (3.12) and (3.8), respectively, and the functional norms are denoted by F for the linearized functional and $\mathcal{F}$ for the nonlinear functional. In other words, $\mathcal{F}^2 = \mathcal{G}$ and $F^2 = G$. We refer to F and $\mathcal{F}$ as the linearized and nonlinear functionals.

With the linearized functional, along with the AMG convergence factors, one may compute an estimate of how much the error of the linear system is

being reduced and at what cost. Likewise, by checking the nonlinear functional and how many Newton steps are performed, one can estimate the accuracy-per-computational cost for solving the nonlinear system. These estimates are then used to establish stopping parameters for the linear iterations as well as for the Newton iterations. For instance, for the linear systems, we can decide if the best strategy is to either perform another V-cycle on the current grid, or to relinearize the system. Work here is an extension of [49], where the iterations were stopped when the functional value of the current iterate came within a given tolerance of the functional minimum.

The Newton iteration is controlled by comparing the linearized and nonlinear functionals after each Newton step. If the linearized functional is within a given tolerance of the nonlinear functional, no more linearizations are performed. Instead, the computation is transferred to a finer grid. The structure of this finer grid can be determined by an adaptive mesh refinement strategy that is described in subsection 5.2.1. In the next two subsections, we describe the stopping criteria in more detail.

5.1.1 The Linear Problem

First, assume we are solving a linear problem, $Lu = f$, and that there are two grids: grid h and grid $2h$. Assume for simplicity that uniform refinement is used to transfer between grids and, therefore, moving to a finer grid increases the number of elements on the finite element mesh by 2^d, where d is the dimension of the problem. The grid spacing on grid h is the half that of grid $2h$. Let the true continuous solution to the problem be u_* and let u_*^{2h} and u_*^h be the discrete solutions on their respective grids. Since the FOSLS method is used, the functional is an estimate of the error. We denote the discretization error, measured in the functional norm, on grids h and $2h$ by F_*^h and F_*^{2h}, respectively. In the process

of solving on grid $2h$, several iterates, u_i^{2h}, $i = 0, 1, ...$ are obtained. At iteration, i, with iterate u_i^{2h}, the error is measured and denoted by F_i^{2h}. It is convenient to write this as a multiple of the discretization error on grid $2h$; that is,

$$(F_i^{2h})^2 = (1 + (\epsilon_i^{2h})^2)(F_*^{2h})^2. \tag{5.1}$$

The value $\epsilon_i^{2h} F_*^{2h}$ is referred to as the algebraic error on grid $2h$. We also assume that the algebraic solver has a steady convergence factor that is independent of the size of the grid or the current iterate. This is not precisely true, but we can assume that the convergence factor is bounded above by a constant, $\rho < 1$, independent of the mesh. Moreover, we can update the value we use for the convergence factor as we move through the grids so that the estimates become more accurate on finer grids. Thus, ρ is assumed to be known and $\epsilon_{i+1}^{2h} = \rho\epsilon_i^{2h}$. In practice, this is obtained from a sequence of residuals of the matrix equations.

Figure 5.1 is intended to demonstrate the strategy for determining when to stop iterating on grid $2h$ and move to grid h. Suppose we are iterating on grid $2h$ and that the current iterate is u_s^{2h}, with algebraic error $\epsilon_s^{2h} F_*^{2h}$. Grid $2h$ is indicated in figure 5.1 by the subspace restricted to the x-axis, while the plane represents grid h, and the true solution, u_*, is shown hovering above u_*^h. We assume that the discrete solution on grid h, u_*^h, is closer to the true solution, u_*, than the coarse grid approximation, u_*^{2h}. In addition, we know that iterating on grid $2h$ restricts us to only getting closer to u_*^{2h}. Therefore, we would have much better accuracy if we were iterating on the finer grid, h. However, iterating on grid h is more costly. Thus, we would like to know how much accuracy we gain by doing one more iteration on grid $2h$ and compare this to the amount of accuracy we gain by interpolating to grid h and moving directly toward u_*^h. That is, from the position denoted by u_s^{2h}, is it better to move to position u_{s+1}^{2h} or to the position denoted by u_1^h. Of course the latter move is more expensive. Therefore,

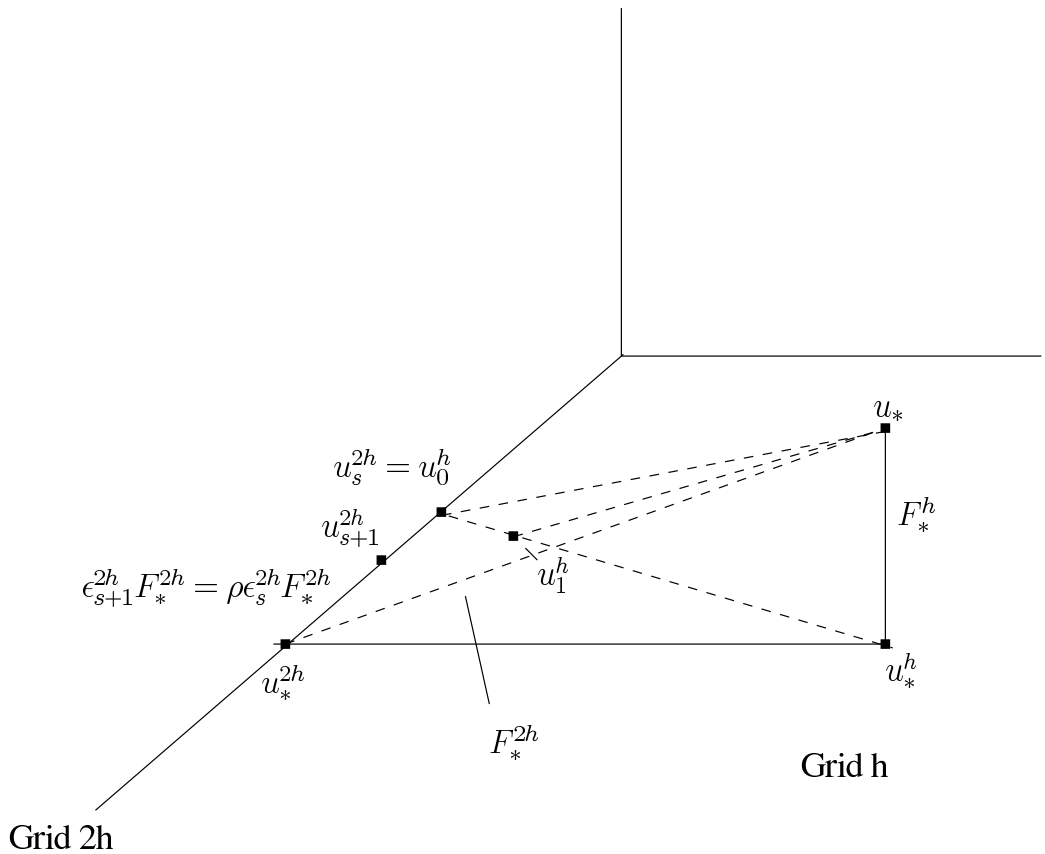

Figure 5.1: Linear Iterations along grid $2h$ and grid h.

we need to establish a measure of the accuracy-per computational cost of both these moves. By assumption, computation on grid h costs 2^d times the cost of an iteration on grid $2h$ and, by examining the triangles in figure 5.1, we can determine the reduction of the two errors from the two types of iteration. In the next few paragraphs, we quantify a measure that helps us determine an optimal strategy.

If we stay on grid $2h$ and iterate once more, the new total error for iterate u_i^{2h} is a sum of the discretization error plus the algebraic error, which we approximate as

$$F_{s+1}^{2h} = ((F_*^{2h})^2 + (\rho \epsilon_s^{2h} F_*^{2h})^2)^{\frac{1}{2}}. \tag{5.2}$$

If, instead, we move to grid h and iterate to obtain u_1^h the error can be approxi-

mated as

$$F_1^h = ((F_*^h)^2 + (\rho\epsilon_0^h F_*^h)^2)^{\frac{1}{2}}. \tag{5.3}$$

However, in order to compare these to errors it is important to take into account the fact that an iteration on grid h takes 2^d times more work than on grid $2h$. Instead of looking at the regular convergence factor, ρ, we look at the effective convergence factor, $\rho^{1/\text{Work}}$, and, thus, the effective error on grid h is approximated as

$$F_1^h = ((F_*^h)^2 + (\rho^{\frac{1}{2^d}}\epsilon_0^h F_*^h)^2)^{\frac{1}{2}}. \tag{5.4}$$

The two errors are equivalent when

$$(F_*^{2h})^2 + (\rho\epsilon_s^{2h} F_*^{2h})^2 = (F_*^h)^2 + \rho^{2^{1-d}}(\epsilon_0^h F_*^h)^2. \tag{5.5}$$

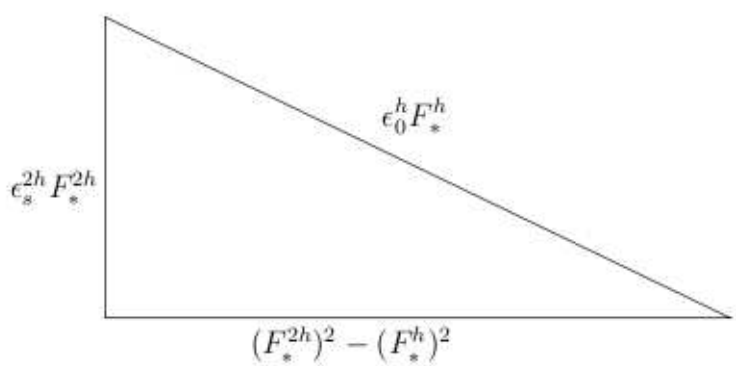

Figure 5.2: Relationship of algebraic errors on different grids.

Extracting the triangle in figure 5.2 from figure 5.1, we can relate the algebraic error on grid $2h$, $\epsilon_s^{2h} F_*^{2h}$, to the algebraic error on grid h, $\epsilon_0^h F_*^h$. Then, after some algebra, we obtain

$$(\hat{\epsilon}^{2h})^2 = \frac{(\rho^{2^{1-d}} - 1)(1 - (\frac{F_*^h}{F_*^{2h}})^2)}{\rho^2 - \rho^{2^{1-d}}}. \tag{5.6}$$

In other words, we have a measure that determines if it is more efficient, in terms of accuracy-per-computational cost, to stay on grid h or to move to grid $2h$. When

the algebraic error on grid $2h$ is less than $\hat{\epsilon}^{2h} F_*^{2h}$, it is more efficient to move to the finer grid.

Of course, this assumes that we know the discretization errors, F_*^{2h} and F_*^h. We only know the total error, or linearized FOSLS functional value, of the current iterate and not the discretization and algebraic errors separately, as in equation (5.1). One way to estimate the discretization errors is to assume that the functional values decrease at a linear rate. Since the functional squared is a quadratic, we can extrapolate an estimate of the discretization error, or the minimum of the quadratic functional. We assume

$$(F_s^{2h})^2 = (1 + (\epsilon_s^{2h})^2)(F_*^{2h})^2,$$

and that

$$(F_{s+1}^{2h})^2 = (1 + (\epsilon_{s+1}^{2h})^2)(F_*^{2h})^2$$

$$= (F_{s+1}^{2h})^2 = (1 + \rho^2(\epsilon_s^{2h})^2)(F_*^{2h})^2.$$

Thus, three consecutive steps yields an estimate of both ϵ_s^{2h} and F_*^{2h}. This requires doing a few extra iterations to get a good estimate of the minimum, but since this work is done on coarse grids, we argue that this does not increase the total work significantly.

For each linear solve, we have a stopping criteria. In order to accomplish this, though, we need to know the convergence factor, ρ, and the reduction factor due to the refinement, $\dfrac{F_*^h}{F_*^{2h}}$. However, we can use the information from a previous grid. For instance, when uniform refinement is used, we approximate F_*^h from F_*^{2h} using the assumption that

$$F_*^h = \frac{1}{2^p} F_*^{2h},$$

where p is the order of the finite element space used. In addition, as we move through the grids, we can update all the parameters, thus, updating our stopping

criteria. This yields a dynamic method of determining how well the linear iterations are being solved, the accuracy-per-computational cost of a V-cycle, and, finally, answers the question 'To Iterate or Not to Iterate.'

5.1.2 The Nonlinear Problem

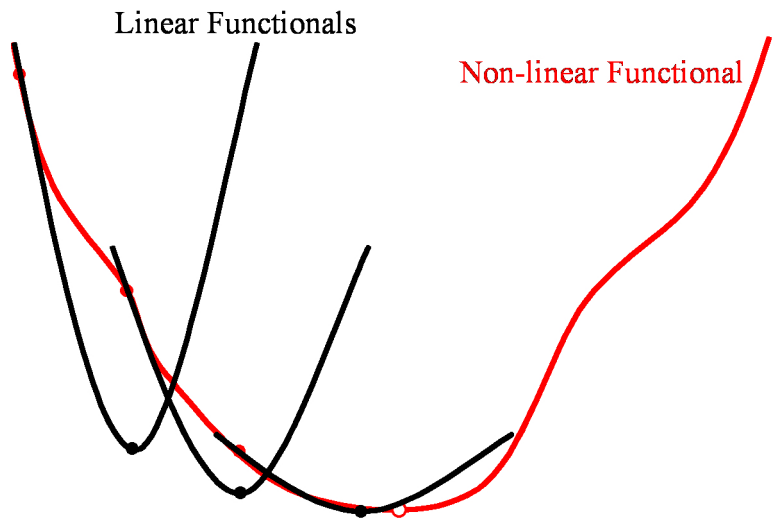

Figure 5.3: The nonlinear functional in red compared to several iterates of the linearized functional in black.

Now that we have a way of determining when to stop the linear iterations, we still have to keep in mind that the real problem is nonlinear. What we truly want to minimize is the nonlinear functional, $\mathcal{F}^h$, on grid h. The linear solves, though, are used to minimize the linearized functional that we assume gets closer to the nonlinear functional with every Newton step . If we consider, for instance, a 1D interpretation of the nonlinear functional in figure 5.3, we can see that, after a few linearizations by a quadratic functional, the minimum of the linearized functional approaches the minimum of the nonlinear functional. Therefore, after each linear solve, we check

$$\frac{|F_j^h - \mathcal{F}_j^h|}{\mathcal{F}_j^h} \leq \eta, \tag{5.7}$$

where F_j^h is the current value of the linearized functional after j Newton steps. This tells us to stop the Newton iteration when the linearized functional is within a certain relative distance from the nonlinear functional. After each Newton step, we have an approximation that is "close" to the predicted minimum of the linearized functional based on the stopping criteria established in subsection 5.1.1. If the linearized functional is "close" to the nonlinear functional, this is a good indicator that the approximation is also "close" to the minimum of the nonlinear functional. In practice, we compare the relative distance between the square functionals as this is easily computed in the code.

For the MHD systems, the highest order, nonzero Fréchet derivative of the nonlinear operator, $\mathcal{L}$, is the second-order derivative, $\mathcal{L}''$. Let $\delta = u - u_0$ be the direction toward the minimum from the initial approximation, u_0, and, thus, the nonlinear operator is rewritten as

$$\mathcal{L}(u_0 + \delta) = \mathcal{L}(u_0) + \mathcal{L}'(u_0)[\delta] + \frac{1}{2}\mathcal{L}''(u_0)[\delta, \delta].$$

Then, the square of the linearized functional, $G(u_0+\delta)$, is the same as in equation (3.8),

$$G(u_0 + \delta) = ||\mathcal{L}(u_0) - f + \mathcal{L}'(u_0)[\delta]||_0^2,$$

and the square of the nonlinear functional, equation (3.12), is simplified to

$$\mathcal{G}(u_0 + \delta) = ||\mathcal{L}(u_0) - f + \mathcal{L}'(u_0)[\delta] + \frac{1}{2}\mathcal{L}''(u_0)[\delta, \delta]||_0^2.$$

The first Fréchet derivatives of the squared linearized and nonlinear functionals become

$$G'(u_0 + \delta)[v] = 2 < \mathcal{L}(u_0) - f + \mathcal{L}'(u_0)[\delta], \mathcal{L}'(u_0)[v] >$$

and

$$\mathcal{G}'(u_0+\delta)[v] = 2 < \mathcal{L}(u_0) - f + \mathcal{L}'(u_0)[\delta] + \frac{1}{2}\mathcal{L}''(u_0)[\delta, \delta], \mathcal{L}'(u_0)[v] + \mathcal{L}''(u_0)[\delta, v] >,$$

respectively. Since the approximation is near the minimum of the linearized functional, $G'(u_0+\delta)[v]$ is very small and

$$\mathcal{G}'(u_0+\delta)[v] = 2 < \mathcal{L}(u_0) - f + \mathcal{L}'(u_0)[\delta] + \frac{1}{2}\mathcal{L}''(u_0)[\delta,\delta], \mathcal{L}''(u_0)[\delta,v] >$$

$$+ < \frac{1}{2}\mathcal{L}''(u_0)[\delta,\delta], \mathcal{L}'(u_0)[v] > .$$

If $||\mathcal{L}''(u_0)[\delta,\delta]||_0$ is small, then it is easily shown that $\mathcal{G}'$ is small and the minimum of the nonlinear functional is obtained.

While the measure in equation (5.7) does not get this precisely, numerical results show that it is a good indicator of when to stop iterating. It can be shown that

$$|F - \mathcal{F}| \leq ||\frac{1}{2}\mathcal{L}''(u_0)[\delta,\delta]||_0. \tag{5.8}$$

Equation (5.8) states that if the stopping estimate, equation (5.7), is large, then $||\frac{1}{2}\mathcal{L}''(u_0)[\delta,\delta]||_0$ is also large. It should be noted that when the difference between the squared functionals is large, equation (5.7) must be large too and the result still holds. This is, then, a good indication that another linearization is needed. In practice, on coarse grids, we do enough Newton iterations for which the linearized and nonlinear functional are almost equal and then do one more step. Then, for finer grids, we expect the initial approximation to the solution to be close enough to the discrete solution that the Newton step, δ, is small and, thus, the minimum is obtained quickly.

5.2 Solution Algorithm

With the pieces above, we now devise an algorithm to solve a system of nonlinear equations, $\mathcal{L}(u) = f$. Starting on a coarse grid, given an initial guess, the system is linearized and the linearized FOSLS functional is minimized. At this point, several AMG V-cycles are performed and it is eventually decided that

performing another iteration does not gain sufficient accuracy-per-computational cost. Here, we relinearize the system and search for the minimum of the new linearized FOSLS functional in the same manner. After each set of linear solves, the relative difference between the computed linearized functional and the nonlinear functional is checked. If they are close and near the minimum of the linearized functional, then it is concluded that we are close enough to the minimum of the nonlinear functional and, hence, we have a good approximation to the solution on the given grid. Next, the approximation is interpolated to a finer grid and the problem is solved on that grid. This process is repeated until an acceptable error has been reached, or until we have run out of computational resources, such as memory. If, as in the case of the MHD equations, it is a time-dependent problem, the whole process is performed at each time step. This algorithm is summarized in the flow chart, figure 5.4. The next few sections discuss some attributes of the algorithm, namely, refinement and timestepping.

5.2.1 Adaptive Refinement

In the nested iteration algorithm, we decide when to stay on a current mesh and iterate further or interpolate to a finer grid. So far, we have assumed that the interpolated grid is uniformly refined. In other words, we have assumed that there are 2^d more points on the next grid than the one before, where, again, d is the dimension of the problem. This does not have to be the case. We can refine the grids locally. On a given mesh, after enough Newton steps and linear iterations have been performed, we check the nonlinear functional in each element. This tells us in which region of the domain the functional and, hence, the error is large compared to the rest of the domain. Then, the best use of new degrees of freedom is to concentrate them where the error is large. Since the goal of the algorithm is to increase the accuracy-per-computational cost, we do not want to oversolve

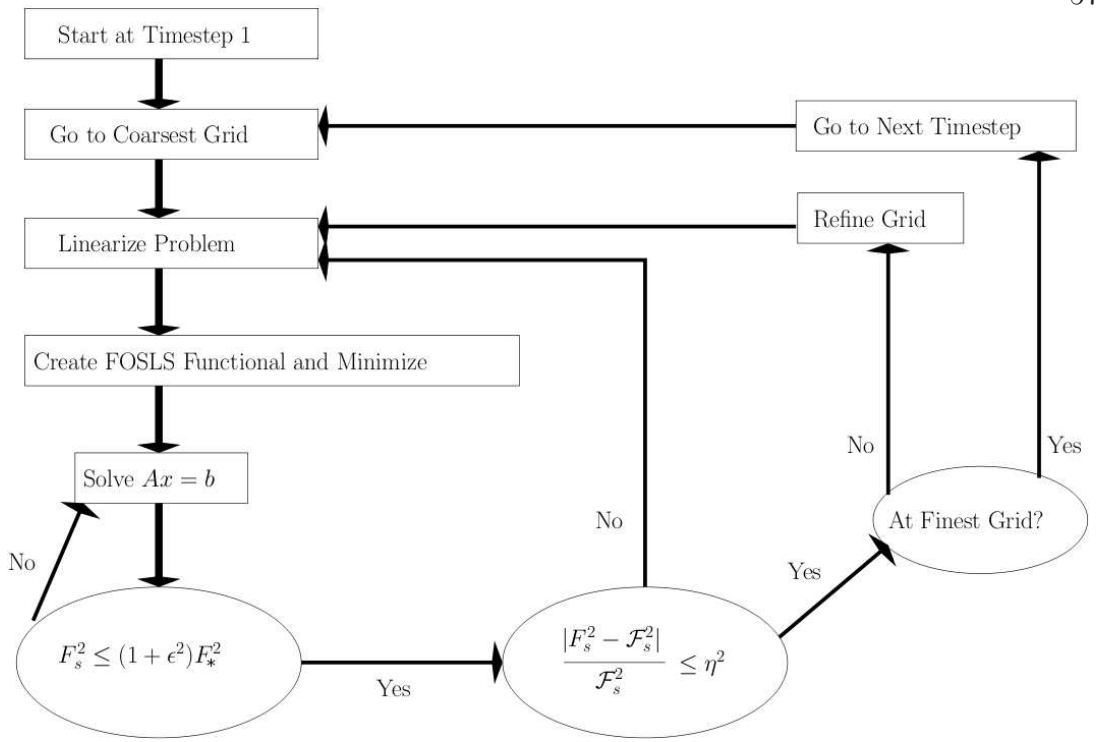

Figure 5.4: Flow chart of nested iteration algorithm.

in areas where the error is already small. The adaptive scheme that we describe here is an efficiency based refinement method, called ACE, that was developed in [22, 37]. This scheme estimates both the reduction in the functional and the computational cost that would result from any given refinement pattern. These estimates are used to establish a refinement pattern that attempts to optimize the Accuracy-per-Computational cost (Efficiency), which gives rise to the acronym ACE.

The square of the functional value on each element, ϵ_i, is computed and is ordered such that the local functional value is decreasing:

$$\epsilon_1 \geq \epsilon_2 \geq \ldots \geq \epsilon_{N_l}, \tag{5.9}$$

where N_l is the total number of elements on level l. Next, we predict the functional reduction and the estimated computational work that is required if we were

to refine a given percentage of the elements with the most error. Denote the percentage by $r \in (0, 1]$ and the number of elements to be refined by $r * N_l$. Define $f(r)$ to be the functional distribution in $r * N_l$ elements, with largest error, that is,

$$f(r) = \frac{\sum_{i<r*N_l} \epsilon_i}{\sum_{i=1}^{N_l} \epsilon_i}. \tag{5.10}$$

Then, we predict that the functional is reduced by the factor

$$\gamma(r) = 1 - f(r) + (\frac{1}{2})^{2p} f(r). \tag{5.11}$$

Here, p is the order of the finite element basis. In other words, the predicted error reduction only comes from the regions where the elements have been refined. The predicted error in the remaining elements is left unchanged. However, the predicted work is assumed to be a function of the total number of elements on the refined grid. Since each refined element yields 2^d children, we predict the number of elements on the refined grid to be

$$N_{l+1} = N_l(1 - r + 2^d r). \tag{5.12}$$

The work required to solve the linear system on the refined grid is assumed to be a function of the number of elements, N_{l+1}, the error reduction factor, $\gamma(r)$, and the AMG convergence factor, ρ. Assume the cost of one AMG V-cycle is C_1 and the overall setup cost for FOSLS and AMG is C_0. The number of V-cycles needed to reduce the error by $\gamma(r)$ using AMG is

$$n = \frac{log\gamma}{log\rho}. \tag{5.13}$$

Thus, the cost of performing AMG on one element is

$$C = (C_0 + C_1 \frac{log\gamma}{log\rho}). \tag{5.14}$$

Then, the overall predicted work of solving on a refined grid is

$$W(r) = C(1 - r + 2^d r). \tag{5.15}$$

With these relations in mind, we choose the r that minimizes the predicted effective functional reduction,

$$\gamma(r)^{eff} = \gamma(r)^{1/W(r)}. \tag{5.16}$$

Therefore, we refine only $r * N_l$ elements on each level that gives us the the best error reduction for the added cost. In addition, one could allow for multiple refinements of each element. For example, the ACE scheme could call for $r_1 N_l$ elements to be refined once and $r_2 N_l$ elements to be refined twice. This changes the predicted error reduction, $\gamma(r_1, r_2)$, and the predicted work estimate, $W(r_1, r_2)$. Now, the optimal r_1 and r_2 are found to minimize the effective error reduction, equation (5.16). This allows for more aggressive refinement.

The ACE scheme fits in nicely with the nested iteration approach. Not only do we try to get the best accuracy-per-computational cost for each linearization and each AMG cycle, but we also take this into account when interpolating to finer grids. A more detailed explanation of the ACE algorithm can be found in [22, 37]. Numerical results in chapter 6 show that using adaptive refinement yields the same accuracy in the MHD test problems while using fewer degrees of freedom.

5.3 Timestepping

Another important aspect of the above algorithm is timestepping. For all tests in Chapter 6, backward differencing formulas (BDF) are used, specifically Backward Euler (BDF-1) and BDF-2. These are fully implicit time-stepping schemes. For a linear, scalar PDE these schemes are unconditionally stable and,

therefore, there is no need to limit the time step to satisfy the Courant-Friedrichs-Levy (CFL) condition. In other words, the size of the time step can be determined by the physics and accuracy, not by any special concern for stability. However, the FOSLS formulation of the incompressible, resistive MHD equations are a set of nonlinear differential algebraic equations (DAEs). This means that time-evolving equations couple with constraint equations. In general, this looks like the following:

$$\frac{\partial \mathbf{u}}{\partial t} + L(\mathbf{u}, \mathbf{v}) = f, \tag{5.17}$$

$$G(\mathbf{u}, \mathbf{v}) = 0, \tag{5.18}$$

where L and G are nonlinear operators. For the MHD equations, (4.21)-(4.28), the fluid field, $\mathbf{u}$, and the magnetic field, $\mathbf{B}$, are time evolved and coupled with auxiliary equations. The question now is whether one can still guarantee stability and accuracy of the implicit time-stepping schemes if the auxiliary equations are not solved exactly. In the FOSLS setting, each equation is solved with an equal amount of accuracy. Each equation is weighted appropriately in the functional, so that the minimization of the functional satisfies all the equations equally. Therefore, the constraint equations are only solved to within this prescribed accuracy and this might deteriorate the convergence or stability of the time-stepping schemes. Thus, we need to determine the level of accuracy to which the auxiliary equations must be solved in order to maintain the good properties of the time-stepping schemes.

In [32], one can find a detailed analysis of DAE systems. Our MHD system is a nonlinear semi-explicit index-two DAE. It is semi-explicit in the sense that none of the nonlinearities involve the time derivatives. In other words, the time

derivatives, $\frac{\partial \mathbf{u}}{\partial t}$ and $\frac{\partial \mathbf{B}}{\partial t}$, can be explicitly solved for in terms of the other variables. The index of a DAE is the minimal number of constraint equation differentiations needed to get explicit time-evolving equations for all the dependent variables. To show that the MHD system has index two, we rewrite equations (4.21)-(4.28), introducing operators L_1, L_2, L_3, and L_4 for simplicity:

$$\begin{aligned}
\frac{\partial \mathbf{u}}{\partial t} &= L_1(\mathbf{u}, \boldsymbol{\omega}, p, \mathbf{j}, \mathbf{B}), && (5.19)\\
\frac{\partial \mathbf{B}}{\partial t} &= L_2(\mathbf{u}, \mathbf{B}, \mathbf{j}), && (5.20)\\
\boldsymbol{\omega} &= L_3(\mathbf{u}), && (5.21)\\
\mathbf{j} &= L_4(\mathbf{B}), && (5.22)\\
\nabla \cdot \mathbf{u} &= 0, && (5.23)\\
\nabla \cdot \boldsymbol{\omega} &= 0, && (5.24)\\
\nabla \cdot \mathbf{B} &= 0, && (5.25)\\
\nabla \cdot \mathbf{j} &= 0. && (5.26)
\end{aligned}$$

In order to get time-dependent equations for $\boldsymbol{\omega}$ and $\mathbf{j}$, we take the time derivative of equations (5.21) and (5.22) and substitute in the time derivatives of $\mathbf{u}$ and $\mathbf{B}$ from equations (5.19) and (5.20), respectively. For the pressure, p, however, one must take two derivatives of the incompressibility constraint in $\mathbf{u}$, equation (5.23). This gives

$$\nabla \cdot \frac{\partial^2 \mathbf{u}}{\partial t^2} = 0.$$

Taking the divergence and another time derivative of equation (5.19) then gives

$$0 = \nabla \cdot \frac{\partial}{\partial t}(\mathbf{u} \times \boldsymbol{\omega} + \mathbf{j} \times \mathbf{B}) - \sqrt{R_e}\nabla^2 \frac{\partial p}{\partial t}.$$

Since the div grad operator, ∇^2, is invertible, we can directly solve for $\frac{\partial p}{\partial t}$ using the other equations and, thus, we have time equations for all the dependent variables, which implies the system has index two. It can then be shown, as is done in [32], that backward differencing formulas (BDF) of order $k < 7$ are convergent and accurate. We state the theorem here.

Theorem 7. *Suppose the nonlinear semi-explicit index-two system is to be solved numerically by the k-step BDF method ($k < 7$). The errors in the initial values are $||e_0|| = O(h^k)$, and the errors in terminating the Newton iteration satisfy $O(h^{k+1})$. Then, the k-step BDF method is convergent and globally accurate to $O(h^k)$, after $k + 1$ steps.*

The proof can be found in [32].

With the above theorem, we now feel safe using a BDF scheme for our timestepping. However, we need to show that the spatial part of the PDE still produces a stable propagation matrix. In other words, if any error is introduced into any part of the approximation, we do not want this to grow with each time step. Therefore, we examine how the error in each dependent variable is propagated at each time step. To see what this means for the continuous problem via FOSLS, for simplicity, consider 2D incompressible, time-dependent Stokes equations:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla p - \nabla^2 \mathbf{u} = 0, \tag{5.27}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{5.28}$$

Again, $\mathbf{u}$ is the fluid velocity and p is the fluid pressure. To write this as a first-order system, we introduce vorticity, ω, as is done for the MHD formulation of

chapter 4. In a 2D setting, the equations become

$$\frac{\partial \mathbf{u}}{\partial t} - \nabla p + \nabla^{\perp}\omega = 0, \tag{5.29}$$

$$\nabla \times \mathbf{u} - \omega = 0, \tag{5.30}$$

$$\nabla \cdot \mathbf{u} = 0. \tag{5.31}$$

In matrix form, we can write the operator as

$$\mathcal{L}\mathcal{U} = \begin{pmatrix} D & -E \\ \partial t & D^* \end{pmatrix} \begin{pmatrix} \mathbf{u} \\ \boldsymbol{\omega} \end{pmatrix}, \tag{5.32}$$

where $\boldsymbol{\omega} = \begin{pmatrix} \omega \\ p \end{pmatrix}$ and

$$D = \begin{pmatrix} \nabla\times \\ \nabla\cdot \end{pmatrix} = \begin{pmatrix} -\partial y & \partial x \\ \partial x & \partial y \end{pmatrix},$$

$$E = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}.$$

Now, we analyze the backward differencing formulas (BDFs) and Crank-Nicholson time-stepping schemes [2] to see if they maintain stability in this context. Let L denote the operator in equation (5.17) and $\mathcal{U}$ some vector of dependent variables. For the Stokes system, $\mathcal{U} = (\mathbf{u}, \omega, p)^T$. First, consider the BDF-k schemes, where k is the order of accuracy:

$$\frac{\partial \mathcal{U}}{\partial t} = L(\mathcal{U}) \Rightarrow \mathcal{U}_{n+1} - \sum_{j=0}^{k-1} \alpha_j \mathcal{U}_{n-j} = \Delta t \beta L(\mathcal{U}_{n+1}). \tag{5.33}$$

For BDF-1, we have $\alpha_0 = \beta = 1$, which is Backward-Euler. Substituting equation (5.33) into the system (5.32), we obtain the following spatially continuous system,

$$\mathcal{L}\mathcal{U}_{n+1} = \begin{pmatrix} \beta D & -\beta E \\ \frac{1}{\Delta t} I & \beta D^* \end{pmatrix} \mathcal{U}_{n+1} = \sum_{j=0}^{k-1} \alpha_j \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix} \mathcal{U}_{n-j}. \tag{5.34}$$

Letting $\sum_{j=0}^{k-1} \alpha_j \mathcal{U}_{n-j} = \mathcal{U}_m$, then we can simplify system (5.34) to

$$\mathcal{L}\mathcal{U} = \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix} \mathcal{U}_m. \tag{5.35}$$

Here, $\mathcal{U}$ is the solution we are looking for, where the subscript $n+1$ has been dropped and $\mathcal{U}_m$ is a linear combination of known solutions on previous time steps.

For the FOSLS setting, we are, equivalently, solving the system

$$\mathcal{L}^* \mathcal{L}\mathcal{U} = \mathcal{L}^* \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix} \mathcal{U}_m, \tag{5.36}$$

where $\mathcal{L}^*$ is the L^2 adjoint of the operator, $\mathcal{L}$. Therefore, the solution at each time step, $\mathcal{U}$, is propagated in the following way:

$$\mathcal{U} = (\mathcal{L}^* \mathcal{L})^{-1} \mathcal{L}^* \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix} \mathcal{U}_m. \tag{5.37}$$

If the norm of $(\mathcal{L}^* \mathcal{L})^{-1} \mathcal{L}^* \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix}$ is bounded by one for all parameters, we have an unconditionally stable time-stepping scheme. In the continuous system, $(\mathcal{L}^* \mathcal{L})^{-1} \mathcal{L}^*$ is merely $\mathcal{L}^{-1}$, which makes things simpler to analyze. We consider both these propagation matrices.

Computing the inverse, $\mathcal{L}^{-1}$, first and multiplying it against the right-hand side, we obtain the following propagation matrix for the Stokes equations using a BDF scheme:

$$\mathcal{U} = \begin{pmatrix} I - \beta^2 \Delta t D^*(\beta \Delta t D D^* + E)^{-1} D & 0 \\ \beta(\beta \Delta t D D^* + E)^{-1} D & 0 \end{pmatrix} \mathcal{U}_m. \tag{5.38}$$

The spectral radius of this matrix is the spectral radius of the top 2 by 2 block since it is lower triangular. Taking a closer look at this block,

$$I - \beta^2 \Delta t D^*(\beta \Delta t D D^* + E)^{-1} D, \tag{5.38a}$$

and noting that the DD^* operator is the vector Laplacian,

$$DD^* = D^*D = -\Delta = \begin{pmatrix} -\nabla^2 & 0 \\ 0 & -\nabla^2 \end{pmatrix},$$

we now show that the 2 by 2 block, (5.38a), has a spectral radius less than one and that it is self-adjoint. This means that its norm is also less than one. With β and Δt always positive, $\beta^2 \Delta t D^*(\beta \Delta t D D^* + E)^{-1} D$ has real positive eigenvalues. In addition, since $||E|| \leq ||I||$ we can see that $\beta^2 \Delta t D^*(\beta \Delta t D D^* + E)^{-1} D$ also only has eigenvalues between 0 and 1. Therefore, the norm of this block is

$$||I - \beta^2 \Delta t D^*(\beta \Delta t D D^* + E)^{-1} D|| \leq 1 - \beta \frac{\beta \Delta t ||\Delta||}{||\beta \Delta t \Delta + I||} < 1. \tag{5.39}$$

Here, E has been replaced with I and the statement is true for $0 < \beta \leq 1$, which is valid for BDF schemes of order less than 7. Thus, the norm of the propagation matrix is bounded by one for BDF-k schemes of order less than 7 and, for the Stokes system, we have a stable time-stepping scheme.

Next, we consider if this still holds in the FOSLS context with a discrete subspace, since, if a spatial discretization is used, $(\mathcal{L}^*\mathcal{L})^{-1}\mathcal{L}^*$ may no longer equal $\mathcal{L}^{-1}$. For the Stoke's system, we get the following propagation matrix for FOSLS,

$$(\mathcal{L}^*\mathcal{L})^{-1}\mathcal{L}^* \begin{pmatrix} 0 & 0 \\ \frac{1}{\Delta t} I & 0 \end{pmatrix}.$$

After some algebra,

$$\mathcal{U} = \begin{pmatrix} -\beta^2\Delta + \frac{1}{\Delta t^2}I & -\beta^2 C^* + \beta\frac{1}{\Delta t}D^* \\ -\beta^2 C + \beta\frac{1}{\Delta t}D & \beta^2(E-\Delta) \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{\Delta t^2}I & 0 \\ \beta\frac{1}{\Delta t}D & 0 \end{pmatrix} \mathcal{U}_m, \tag{5.40}$$

where $-\Delta = \begin{pmatrix} -\nabla^2 & 0 \\ 0 & -\nabla^2 \end{pmatrix}$ and $C = \begin{pmatrix} \nabla\times \\ 0 \end{pmatrix}$. These, along with D are now the discrete operators. To simplify even further, we perform a singular value decomposition on the discretization of D:

$$D = U\Sigma V^*,$$

where Σ is a diagonal matrix containing the singular values, σ_j, $j = 1, ..., n$. Then, after performing a similarity transformation on $\mathcal{L}$, we obtain a system that has the same eigenvalues and, thus, the same spectral radius as the original system,

$$\begin{pmatrix} \beta^2\Sigma^2 + \frac{1}{\Delta t^2}I & -\beta^2\Sigma + \beta\frac{1}{\Delta t}\Sigma \\ -\beta^2\Sigma + \beta\frac{1}{\Delta t}\Sigma & \beta^2(E+\Sigma^2) \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{\Delta t^2}I & 0 \\ \beta\frac{1}{\Delta t}\Sigma & 0 \end{pmatrix}. \tag{5.41}$$

Since this is a 2 by 2 block matrix consisting of n by n (the number of singular values) diagonal matrices, we can look at the scalar 2 by 2 form of the matrix using the singular values of the discretized D operator. Hence, for every singular value, σ_j, the propagation matrix is of the form

$$\begin{pmatrix} \beta^2\sigma_j^2 + \frac{1}{\Delta t^2} & -\beta^2\sigma_j + \beta\frac{1}{\Delta t}\sigma_j \\ -\beta^2\sigma_j + \beta\frac{1}{\Delta t}\sigma_j & \beta^2(1+\sigma_j^2) \end{pmatrix}^{-1} \begin{pmatrix} \frac{1}{\Delta t^2}I & 0 \\ \beta\frac{1}{\Delta t}\sigma_j & 0 \end{pmatrix}. \tag{5.42}$$

The propagation matrix is then reduced to

$$\begin{pmatrix} \frac{\frac{1}{\Delta t}}{\frac{1}{\Delta t} + \beta\sigma_j^2} & 0 \\ \frac{\frac{1}{\Delta t}\sigma_j}{\frac{1}{\Delta t} + \beta\sigma_j^2} & 0 \end{pmatrix}. \tag{5.43}$$

Being a lower triangular matrix, this system has a spectral radius of $\frac{\frac{1}{\Delta t}}{\frac{1}{\Delta t}+\beta\sigma_j^2}$. Since β is positive for BDF schemes of order less than 7, the spectral radius of this propagation matrix is bounded by one. Thus, we have a stable system. All modes decay at the appropriate rate and errors in the timestepping do not blow up. Starting with some initial condition, if we take several time steps, the solution looks like the following at time T:

$$\mathcal{U}_T = \begin{pmatrix} \frac{\frac{1}{\Delta t}}{\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \\ \frac{\frac{1}{\Delta t}\sigma_j}{\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \end{pmatrix}^{\frac{T}{\Delta t}} \mathcal{U}_0. \tag{5.44}$$

Taking the limit as Δt goes to zero, we see that the modes decay as expected,

$$\lim_{\Delta t\to 0} \begin{pmatrix} \frac{\frac{1}{\Delta t}}{\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \\ \frac{\frac{1}{\Delta t}\sigma_j}{\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \end{pmatrix}^{\frac{T}{\Delta t}} = \begin{pmatrix} e^{-\beta\sigma_j^2 T} & 0 \\ \sigma_j e^{-\beta\sigma_j^2 T} & 0 \end{pmatrix}. \tag{5.45}$$

It should be noted here that the Reynolds number scaling used in the current-vorticity formulation described above does not affect this result. Including that scaling, one would obtain the following propagation matrix:

$$\begin{pmatrix} \frac{R_e\frac{1}{\Delta t}}{R_e\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \\ \frac{\frac{1}{\Delta t}\sigma_j}{R_e\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \end{pmatrix}. \tag{5.46}$$

Also, it can be shown that

$$\lim_{\Delta t\to 0} \begin{pmatrix} \frac{R_e\frac{1}{\Delta t}}{R_e\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \\ \frac{\frac{1}{\Delta t}\sigma_j}{R_e\frac{1}{\Delta t}+\beta\sigma_j^2} & 0 \end{pmatrix}^{\frac{T}{\Delta t}} = \begin{pmatrix} e^{-\frac{\beta\sigma_j^2}{R_e}T} & 0 \\ \frac{\sigma_j}{R_e}e^{-\frac{\beta\sigma_j^2}{R_e}T} & 0 \end{pmatrix}. \tag{5.47}$$

The spectral radius is still less than one and the decay of error modes is reduced at the appropriate rates, decreasing more quickly with higher Reynolds Number.

Performing a similar analysis on the Crank-Nicholson scheme, shows that we do not get the same nice properties. The Crank-Nicholson formula is as follows:

$$\frac{\partial \mathcal{U}}{\partial t} = L(\mathcal{U}) \Rightarrow \mathcal{U}_n - \mathcal{U}_{n-1} = \frac{\Delta t}{2}(L(\mathcal{U}_n) + L(\mathcal{U}_{n-1})). \tag{5.48}$$

Substituting equation (5.48) into the system (5.32) now gives us

$$\mathcal{L}\mathcal{U}_n = \begin{pmatrix} D & -E \\ \frac{2}{\Delta t}I & D^* \end{pmatrix} \mathcal{U}_n = \begin{pmatrix} -D & E \\ \frac{2}{\Delta t}I & -D^* \end{pmatrix} \mathcal{U}_{n-1}. \tag{5.49}$$

We rearrange this for simplicity to get

$$\begin{pmatrix} D & -E \\ \frac{2}{\Delta t}I & D^* \end{pmatrix} \mathcal{U}_n = \left[\begin{pmatrix} 0 & 0 \\ \frac{4}{\Delta t}I & 0 \end{pmatrix} - \begin{pmatrix} D & -E \\ \frac{2}{\Delta t}I & D^* \end{pmatrix} \right] \mathcal{U}_{n-1}. \tag{5.50}$$

This leads to the following propagation matrix:

$$\begin{pmatrix} D^*D + \frac{4}{\Delta t}I & -D^*E + \frac{2}{\Delta t}D^* \\ -ED + \frac{2}{\Delta t}D & E + DD^* \end{pmatrix}^{-1} \begin{pmatrix} \frac{8}{\Delta t^2}I & 0 \\ \frac{4}{\Delta t}D & 0 \end{pmatrix} - \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}. \tag{5.51}$$

Applying the same singular value decomposition trick, we can reduce the propagation matrix to a 2 by 2 scalar matrix and obtain a diagonal entry of one in the vorticity-pressure block:

$$\begin{pmatrix} \frac{\frac{2}{\Delta t} - \sigma_j^2}{\frac{2}{\Delta t} + \sigma_j^2} & 0 \\ \frac{\frac{4}{\Delta t}\sigma_j}{\frac{2}{\Delta t} + \sigma_j^2} & 1 \end{pmatrix}. \tag{5.52}$$

Now, any error we introduce in vorticity or pressure is propagated at every time step with no decay. Therefore, if we do not solve these equations well enough, the error will accumulate and eventually dominate our solution over time. Thus, in a FOSLS setting it is best to use the BDF-k schemes for timestepping. In addition,

the result from the DAE analysis in [32] shows that, if the constraint equations are solved well enough, we still get the accuracy that is expected.

Generalizing these results to the full MHD system can be difficult. Not much has been proved on the stability analysis for nonlinear time-dependent partial differential equations. In [25, 40, 41, 42], it has been shown for nonlinear fluid flow that the Crank-Nicholson scheme can be unstable under certain conditions. This supports the argument for using the BDF schemes.

Consider the difficulty in going from Stokes to Navier Stokes. Assume the first order system becomes

$$\mathcal{L} = \begin{pmatrix} D & -E \\ \frac{1}{\Delta t} + W & D^* + U \end{pmatrix}. \tag{5.53}$$

The matrices D and E are the same as above, possibly with a Reynolds number scaling, and W and U come from the linearization of the term $\mathbf{u} \times \boldsymbol{\omega}$:

$$W = \begin{pmatrix} 0 & -\omega^n \\ \omega^n & 0 \end{pmatrix} \quad U = \begin{pmatrix} -u_2^n & 0 \\ u_1^n & 0 \end{pmatrix}. \tag{5.54}$$

The superscript, n, indicates a finite element solution from the previous linearization. The propagation matrix for a BDF scheme is again a 2 by 2 block with 0 entries in the second column. However, the propagation matrix for the modes in $\mathbf{u}$ is

$$\frac{1}{\Delta t}(D + E(D^* + U)^{-1}(\frac{1}{\Delta t}I + W))^{-1}E(D^* + U)^{-1}$$

The norm of this block depends on $\mathbf{u}^n$ and ω^n. Therefore, determining stability is highly problem dependent.

If we now define $\mathcal{L}_s$ to be the Stokes system described above, we can rewrite the full incompressible resistive MHD equations in 2D as follows:

$$\left(\begin{array}{c|c} \mathcal{L}_s & 0 \\ \hline 0 & \mathcal{L}_s \end{array}\right) \left(\begin{array}{c} \mathbf{u} \\ \boldsymbol{\omega} \\ \hline \mathbf{B} \\ \mathbf{j} \end{array}\right) + \left(\begin{array}{cc|cc} 0 & 0 & 0 & 0 \\ W & U & J & B \\ \hline 0 & 0 & 0 & 0 \\ G_b & 0 & G_u & 0 \end{array}\right) \left(\begin{array}{c} \mathbf{u} \\ \boldsymbol{\omega} \\ \hline \mathbf{B} \\ \mathbf{j} \end{array}\right).$$

Again, $\boldsymbol{\omega} = (\omega, p)^T$ and, similarly, $\mathbf{j} = (j, \gamma)^T$, where γ is a slack variable to make the system square. As in Chapter 4, appropriate boundary conditions can be chosen to make γ zero. The matrices W and U are the same as in equation (5.54), corresponding to the linearization of $\mathbf{u} \times \boldsymbol{\omega}$. Analogously, matrices J and B come from the linearization of $\mathbf{j} \times \mathbf{B}$:

$$J = \begin{pmatrix} 0 & -j^n \\ j^n & 0 \end{pmatrix} \qquad B = \begin{pmatrix} -B_2^n & 0 \\ B_1^n & 0 \end{pmatrix}. \tag{5.55}$$

Matrices G_b and G_u come from the linearization of $\mathbf{u} \cdot \nabla \mathbf{B} - \mathbf{B} \cdot \nabla \mathbf{u}$:

$$G_b = \begin{pmatrix} \partial_x(B_1^n) - \mathbf{B}^n \cdot \nabla & \partial_y(B_1^n) \\ \partial_x(B_2^n) & \partial_y(B_2^n) - \mathbf{B}^n \cdot \nabla \end{pmatrix} \tag{5.56}$$

and

$$G_u = \begin{pmatrix} \mathbf{u}^n \cdot \nabla - \partial_x(u_1^n) & -\partial_y(u_1^n) \\ -\partial_x(u_2^n) & \mathbf{u}^n \cdot \nabla - \partial_y(u_2^n) \end{pmatrix}. \tag{5.57}$$

The diagonal part of this system gives us the same results as for Stokes alone, since it has the same structure. The off-diagonal nonlinear entries, though, depend on the linearized solutions in the finite element space. Thus, from problem to problem, there could be varying results.

For this thesis and the test problems studied, we confirm numerically that stable results for the full nonlinear system are obtained. Figures 5.5 and 5.6 show that with the FOSLS formulation on a 2D reduced MHD system, which is described more in Chapter 6, we get the expected convergence of the time-stepping

schemes. For the figures, a system is solved with a very small time step, $0.04\tau_A$. The time step is increased on sucessive runs and the L^2 norm of the difference between the solution with the current time step and the solution with the small time step is observed. For BDF-1, we get $O(\Delta t)$ convergence as expected and for BDF-2, we see near $O(\Delta t^2)$ convergence. Therefore, we are confident that the BDF schemes are adequate when coupled with our solution algorithm described above and in the context of a FOSLS formulation.

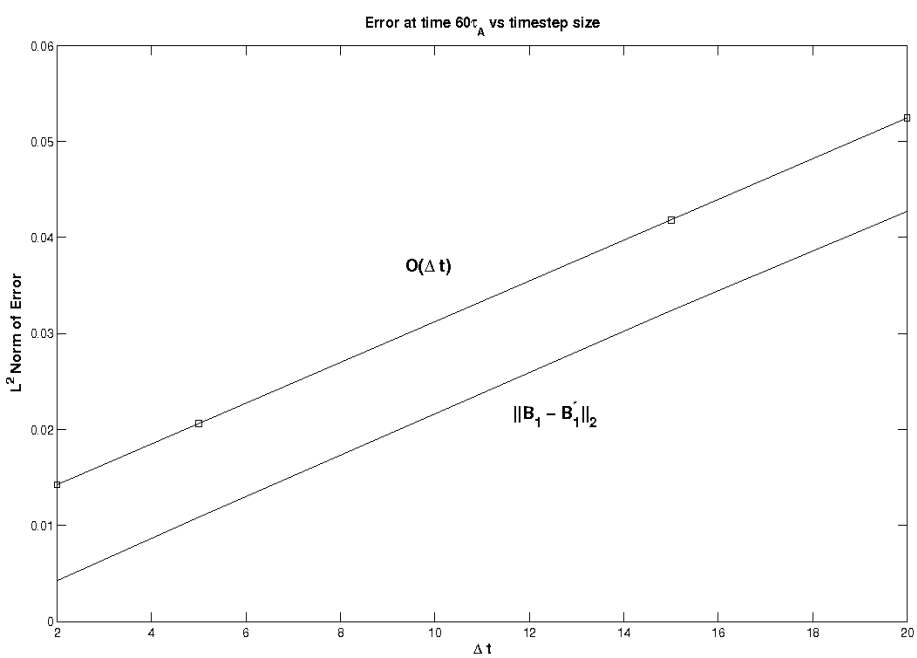

Figure 5.5: $O(\Delta t)$ convergence of the BDF-1 (Backward-Euler) method on the MHD system.

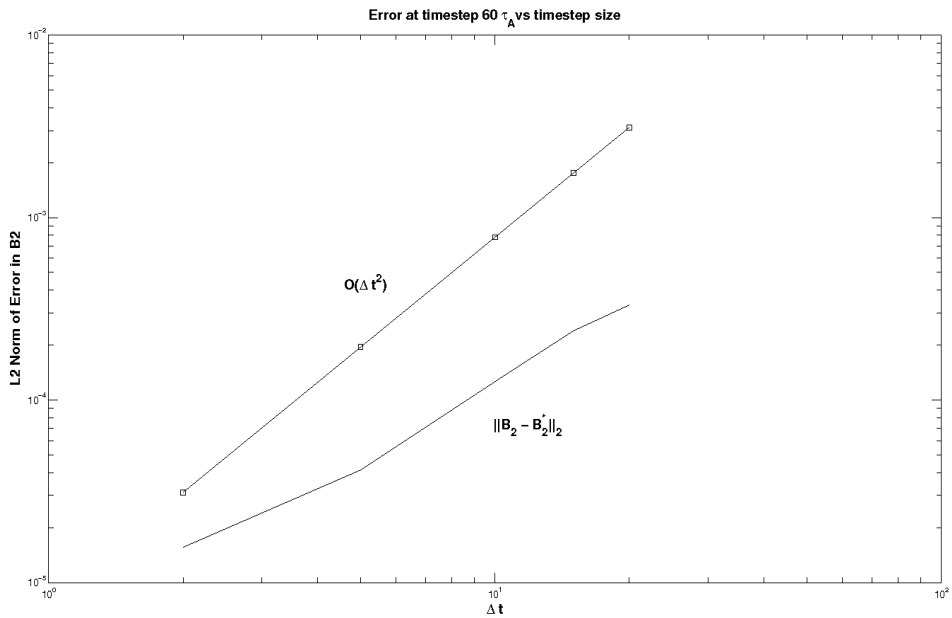

Figure 5.6: $O(\Delta t^2)$ convergence of the BDF-2 method on the MHD system.

Now, with the above discretization and solution algorithm, we can begin to test our methods.

Chapter 6

Numerical Results

In this chapter, we investigate several MHD test problems to show that the nested iteration Newton-FOSLS AMG method works well. We show that these methods are capable of solving the complex nonlinear systems in a minimal amount of work units or fine-grid relaxations.

6.1 3D Test Problem

Our first test of the Newton-FOSLS approach on incompressible resistive MHD is a steady-state test problem with manufactured solutions. The first-order system is as in equations (4.21)-(4.28) with the time-derivatives removed. Our goal here is to show how well a straight, nested iteration algorithm performs on the complex MHD system. We do not use any stopping criteria as described in Chapter 5. This is to show that the work done on coarse grids allows solutions to be approximated on the fine grids with a minimal amount of work. We ignore the efficiency of computing on the coarse grids for now. Let the true solution be

as follows:

$$\mathbf{u}(x,y,z) = \begin{pmatrix} \sin(\pi x)(\cos(\pi y) - \cos(\pi z)) \\ \sin(\pi y)(\cos(\pi z) - \cos(\pi x)) \\ \sin(\pi z)(\cos(\pi x) - \cos(\pi y)) \end{pmatrix}, \quad \boldsymbol{\omega} = \begin{pmatrix} 2\pi \sin(\pi y)\sin(\pi z) \\ 2\pi \sin(\pi z)\sin(\pi x) \\ 2\pi \sin(\pi x)\sin(\pi y) \end{pmatrix},$$

$$\mathbf{B}(x,y,z) = \begin{pmatrix} \sin(\pi x)(\cos(\pi y) - \cos(\pi z)) \\ \sin(\pi y)(\cos(\pi z) - \cos(\pi x)) \\ \sin(\pi z)(\cos(\pi x) - \cos(\pi y)) \end{pmatrix}, \quad \mathbf{j} = \begin{pmatrix} 2\pi \sin(\pi y)\sin(\pi z) \\ 2\pi \sin(\pi z)\sin(\pi x) \\ 2\pi \sin(\pi x)\sin(\pi y) \end{pmatrix},$$

$$p = 1 - z.$$

On a cube domain, $\Omega = [0,1]^3$, we have the following boundary conditions for a perfect conductor:

On all faces:

$$\begin{aligned} \mathbf{n} \cdot \mathbf{u} &= 0, \\ \mathbf{n} \cdot \mathbf{B} &= 0, \\ \mathbf{n} \times \mathbf{E} = \mathbf{n} \times (\mathbf{j} - S_L(\mathbf{u} \times \mathbf{B})) &= 0, \\ \mathbf{n} \times \boldsymbol{\omega} &= 0. \end{aligned}$$

On the in/out boundaries:

$$p = \begin{cases} 1 & \text{inflow wall} \quad z = 0 \\ 0 & \text{outflow} \quad z = 1. \end{cases}$$

This allows the right-hand sides of the MHD equations to be built and gives a well-posed problem, for which there is a known solution.

6.1.1 Nested Iteration Tests

To test our nested iteration strategy, the first-order system, (4.21)-(4.28), is solved on several grids using bilinear elements. The problem is first solved on a grid with spacing $h = 1/2$. The system is oversolved, in that as many Newton steps as needed are performed for the linear functional to converge to the nonlinear functional to within 5 digits. At each Newton step, the linear system is solved with as few V-cycles as possible to reach discretization error. The solution is then moved up to the next level using uniform refinement to get a grid spacing of $h = 1/4$ and the process is repeated. The test was initialized with a random initial guess on the coarsest grid. The results are shown in table 5.1.

h	Newton Step	Linear Functional	Nonlinear Functional	V-cycles
1/2	1	15.758	15.910	4
1/2	2	14.823	14.909	3
1/2	3	14.712	14.760	3
1/2	4	14.710	14.727	3
1/2	5	14.715	14.720	2
1/2	6	14.717	14.718	2
1/2	7	14.718	14.718	2
1/4	1	8.195	8.207	5
1/4	2	8.146	8.144	4
1/4	3	8.142	8.142	3
1/8	1	4.664	4.663	10
1/8	2	4.657	4.657	5
1/16	1	1.768	1.768	15

Table 6.1: Nested iteration tests for 3D problem. Shows number of Newton steps and V-cycles used on each level or refinement.

On finer and finer grids, we see that the number of Newton steps goes to one with only a few V-cycles being used. By the time the finest grid is reached, only one Newton-step is needed. The key feature here is that the work done on the coarse grids gives a good approximation of the solution on the fine grid and, therefore,

not much computation is needed to solve the system at that level. To judge how well this approach performs, we next determine how much work is actually being done.

6.1.1.1 Work

Again, a work unit (WU) is defined to be the cost of one matrix-vector multiplication at the finest level. Multigrid theory tells us that the cost of one V(1,1) cycle in 3D at that level is 16/7 WU [12]. Since going to a coarser level (h is reduced by a half) decreases the number of unknowns by 1/8, the total work at a given level is $(\frac{1}{8^L})(\frac{16k}{7})$ WU, where L = 0 is the finest level and k is the number of V-cycles performed. Ignoring setup cost, we can crudely consider the work of one Newton step with k V-cycles at that level to be $\alpha\frac{16k}{7}$ WU, so the total work at a given level is

$$\sum_{i=1}^{m} \alpha(\frac{1}{8^L})(\frac{16k_i}{7}), \tag{6.1}$$

where m is the number of Newton steps, k_i is the number of V-cycles for the ith Newton step, and α is such that a Newton step is α times more work than a V-cycle.

For now, assume $\alpha = 1$. In reality, the Newton step is more costly and depends on the amount of nonlinearity and the FOSLS setup. Thus, we want to perform fewer of them. For this test problem, where the finest level has a grid spacing of $h = 1/16$ and discretization error is reached on each level, we have

- 0.085 WU for h=1/2,
- 0.429 WU for h=1/4,
- 4.286 WU for h=1/8,

- 34.286 WU for h=1/16.

This leads to about $39WU$ to get the results. This shows that by performing most of the linearizations on coarser grids, it is possible to get better coarse grid estimates to the solution for finer grids and, thus, less work is needed on the finer grids to get the desired solution. The above results show that by the time the finest grid is reached, the complex nonlinear system is solved with only one Newton linearization and a few V-cycles. Therefore, nested iteration is crucial to solving these systems efficiently. This also shows that the FOSLS discretization allowed the large 16 by 13 system to be solved correctly, thus making it a viable method for MHD. Next, we see how well this process does for a fusion application.

6.2 2D Reduced MHD

The full algorithm, as in figure 5.4, was next applied to a more realistic test problem [16, 17, 39, 47]. From the papers by Chacón, Knoll, and Finn [16] and Philip [39], a reduced set of MHD equations is obtained. These equations simulate a "large aspect-ratio" tokamak, with non-circular cross-sections. Here, the magnetic B-field along the z-direction, or the toroidal direction, is very large and mostly constant. In this context, we are able to look at plasma behavior in the poloidal cross-section. Consider a regular torus as in figure 6.1. With the strong magnetic field in the toroidal direction and with $R >> r$, the fluid motion in the poloidal plane is studied. The tokamak is modeled as a periodic cylinder as in figure 6.2. The poloidal plane is modeled as a cross-section of this cylinder. Finally, a thin annulus in the poloidal plane is modeled as a periodic strip in the x-y plane. See figure 6.3.

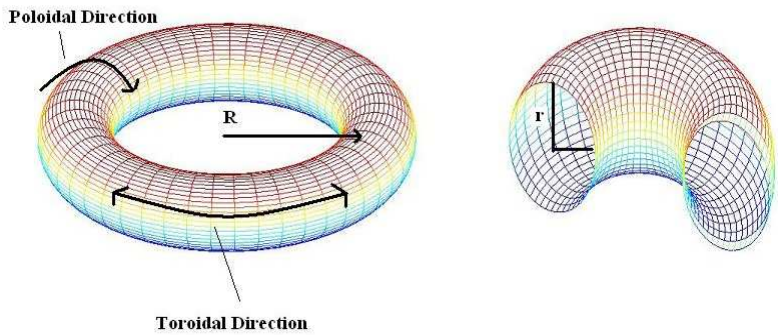

Figure 6.1: Torus-shaped tokamak.

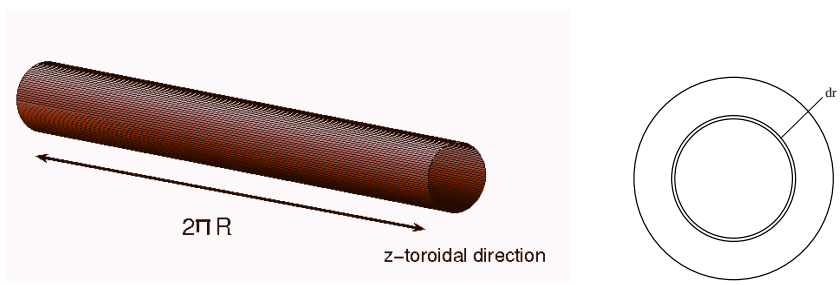

Figure 6.2: Cross-section in poloidal direction.

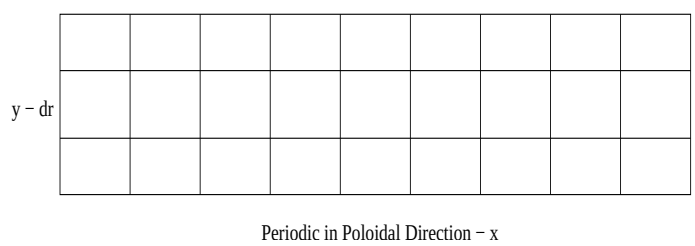

Figure 6.3: Cartesian grid upon which the problem is solved.

In this context, assume B_3 is a constant and

$$\mathbf{B} = \begin{pmatrix} B_1(x,y) \\ B_2(x,y) \\ B_3 >> 1 \end{pmatrix} \quad \Rightarrow \mathbf{j} = \begin{pmatrix} 0 \\ 0 \\ j_3(x,y) \end{pmatrix},$$

$$\mathbf{u} = \begin{pmatrix} u_1(x,y) \\ u_2(x,y) \\ 0 \end{pmatrix} \quad \Rightarrow \boldsymbol{\omega} = \nabla \times \mathbf{u} = \begin{pmatrix} 0 \\ 0 \\ \omega_3(x,y) \end{pmatrix}.$$

A poloidal velocity stream function, Φ, and a poloidal flux, Ψ, are introduced, such that

$$-\nabla^{\perp}\Psi = \mathbf{B}, \qquad -\nabla^{\perp}\Phi = \mathbf{u}.$$

This enforces the incompressibility of $\mathbf{u}$ and $\mathbf{B}$. The reduced equations become

$$\frac{\partial \Psi}{\partial t} + \mathbf{u}\cdot\nabla\Psi - \frac{1}{\sigma\mu_0}\nabla^2\Psi = -E_0, \tag{6.2}$$

$$\rho\frac{\partial \omega_3}{\partial t} + \rho\mathbf{u}\cdot\nabla\omega_3 - \rho\nu\nabla^2\omega_3 - \mathbf{B}\cdot\nabla j_3 = -S_\omega, \tag{6.3}$$

$$\nabla^2\Phi = \omega_3, \tag{6.4}$$

$$\nabla^2\Psi = \mu_0 j_3. \tag{6.5}$$

For a complete derivation, see [47]. The functions E_0 and S_w are source functions used to balance the decay of the equilibrium solution due to transport terms. They are obtained by plugging in the initial conditions to the equation and dropping the full time derivative, $D/Dt = \dfrac{\partial}{\partial t} + \mathbf{u}\cdot\nabla$.

By applying $-\nabla^{\perp}$ to equation (6.2), we arrive at

$$\frac{\partial \mathbf{B}}{\partial t} - \nabla\times(\mathbf{u}\times\mathbf{B}) + (\nabla\times\frac{\mathbf{j}}{\sigma}) = \nabla^{\perp}E_0, \tag{6.6}$$

which is exactly equation (4.2) in the original formulation plus the source term. Also note that, since $\nabla\cdot\mathbf{u} = 0$ and $\nabla\cdot\mathbf{B} = 0$, we have $\nabla\times(\mathbf{u}\times\mathbf{B}) = \mathbf{B}\cdot\nabla\mathbf{u} - \mathbf{u}\cdot\nabla\mathbf{B}$. By taking the curl of the Navier-Stokes part of the original formulation, equation (4.1), and using the above assumptions, equation (6.3) is obtained. Thus, the reduced model, equations (6.2)-(6.5), is equivalent to the 2D model derived in

Chapter 4, equations (4.21)-(4.28). We now apply our methodology to two test problems, known as the tearing mode and the island coalescence problem.

6.2.1 Test Problem: Tearing Modes

The first test problem simulates a tearing mode instability arising from perturbations in a steady-state current density sheet of a tokamak generator. In this case, a current density sheet in the toroidal direction of the tokamak is perturbed, resulting in an instability that causes a reconnection in the magnetic field lines. The result is a formation of an island in the contours of the current density field and a "tearing" in the contours of the vorticity. More details can be found in [3, 16, 17, 39]. For the following simulations, we define

$$\begin{aligned} \Omega &= [0,4] \times [0,1], \\ R_e &= S_L = 1000. \end{aligned}$$

The initial conditions at equilibrium are

$$\mathbf{B}_0(x,y) = \begin{pmatrix} -\tanh(\frac{1}{\lambda}(y-1/2)) + \frac{4}{3\lambda}((y-1/2)^3 sech^2(\frac{1/2}{\lambda}) \\ 0 \end{pmatrix}, \quad (6.7)$$

$$\mathbf{u}_0(x,y) = \mathbf{0}, \quad (6.8)$$

$$\omega_0(x,y) = 0, \quad (6.9)$$

$$j_{30}(x,y) = -\frac{4}{\lambda}(y-1/2)^2 sech^2(\frac{1/2}{\lambda}) + \frac{1}{\lambda}(sech^2(\frac{y-1/2}{\lambda})), \quad (6.10)$$

$$p_0(x,y) = 0.303 + \frac{1}{2}|\mathbf{B}_0|^2, \quad (6.11)$$

where $\lambda = 0.2$ and $\epsilon = 0.1$. These initial conditions are perturbed away from

equilibrium as follows:

$$\delta\mathbf{B}_0(x,y) = \begin{pmatrix} -\epsilon\pi\cos(\pi y)\sin(\pi x/2-\pi/2) \\ \frac{1}{2}\epsilon\pi\sin(\pi y)\cos(\pi x/2-\pi/2) \\ 0 \end{pmatrix}, \tag{6.12}$$

$$\delta j_{30}(x,y) = -\frac{5}{4}\epsilon\pi^2\sin(\pi y)\sin(\pi x/2-\pi/2), \tag{6.13}$$

$$\delta p_0(x,y) = \frac{5}{8}\epsilon^2\pi^2\sin^2(\pi y)\sin^2(\pi x/2-\pi/2). \tag{6.14}$$

The boundary conditions are periodic in x and homogeneous Dirichlet for the current density and vorticity on the top and bottom of Ω. We also have $\mathbf{n}\cdot\mathbf{u}=0$ and $\mathbf{n}\cdot\mathbf{B}=0$ on the top and bottom. Again, this allows for the FOSLS formulation we prescribe to be H^1 elliptic for the Navier-Stokes and the Maxwell blocks and for the whole linearized system as shown in Chapter 4. We, therefore, expect our algorithm to perform well on this tearing mode problem and to resolve the physical phenomena.

6.2.2 Results

The problem was run to time $200\tau_A$ with a timestep of $1\tau_A$ using a BDF-2 timestepping scheme. Again, τ_A is the time in Alfvén units. At this point, the tearing mode came into full gear. The nested iteration algorithm was applied at every timestep. Stopping criteria, as described in chapter 5, were used for the AMG V-cycles and the linearizations so that the problem was solved to within a specified fraction of the discretization error. In other words, minimal work was done to obtain the solution on each grid. As the results show, this was sufficient to faithfully produce the relevant physics on a 64×256 fine grid with quadratic elements. This requires $463,239$ degrees of freedom. The linear stopping

parameter was

$$F_s^2 \leq 1.01 F_*^2.$$

The nonlinear stopping parameter was

$$\frac{|F_s^2 - \mathcal{F}_s^2|}{\mathcal{F}_s^2} \leq 0.1.$$

On the finest grid and for later time steps, only 1 Newton step was needed for the linearization to converge and only a small number of V-cycles were needed for the linear solves to achieve the desired accuracy.

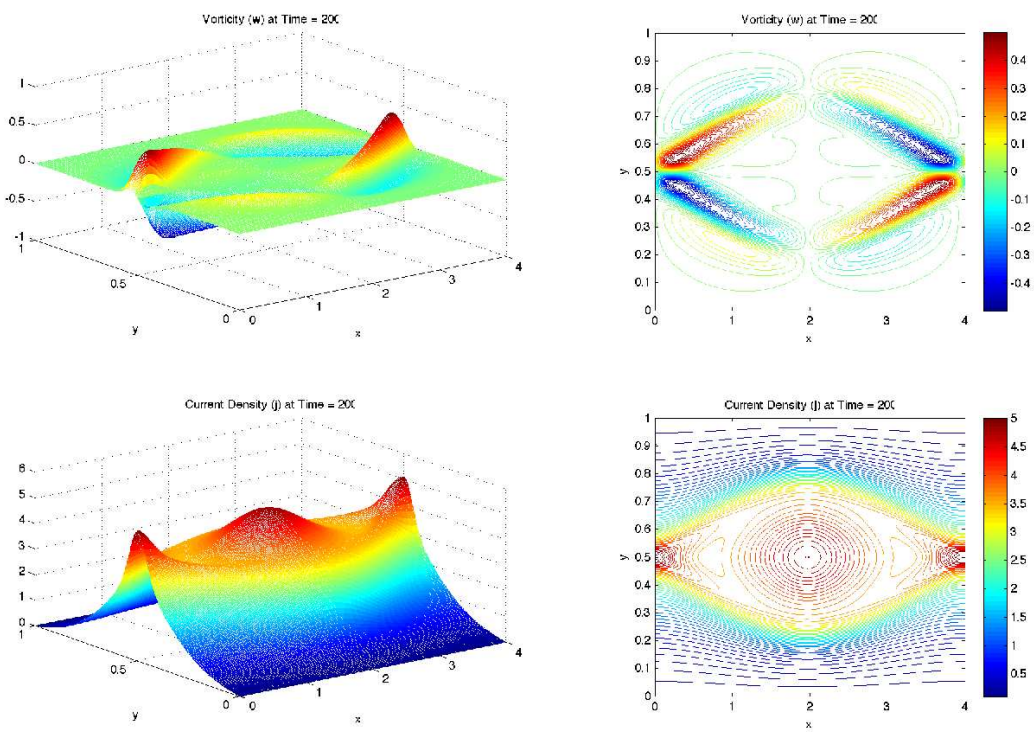

Figure 6.4: Numerical solution after $200\tau_A$. $h = 1/64$ $p = 2$ using uniform refinement. $R_e = S_L = 1000$. Top: vorticity. Bottom: current density.

Figure 6.4 shows that the method resolves the problem correctly. The main properties of the tearing mode have been captured. As was done for the 3D

problem, the total work is shown in table 6.2. These results show that, using the FOSLS formulation, we were capable of capturing complex instabilities in a fusion reactor simulation in 30-65 work units. For later time steps, we do see a deterioration in AMG convergence on finer grids. Thus, we have to address the scalability of our AMG solver. However, FOSLS does give a discretization that AMG can solve accurately and efficiently, especially in the context of a nested iteration algorithm. It also gives sharp error estimates that aid in continuously calculating the accuracy-per-computational-cost.

Time	Grid	Nwt Steps	Avg V-cycles	WU	Avg WU/Timestep
1-86	1	2	7.5	0.020	
1-86	2	2	10.9	0.114	
1-86	3	2	14.4	0.601	
1-86	4	2	17.9	2.986	$\approx$ 27 WU
1-86	5	1	16	10.636	
1-86	6	1	4.9	12.961	
87-200	1	2	8	0.021	
87-200	2	2	11	0.115	
87-200	3	2	18.5	0.770	
87-200	4	2	21	3.497	$\approx$ 64 WU
87-200	5	1	17.2	13.947	
87-200	6	1	43.4	45.988	

Table 6.2: Number of Newton steps and V-cycles used at each level and timestep. The number of work units (WU) or equivalent fine-grid relaxations are also computed here.

Next, we show that, by using an adaptive refinement scheme such as ACE [22, 37], we can get the same qualitative behavior with fewer degrees of freedom than using uniform refinement as is done above. Figure 6.5, shows the development of the tearing mode on an adaptively refined mesh. Here, we can see that the algorithm is designed so that refinement occurs in areas where the reconnection happens. This is where steep gradients develop and, therefore, we expect larger errors to develop.

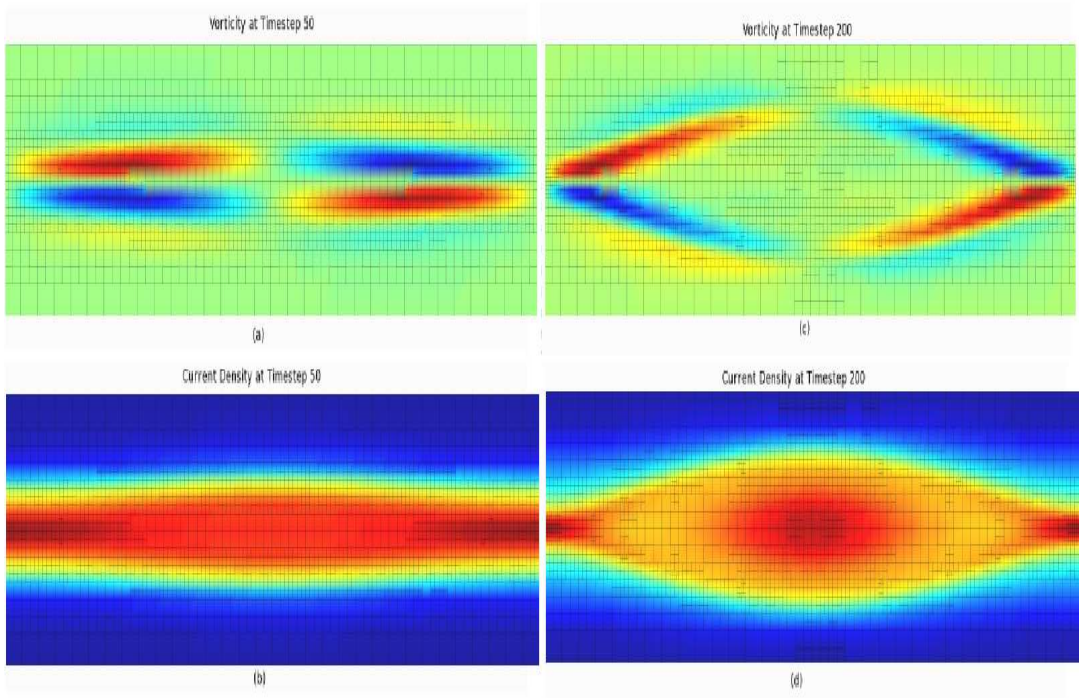

Figure 6.5: Numerical solution using 7 levels of adaptive refinement. $R_e = S_L = 1000$. (a) Vorticity at Time $50\tau_A$. (b) Current Density at Time $50\tau_A$. (c) Vorticity at Time $200\tau_A$.(d) Current Density at Time $200\tau_A$.

Comparing the amount of work on one adaptive grid to another, or comparing to a uniformly refined grid is difficult. A relaxation on the finest grid of an adaptive mesh has a different complexity than that of a uniform mesh. We can, though, compare the number of non-zeros in the algebraic system for each grid. This gives us an idea of how much work a V-cycle does.

Assume that we are on a specific level of our nested iteration grids, grid l. Define n to be the maximum number of levels in a V-cycle for grid l, r_j to be the number of relaxations per V-cycle at level j on grid l, nz_j to be the number of non-zeros in the algebraic system at level j on grid l, and V to be the number of V-cycles performed on grid l. For all tests, we are using $V(1,1)$ cycles and, therefore, $r_j = 2$. Assume that the finest level of the V-cycle is $j = 0$. Let

$R_l = \dfrac{\sum_{j=0}^{n} nz_j * r_j}{nz_0} * V$ be the number of equivalent relaxations on the given grid, l, of the recent timestep and specific scheme. The number of nonzeros in the algebraic system on grid l, NZ_l, is equivalent to the finest level non-zeros of a V-cycle on that grid, $NZ_l = nz_0$. With these relations, one can compute the equivalent fine grid relaxations, or work units, for each timstep. This is denoted by

$$R = \frac{\sum_{i=0}^{N} R_i * NZ_i}{NZ_0}.$$

Thus, R is the number of work units, relative to the finest grid, to resolve the problem on a given time step. For the following tables, we also define

$NZ = NZ_0 =$ Non-zeros on finest grid of a given time step,

$Rat = R * NZ$ of specific scheme divided by $R * NZ$ of uniform scheme,

$NF =$ The nonlinear functional squared (i.e. $||\mathcal{L}(\mathbf{u}) - \mathbf{f}||_0^2$).

Table 6.3 uses these relations to compare the amount of work needed to achieve a given functional value using both uniform refinement and ACE.

		Uniform	6	Levels		ACE	7	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
1	38.36	12,983,383	1	5.74e-05	40.09	6,989,311	0.56	9.29e-05
5	41.58	12,983,383	1	5.37e-05	12.62	18,907,287	0.44	1.96e-05
10	41.59	12,983,383	1	5.36e-05	12.61	18,888,471	0.44	1.94e-05
15	41.60	12,983,383	1	5.37e-05	12.64	18,832,023	0.44	1.95e-05
20	41.60	12,983,383	1	5.38e-05	37.14	7,528,213	0.52	7.19e-05
25	41.61	12,983,383	1	5.40e-05	56.89	19,881,701	2.09	1.43e-05
30	41.58	12,983,383	1	5.46e-05	41.11	18,908,855	1.44	1.61e-05

		Uniform	6	Levels		ACE	7	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
35	41.58	12,983,383	1	5.44e-05	59.13	5,927,579	0.65	1.12e-04
40	41.63	12,983,383	1	5.53e-05	43.98	6,460,993	0.53	9.66e-05
45	57.60	12,983,383	1	5.44e-05	52.90	6,452,467	0.46	9.16e-05
50	57.60	12,983,383	1	5.49e-05	58.34	6,815,851	0.53	7.78e-05
55	57.57	12,983,383	1	5.55e-05	58.30	6,699,329	0.52	7.98e-05
60	57.81	12,983,383	1	5.61e-05	46.93	6,759,011	0.42	7.83e-05
65	57.67	12,983,383	1	5.66e-05	43.22	6,905,619	0.40	7.68e-05
70	57.79	12,983,383	1	5.69e-05	40.76	6,847,505	0.37	7.54e-05
75	64.26	12,983,383	1	5.74e-05	41.67	7,586,131	0.38	6.54e-05
80	64.15	12,983,383	1	5.68e-05	44.61	8,070,839	0.43	5.82e-05
85	64.29	12,983,383	1	5.65e-05	50.10	5,867,799	0.35	9.40e-05
90	64.36	12,983,383	1	5.77e-05	50.15	5,746,475	0.34	9.93e-05
95	64.20	12,983,383	1	5.72e-05	50.50	5,915,427	0.36	9.48e-05
100	64.20	12,983,383	1	5.67e-05	47.61	6,538,119	0.37	8.22e-05
105	64.26	12,983,383	1	5.59e-05	41.34	7,608,671	0.38	6.40e-05
110	64.13	12,983,383	1	5.52e-05	47.87	5,693,359	0.33	1.06e-04
115	64.07	12,983,383	1	5.44e-05	50.69	4,386,529	0.27	2.06e-04
120	64.32	12,983,383	1	5.37e-05	50.83	4,264,715	0.26	2.25e-04
125	67.37	12,983,383	1	5.28e-05	53.70	4,197,291	0.26	2.41e-04
130	67.56	12,983,383	1	5.22e-05	50.70	5,726,679	0.33	1.04e-04
135	67.53	12,983,383	1	5.16e-05	50.85	5,799,983	0.34	1.02e-04
140	67.54	12,983,383	1	5.10e-05	50.49	5,824,385	0.34	10.00e-05
145	67.45	12,983,383	1	5.04e-05	50.79	6,173,951	0.36	8.91e-05
150	67.44	12,983,383	1	5.00e-05	54.07	5,022,255	0.31	1.45e-04

		Uniform	6	Levels		ACE	7	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
155	67.64	12,983,383	1	4.95e-05	48.05	8,437,065	0.46	4.68e-05
160	67.60	12,983,383	1	4.91e-05	47.74	7,248,423	0.39	6.75e-05
165	67.55	12,983,383	1	4.87e-05	50.91	7,338,191	0.43	6.74e-05
170	67.66	12,983,383	1	4.84e-05	54.27	5,807,137	0.36	1.043e-05
175	70.74	12,983,383	1	4.84e-05	53.65	5,954,039	0.35	9.88e-05
180	70.72	12,983,383	1	4.81e-05	50.71	6,323,695	0.35	8.42e-05
185	70.72	12,983,383	1	4.79e-05	47.59	5,996,277	0.31	9.24e-05
190	70.70	12,983,383	1	4.77e-05	54.43	7,794,577	0.46	6.19e-05
195	70.69	12,983,383	1	4.75e-05	54.02	7,544,089	0.44	6.49e-05
200	70.74	12,983,383	1	4.73e-05	51.07	7,699,909	0.43	6.37e-05
								Elements
Avgs	R_{avg}	NZ_{avg}	Rat		R_{avg}	NZ_{avg}	Rat	ACE/Unif
	60.31	1.3e7	1		47.32	8.07e6	0.45	0.59

Table 6.3: Work comparison between uniform and adaptive refinement of the Tearing Mode problem.

The results in table 6.3, show that the adaptive scheme does less work on average than the uniform scheme. Using the adaptive meshes, we achieve the same functional value as with the uniform meshes, but with fewer degrees of freedom. For the tearing mode problem, it appears that a large portion of the region needs to be refined to resolve the problem appropriately. However, there are some areas of steep gradients and we do see more refinement in these regions. ACE requires 0.45 times the amount of work that is required of uniform refinement. Just comparing the number of elements on the finest grid at each timestep shows that ACE uses about 59% of the elements that uniform refinement uses. Again, we observe a deterioration in AMG convergence on the ACE grids. This is most likely due to the fact that the performance of the solver is not ideal on these adaptively

refined meshes. One way to improve the results further would be to develop better AMG-based solvers that accounts more effectively for locally refined grids. Yet, even without improved solvers, using ACE reduced the work needed to resolve the tearing mode problem by more than half, making the algorithm even more viable for use in resolving MHD instabilities.

In the next problem, the island coalescence instability, we again see the effectiveness of the adaptive scheme and the nested iteration algorithm.

6.2.3 Test Problem: Island Coalescence

The second test problem simulates an island coalescence in the current density arising from perturbations in an initial current density sheet. Again, a current density sheet in the toroidal direction of the tokamak is perturbed, resulting in an instability that causes a reconnection in the magnetic field lines and merging of two islands in the current density field. This produces a sharp peak in current density where the magnetic field lines reconnect. This region is known as the reconnection zone and the point at which the magnetic field lines break is known as the $\mathcal{X}$ point. See [3, 33, 39] for more details. For the following simulations, we define

$$\begin{aligned} \Omega &= [-1,1] \times [-1,1], \\ R_e &= S_L = 50,001. \end{aligned}$$

The initial conditions at equilibrium are

$$\mathbf{B}_0(x,y) = \frac{1}{\cosh(2\pi y) + k\cos(2\pi x)} \begin{pmatrix} \sinh(2\pi y) \\ k\sin(2\pi x) \end{pmatrix}, \tag{6.15}$$

$$\mathbf{u}_0(x,y) = \mathbf{0}, \tag{6.16}$$

$$\omega_0(x,y) = 0, \tag{6.17}$$

$$j_{30}(x,y) = \nabla \times \mathbf{B}_0 = \frac{2\pi(k^2-1)}{(\cosh(2\pi y) + 0.2\cos(2\pi x))^2}, \tag{6.18}$$

$$p_0(x,y) = \frac{(1-k^2)}{2}(1 + \frac{1}{(\cosh(2\pi y) + 0.2\cos(2\pi x))^2}). \tag{6.19}$$

where $k = 0.2$ and $\epsilon = -0.01$. These initial conditions are perturbed away from equilibrium as follows:

$$\delta\mathbf{B}_0(x,y) = \begin{pmatrix} -\epsilon\frac{1}{\pi}\cos(\pi x)\sin(\pi\frac{y}{2}) \\ \frac{1}{2}\epsilon\frac{1}{\pi}\cos(\pi\frac{y}{2})\sin(\pi x) \\ 0 \end{pmatrix}, \tag{6.20}$$

$$\delta j_{30}(x,y) = \epsilon\cos(\pi\frac{y}{2})\cos(\pi x). \tag{6.21}$$

The boundary conditions are periodic in x and Dirichlet for the current density and vorticity on the top and bottom of the domain. We also have $\mathbf{n}\cdot\mathbf{u}$ and $\mathbf{n}\cdot\mathbf{B}$ known on the top and bottom. Again, the FOSLS formulation, (4.21)-(4.28), is H^1 elliptic.

6.2.4 Results

The problem was run to time $8\tau_A$ with a timestep of $0.1\tau_A$ using a BDF-2 scheme. At this point, the islands have coalesced and the large peak in current density has occurred at the reconnection point. Using uniform refinement and

quadratic elements, the nested iteration was performed up to a 128×128 grid. This leads to $462,343$ degrees of freedom. The linear stopping parameter was

$$F_s^2 \leq 1.001 F_*^2.$$

The nonlinear stopping parameter was

$$\frac{|F_s^2 - \mathcal{F}_s^2|}{\mathcal{F}_s^2} \leq 0.1.$$

As with the Tearing Mode instability, on the finest grid and for all time steps, only 1 Newton step was needed for the Newton iterations to converge and a small number of V-cycles were needed for AMG to resolve the linear system. The results with uniform refinement at time $8\tau_A$ are seen in figure 6.6. This shows the sharp peak in the current density as well as the coalescence of the two islands.

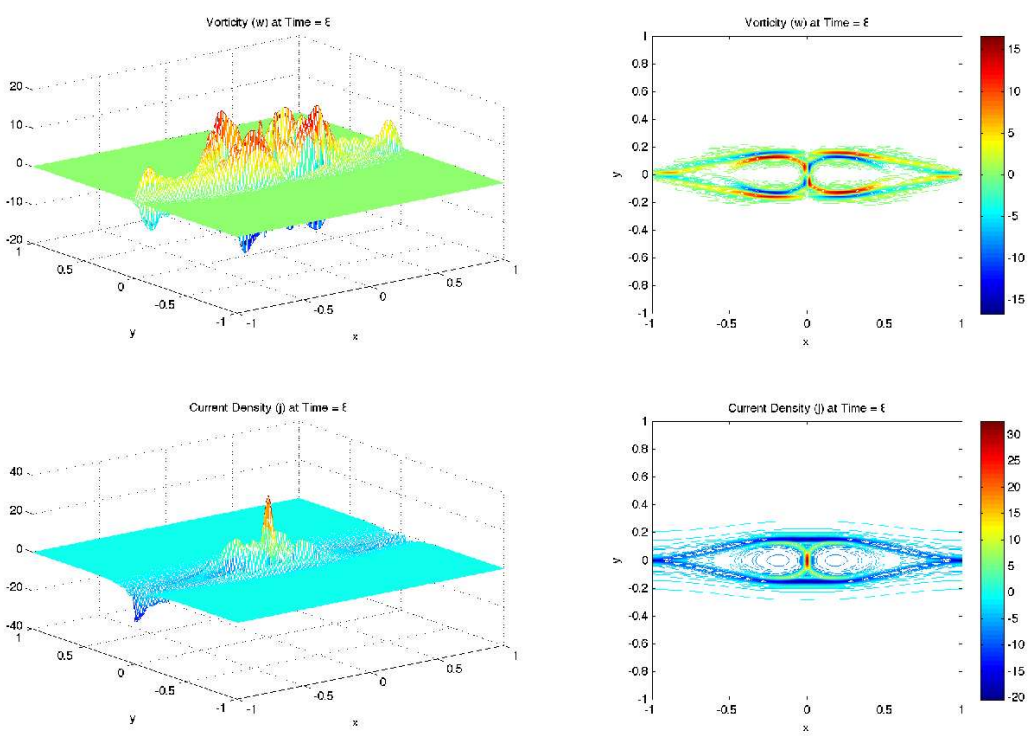

Figure 6.6: Numerical solution after $8\tau_A$. $h = 1/64$ $p = 2$ using uniform refinement. $S_L = R_e = 50,001$. Top: vorticity. Bottom: current density.

Again, using ACE, we see that the grids evolve over time to refine in areas with steeper gradients. In this problem, as time progresses, a steep gradient occurs at the reconnection point. This is seen in the bottom graph in figure 6.7. We expect, then, that most of the refinement occurs at this point. As shown in figure 6.7, by later timesteps, this is indeed the case. Table 6.4 shows that using adaptive refinement greatly reduces the amount of work needed.

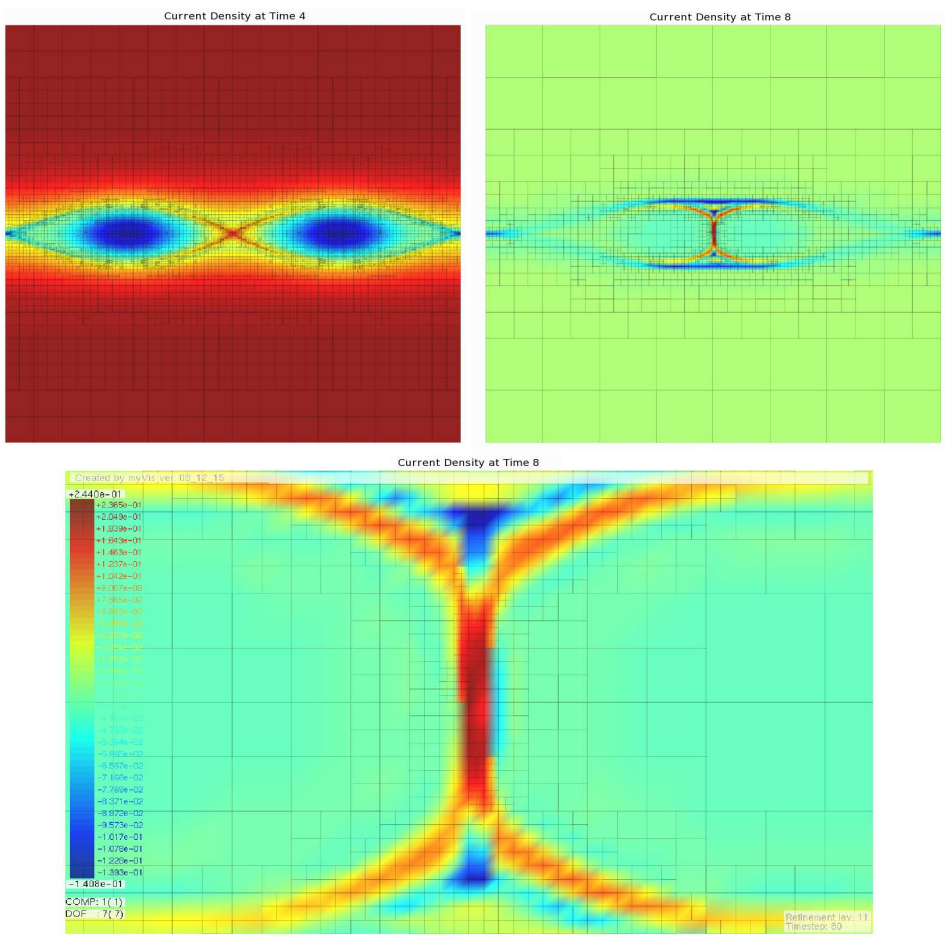

Figure 6.7: Numerical solution using 10 levels of adaptive refinement. $S_L = R_e = 50,001$. Top Left: Current Density at Time $4\tau_A$. Top Right: Current Density at Time $8\tau_A$. Bottom: Zoomed in plot of current density peak at Time $8\tau_A$.

		Uniform	8	Levels		ACE	10	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
1	40.76	51,781,975	1	6.08e-05	35.41	35,191,751	0.59	5.8e-05
2	40.98	51,781,975	1	5.17e-05	54.72	11,936,547	0.31	6.31e-05
3	42.00	51,781,975	1	4.13e-05	63.28	14,008,757	0.41	4.84e-05
4	42.10	51,781,975	1	3.29e-05	55.00	9,483,509	0.24	5.22e-05
5	42.05	51,781,975	1	2.65e-05	68.15	11,383,533	0.36	3.57e-05
6	45.31	51,781,975	1	2.16e-05	62.41	12,496,911	0.33	2.76e-05
7	54.88	51,781,975	1	1.79e-05	60.85	14,875,665	0.32	2.11e-05
8	54.85	51,781,975	1	1.52e-05	54.76	24,464,867	0.47	1.20e-05
9	52.53	51,781,975	1	1.32e-05	61.82	14,977,193	0.34	1.54e-05
10	55.78	51,781,975	1	1.17e-05	67.18	9,112,775	0.21	3.04e-05
11	55.15	51,781,975	1	1.06e-05	69.31	9,817,493	0.23	2.55e-05
12	55.17	51,781,975	1	9.77e-06	67.67	10,135,895	0.24	2.25e-05
13	55.16	51,781,975	1	9.16e-06	66.29	14,334,705	0.33	1.16e-05
14	55.21	51,781,975	1	8.72e-06	73.76	15,265,313	0.39	9.93e-06
15	55.20	51,781,975	1	8.41e-06	66.79	5,606,923	0.13	8.02e-05
16	55.04	51,781,975	1	8.20e-06	68.38	10,616,095	0.25	1.83e-05
17	54.24	51,781,975	1	8.08e-06	70.12	16,476,593	0.41	8.09e-06
18	55.26	51,781,975	1	8.02e-06	81.68	17,736,187	0.51	6.72e-06
19	61.83	51,781,975	1	8.01e-06	66.62	9,788,093	0.20	2.33e-05
20	61.83	51,781,975	1	8.05e-06	73.66	11,472,027	0.26	1.53e-05
21	61.77	51,781,975	1	8.15e-06	75.76	11,846,681	0.28	1.43e-05
22	64.87	51,781,975	1	8.31e-06	66.29	8,593,963	0.17	3.24e-05
23	64.99	51,781,975	1	8.54e-06	73.32	22,924,797	0.50	5.16e-06
24	64.86	51,781,975	1	8.86e-06	77.52	10,754,863	0.25	1.91e-05
25	65.00	51,781,975	1	9.28e-06	77.63	10,560,333	0.24	1.96e-05

		Uniform	8	Levels		ACE	10	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
26	64.98	51,781,975	1	9.85e-06	77.38	10,727,619	0.25	1.96e-05
27	65.04	51,781,975	1	1.06e-05	78.68	10,030,153	0.23	2.33e-05
28	65.07	51,781,975	1	1.15e-05	80.52	9,664,613	0.23	2.64e-05
29	64.28	51,781,975	1	1.27e-05	84.11	9,524,571	0.24	3.03e-05
30	64.29	51,781,975	1	1.41e-05	86.20	24,150,091	0.63	6.46e-06
31	67.67	51,781,975	1	1.59e-05	93.72	14,002,191	0.37	1.96e-05
32	67.74	51,781,975	1	1.79e-05	87.45	14,528,843	0.36	1.98e-05
33	67.68	51,781,975	1	2.04e-05	99.47	23,946,251	0.68	8.49e-06
34	72.81	51,781,975	1	2.34e-05	100.20	24,023,867	0.64	9.20e-06
35	75.82	51,781,975	1	2.70e-05	96.09	12,428,507	0.30	4.10e-05
36	75.82	51,781,975	1	3.12e-05	109.39	21,882,077	0.61	1.34e-05
37	75.93	51,781,975	1	3.62e-05	66.82	9,989,973	0.17	8.38e-05
38	75.98	51,781,975	1	4.21e-05	105.37	21,740,957	0.58	1.50e-05
39	74.69	51,781,975	1	4.90e-05	127.43	32,788,693	1.08	6.99e-06
40	74.68	51,781,975	1	5.72e-05	130.87	32,285,855	1.09	1.01e-05
41	74.62	51,781,975	1	6.67e-05	140.40	30,855,055	1.12	1.37e-05
42	71.39	51,781,975	1	7.79e-05	102.04	16,703,561	0.46	6.39e-05
43	71.23	51,781,975	1	9.11e-05	142.00	28,990,213	1.11	1.87e-05
44	71.22	51,781,975	1	1.06e-04	103.58	13,437,711	0.38	1.16e-04
45	71.29	51,781,975	1	1.24e-04	100.78	15,636,635	0.43	9.02e-05
46	74.46	51,781,975	1	1.44e-04	101.75	15,848,413	0.42	8.50e-05
47	72.20	51,781,975	1	1.68e-04	103.10	14,408,695	0.40	1.21e-04
48	74.24	51,781,975	1	1.95e-04	149.00	25,188,499	0.98	3.64e-05
49	77.09	51,781,975	1	2.26e-04	85.71	9,211,657	0.20	5.60e-04
50	72.10	51,781,975	1	2.62e-04	111.27	16,253,545	0.48	1.01e-04

		Uniform	8	Levels		ACE	10	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
51	77.08	51,781,975	1	3.03e-04	107.64	14,490,623	0.39	1.60e-04
52	77.24	51,781,975	1	3.51e-04	108.38	13,865,481	0.38	1.94e-04
53	82.05	51,781,975	1	4.07e-04	112.33	14,189,763	0.39	2.14e-04
54	82.28	51,781,975	1	4.70e-04	162.78	20,690,495	0.79	1.08e-04
55	82.55	51,781,975	1	5.45e-04	104.91	13,090,399	0.32	5.71e-04
56	80.53	51,781,975	1	6.33e-04	168.33	19,288,801	0.78	1.76e-04
57	78.54	51,781,975	1	7.37e-04	110.32	13,119,995	0.36	6.82e-04
58	79.51	51,781,975	1	8.63e-04	100.69	7,611,415	0.19	1.79e-03
59	77.10	51,781,975	1	1.10e-03	152.42	19,747,441	0.75	1.80e-03
60	92.81	51,781,975	1	1.19e-03	170.26	19,386,311	0.69	2.57e-04
61	91.60	51,781,975	1	1.42e-03	148.53	10,747,023	0.34	1.31e-03
62	91.60	51,781,975	1	1.71e-03	206.83	18,301,059	0.80	4.42e-04
63	92.47	51,781,975	1	2.10e-03	171.51	11,483,591	0.41	1.76e-03
64	93.15	51,781,975	1	2.61e-03	164.49	12,551,693	0.43	1.63e-03
65	90.24	51,781,975	1	3.27e-03	167.66	8,173,935	0.29	4.49e-03
66	92.54	51,781,975	1	4.14e-03	175.61	7,243,621	0.27	4.86e-03
67	92.86	51,781,975	1	5.26e-03	232.33	8,290,947	0.40	3.46e-03
68	93.15	51,781,975	1	6.64e-03	149.53	5,524,015	0.17	1.30e-02
69	95.47	51,781,975	1	8.32e-03	172.68	5,751,669	0.20	8.92e-03
70	105.81	51,781,975	1	1.10e-02	310.07	10,168,431	0.59	3.34e-03
71	105.42	51,781,975	1	1.22e-02	416.81	11,620,791	0.89	4.50e-03
72	105.71	51,781,975	1	1.43e-02	289.82	6,469,715	0.34	2.65e-02
73	109.36	51,781,975	1	1.63e-02	390.43	11,971,729	0.83	5.46e-03
74	107.40	51,781,975	1	1.80e-02	329.67	7,983,325	0.47	2.29e-02
75	116.82	51,781,975	1	1.93e-02	235.11	6,855,443	0.27	2.83e-02

		Uniform	8	Levels		ACE	10	Levels
Time	R	NZ	Rat	NF	R	NZ	Rat	NF
76	116.61	51,781,975	1	2.02e-02	292.78	7,683,739	0.37	1.90e-02
77	117.05	51,781,975	1	2.03e-02	251.36	6,114,857	0.25	3.49e-02
78	116.23	51,781,975	1	1.99e-02	219.54	5,289,599	0.19	4.60e-02
79	106.72	51,781,975	1	1.92e-02	343.20	14,824,607	0.92	3.82e-03
80	106.66	51,781,975	1	1.84e-02	298.36	8,135,029	0.44	2.37e-02
								Elements
Avgs	R_{avg}	NZ_{avg}	Rat		R_{avg}	NZ_{avg}	Rat	ACE/Unif
	74.95	$5.18 \cdot 10^7$	1		131.08	$1.43 \cdot 10^7$	0.45	0.26

Table 6.4: Work comparison of island coalescence problem between uniform and adaptive refinement.

Table 6.4 shows that ACE requires 45% of the work that uniform refinement requires. The physics is more localized in this problem, especially by time $8\tau_A$ and, thus, the refinement is more localized. Again, we are not getting the best possible performance as ACE uses only 26% of the elements that uniform refinement does. This is due to the fact that AMG has a larger convergence factor on locally refined grids than on a uniform grid. However, from these results, we can see the potential for greater accuracy-per-computational cost on these MHD systems than what we have been getting, if a better multigrid solver is used.

In addition to showing that our algorithm works well numerically on the island problem, we also confirm that we are capturing the physics of the problem. To do this, we measure the growth and reconnection rates of the instabilities. These are described in [3, 16, 33, 39].

First, the growth rate, which measures how the perturbation from the equilibrium state in all the primitive variables grows over time, is computed as follows:

$$\Gamma(t) = log||\mathcal{U} - \mathcal{U}_0||_0. \tag{6.22}$$

Here, $\mathcal{U}$ is any of the dependent variables and $\mathcal{U}_0$ is the unperturbed initial equilibrium state of that variable. We expect the perturbations for the island instability to grow exponentially with time up until the reconnection occurs. This is seen as a linear rate for $\Gamma(t)$. After this, the growth rate should remain steady with order one. This is shown in figure 6.8 and 6.9 for Reynolds and Lundquist numbers of 50,001 and 5,000, respectively. Thus, the instabilities are propagating at the expected rate, giving us more confirmation that our timestepping is stable and that we are capturing the appropriate physics.

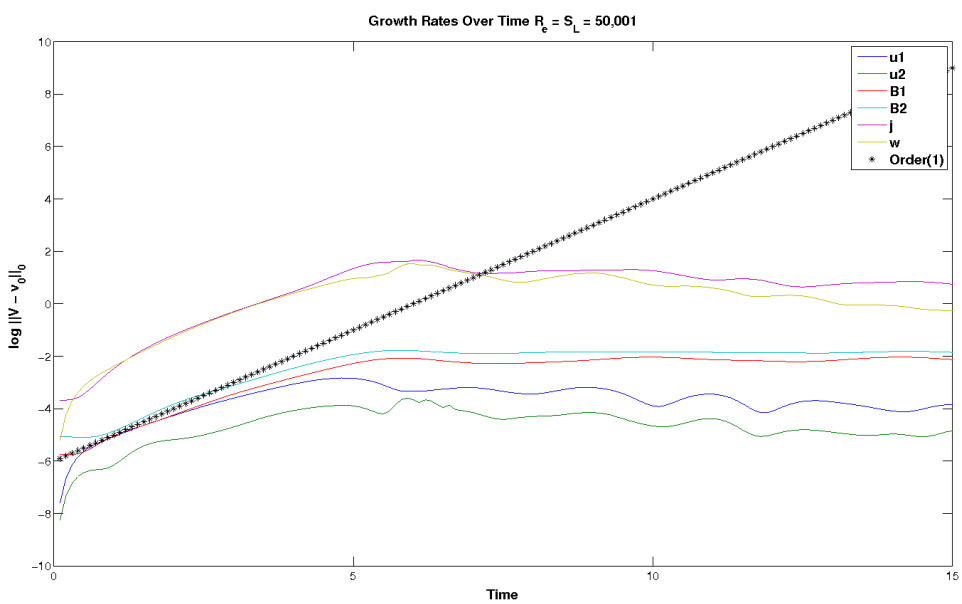

Figure 6.8: Growth rate of all variables over time. $R_e = S_L = 50,001$.

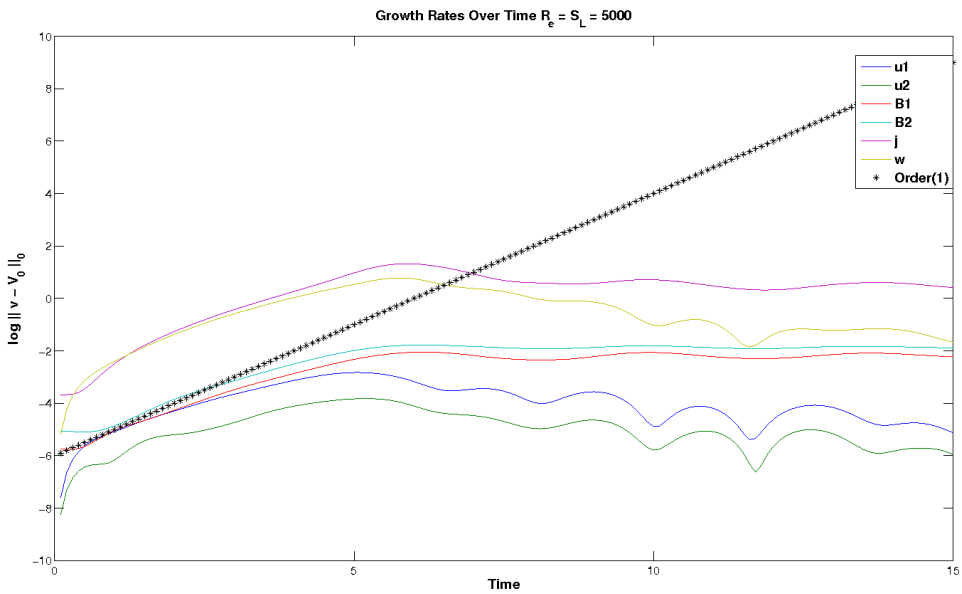

Figure 6.9: Growth rate of all variables over time. $R_e = S_L = 5,000$.

Another measure of the quality of the approximated solution is the reconnection rate. This describes how the magnetic field lines break and reconnect over time. It is measured by checking how the current density, j, changes with time relative to its equilibrium state, proportionally to the resistivity of the system, specifically at the point where the reconnection occurs. For the island coalescence problem, this is known as the $\mathcal{X}$-point, or where the two islands collide in the center. We have

$$E_z|_{\mathcal{X}} = \frac{\partial \Psi}{\partial t}|_{\mathcal{X}} = \frac{1}{S_L}(j(t) - j(0))|_{\mathcal{X}}, \tag{6.23}$$

where Ψ is the poloidal flux function. As in [16, 33], we expect that the reconnection rate increases over time until the reconnection occurs, when it reaches its peak magnitude. It then decays back to zero as the islands merge into one. For higher Lundquist numbers (i.e. lower resistivities), this decay involves some

oscillations known as "sloshing" [33]. We also observe this behavior as seen in figure 6.11. For a Lundquist number of $5,000$, the reconnection rate peaks and then smoothly decays to zero. For Lundquist numbers of $10,000$ and $50,001$, we start to see the sloshing after the reconnection has occurred. In addition, it is predicted that the peak reconnection rate decreases with Lundquist number with order $O(\frac{1}{\sqrt{S_L}})$. Our method captures this relation, as seen in figure 6.10. The gradient of the current density peak also becomes sharper with higher Lundquist numbers, causing it to be harder to resolved. A comparison of the gradients for Lundquist numbers of $10,000$ and $100,000$ is shown in figure 6.12. The ratio of the width of the current density peak, denoted by δ, for $S_L = 100,000$ to the width for $S_L = 10,000$ is about $\frac{1}{3}$. This corresponds to the width being proportional to $\frac{1}{\sqrt{S_L}}$, as predicted by experiment [33].

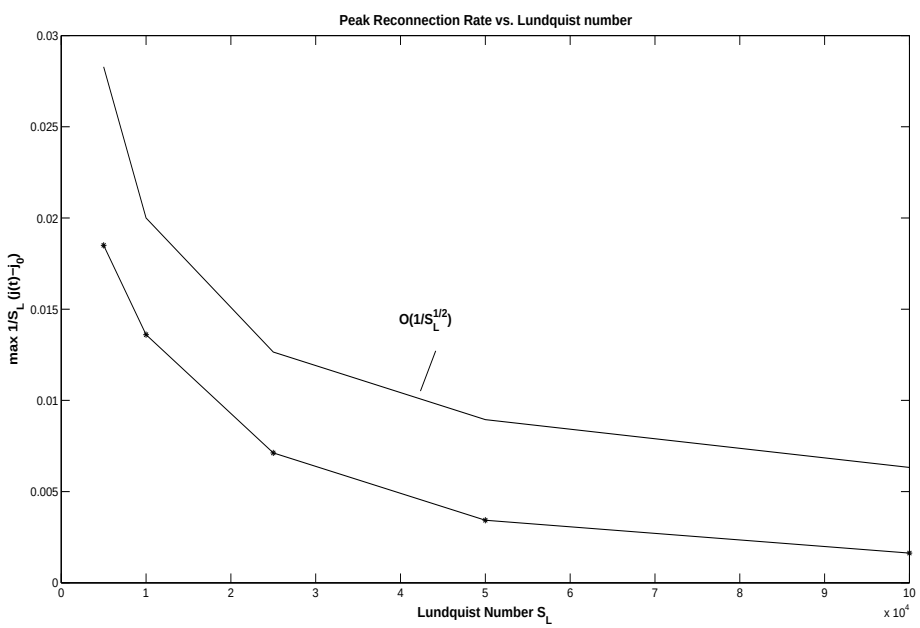

Figure 6.10: Peak reconnection rate vs. Lundquist number, S_L.

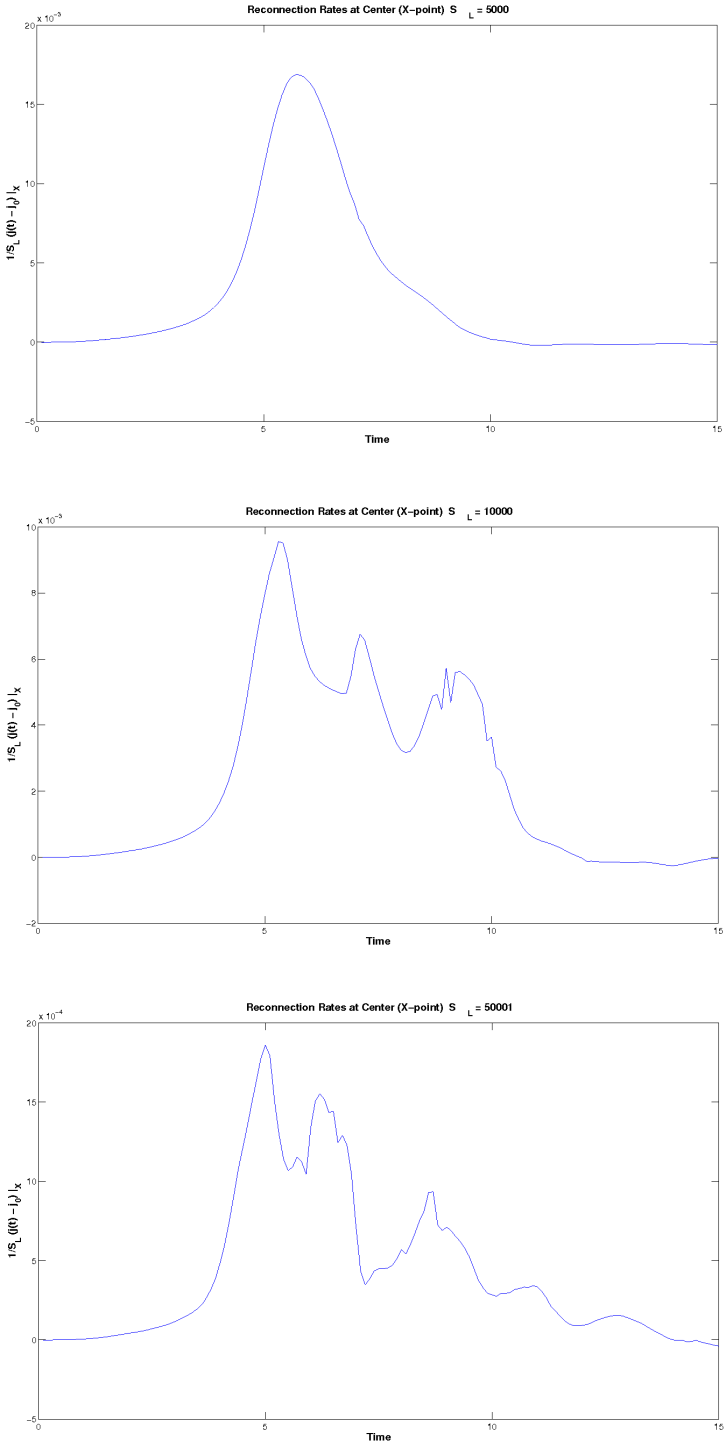

Figure 6.11: Reconnection Rate vs Time. Top: $S_L = 5,000$. Middle: $S_L = 10,000$. Bottom: $S_L = 50,001$.

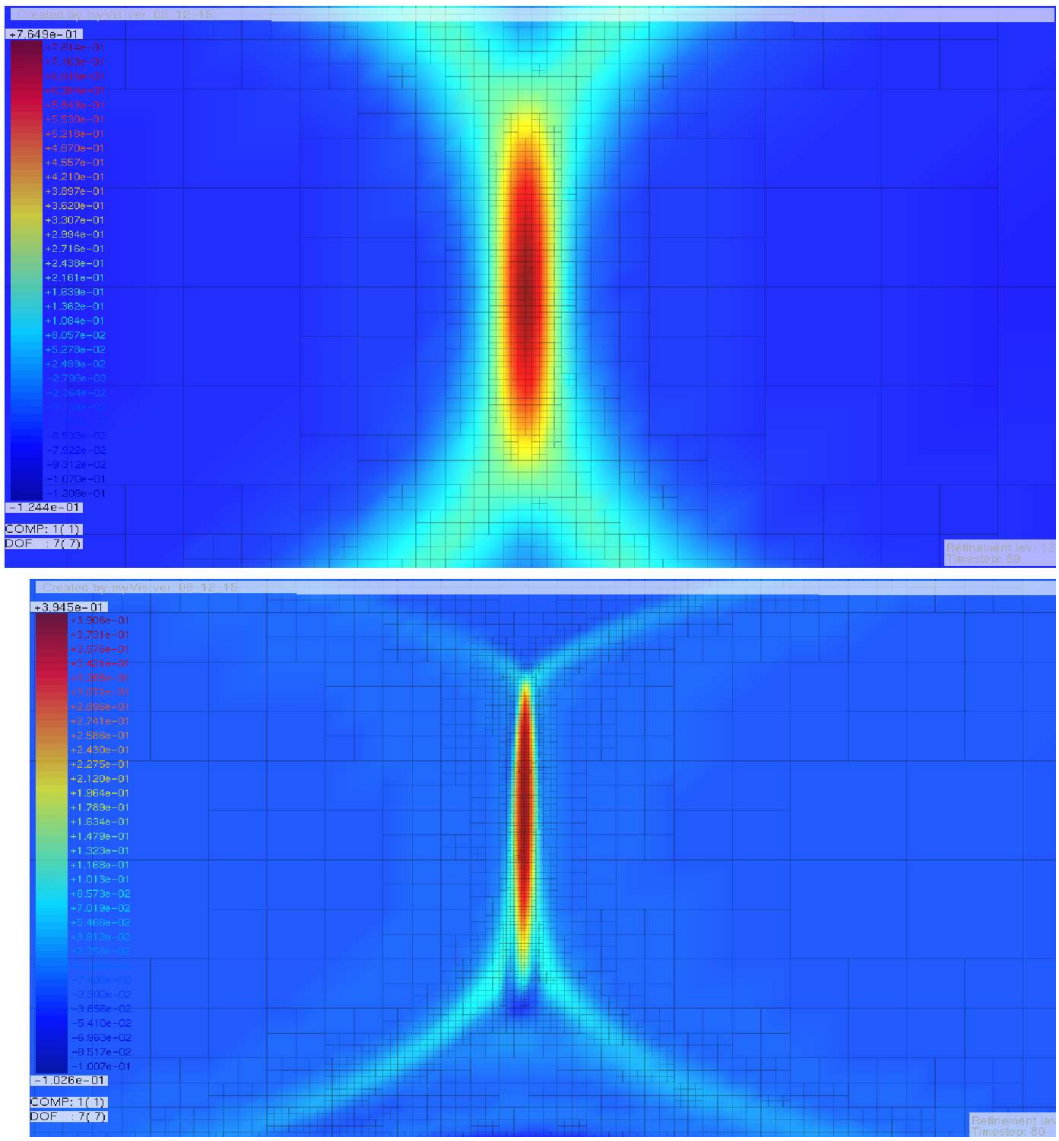

Figure 6.12: Current density near $\mathcal{X}$ point, where a sharp peak occurs during reconnection. (a) $R_e = S_L = 10,000$, $\delta = 0.0117$. (b) $R_e = S_L = 100,000$, $\delta = 0.0039$.

The results here indicate the solutions are being resolved correctly. Our approach faithfully captures the physics at the cost of approximately 40 work units per time step.

Chapter 7

Discussion

7.1 Concluding Remarks

We showed that the FOSLS finite element method, along with nested iteration, is highly effective for the complicated, current-vorticity form of resistive MHD. Real world MHD applications are solved very efficiently when the focus is on accuracy-per-computational-cost. The crux of the above methodology is that most of the difficult computations, such as the linearizations, are done on coarser grids where computing is less expensive. Moreover, the solution on a coarse grid provides a good starting guess for the solution on the next finer grid. As a result, when the desired accuracy is reached, an approximation to the solution is obtained in only a few work units. The use of FOSLS greatly aids this process. Its sharp, a posteriori error estimate allows parameters to be computed that are used to estimate the current accuracy-per-computational-cost. From this, judgements as to what further computation is necessary are made. For example, this information is used to determine whether the method performs another V-cycle, stops iterating and relinearizes the problem, or quits on the current grid altogether and moves to a finer level. Such decision making is important for efficiently solving complicated systems, such as the MHD equations and also facilitates an efficient local adaptive refinement process.

In the process of developing this algorithm, each stage of it is analyzed with

the goal of achieving the most efficiency. It is first proved that the incompressible, resistive MHD equations can be reformulated in a way that gives a coercive and continuous FOSLS formulation, thus giving a well-posed discrete problem. In addition, a rescaling of the MHD equations is done so that AMG worked well on the resulting linear systems, while still capturing the physics of the problem. Next, the solution algorithm is considered. At each level and each Newton step a linearized system is solved. Therefore, not only do we need to make sure our linear solvers are efficient, but we must go back and determine how well the nonlinear system is being resolved as well. We show that, by doing this in a nested iteration approach, we can do all these checks and tests on coarser grids.

Timestepping is also analyzed, particularly on how it affects the solution. Without a stable timestepping scheme, it is possible to lose accuracy over time and, thus, miss much of the physics. This thesis shows that the BDF-k schemes are stable for $k < 7$. In our numerical tests, we use BDF-1 (Backward Euler) and BDF-2, which are $O(\Delta t)$ and $O(\Delta t^2)$, respectively.

Finally, the fact that moving up through the grids can be done with an efficiency-based adaptive mesh refinement algorithm is considered. This allows for the meshes to be locally refined at each timestep and, therefore, follow the physics of the solution as it evolves over time. Thus, by considering each part of the algorithm, we are able to solve a time-dependent nonlinear system of equations in only a handful of work units, or equivalently, a handful of fine-grid relaxations.

7.2 Future Work

Several aspects still need to be studied. First, if previous data are used to determine when to move to a finer grid or to keep iterating, the previous data must be solved "well enough" to get good estimates. For instance, is it necessary to solve the problem on the previous time step to a greater accuracy than otherwise

required in order to obtain accurate parameters? For this thesis, all the parameters were chosen based on average estimates and were fixed for the entire run. In the future, we hope to develop heuristics that allow these parameters to be adjusted dynamically. At some point, more complicated physics could occur and, then, the algorithm needs to determine those parameters more precisely. At other times, the solution could be nice and smooth and, therefore, the problem does not need to be solved as rigorously. Determining when that occurs could greatly reduce the amount of work being done to get to the desired accuracy.

Secondly, the linear system solvers are what dictate the overall efficiency of the NI-Newton-FOSLS method. For this thesis, the algebraic systems are solved with a classical algebraic multigrid method. Deteriorations in the algebraic convergence for increased timestep size as well as Reynolds and Lundquist numbers are observed. There are a number of ways in which one could improve the current AMG algorithm. One might develop an improved AMG for the above type of systems of PDEs. This might involve the use of newly developed adaptive multigrid algorithms described more in [10, 11, 35]. In the test problems addressed in this thesis, tensor product grids are used, with local refinement. Therefore, one might employ a geometric multigrid solver instead that takes into account the hierarchy of grids used, or one that is based on block structured grids. Specifically, using the same multigrid method to solve on a uniform mesh as for an adaptively refined mesh might not be ideal. One would prefer a solver that takes into account the unstructured nature of the grid.

In addition, there are many aspects of the adaptive refinement algorithm that can be improved. In the above tests, the ACE algorithm requires more levels than uniform refinement to reach the same accuracy because it does not refine as aggressively as uniform refinement. Currently, we are examining modifications to ACE that dictate the number of unknowns on the refined grid. This number is

fixed and, then, the ACE algorithm chooses where to put these new nodes. This is accomplished by allowing ACE to refine an element more than once at each step. This approach allows for fewer refinement steps and, hence, fewer computations overall.

Finally, there are many other MHD problems to be tested, as well as other time-dependent problems in fluid dynamics that have large nonlinearities. Doing most of the hard work on the coarser grids allows us to solve these problems more efficiently. Using a first-order system least squares formulation, we were able to resolve the above MHD physics and we believe that with a careful formulation it can be used for many other time-dependent nonlinear systems. This, with the addition of a parallel implementation, could allow us to tackle even more complicated problems such as Extended or Hall MHD, as well as other complex fluid problems.

Bibliography

[1] T.D. Arber and R.G.L. Vann. A Critical Comparison of Eulerian-Grid-Based Vlasov Solvers. J. Comp. Phys, 180:339–357, April 2002.

[2] Kendall E. Atkinson. An Introduction to Numerical Analysis. John Wiley and Sons, 2nd edition, 1989.

[3] G. Bateman. MHD Instabilities. The MIT Press, 1978.

[4] P. Bochev, Z. Cai, T. Manteuffel, and S. McCormick. Analysis of Velocity-Flux First-Order System Least-Squares Principles for the Navier-Stokes Equations: Part I. SIAM J. Numer. Anal, 35:990–1009, 1998.

[5] P. Bochev, Z. Cai, T. Manteuffel, and S. McCormick. Analysis of Velocity-Flux First-Order System Least-Squares Principles for the Navier-Stokes Equations: Part II. SIAM J. Numer. Anal, 36:1125–1144, 1999.

[6] P. Bochev and M. Gunzburger. Analysis of Least-Squares Finite Element Mehtods for the Stokes Equations. Mathematics of Computation, 63(208):479–506, October 1994.

[7] J. H. Bramble, T. V. Kolev, and J.E. Pasciak. A Least-Squares Approximation Method for the Time-Harmonic Maxwell Equations. J. Numer. Math., 13:237–263, 2005.

[8] A. Brandt. Algebraic Multigrid Theory: The Symmetric Case. Appl. Math. Comput., 19(1-4):23–56, 1986.

[9] S. C. Brenner and L. R. Scott. Mathematical Theory of Finite Element Methods. Springer, 2nd edition edition, 2002.

[10] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation (αsa) multigrid. SIAM Review (SIGEST), 47:317–346, 2005.

[11] M. Brezina, R. Falgout, S. Maclachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive algebraic multigrid. SIAM J. on Sci. Comp. (SISC), 27:1261–1286, 2006.

[12] W. L. Briggs, V. E. Henson, and S. F. McCormick. A Multigrid Tutorial. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000.

[13] P. Brown, P. Vassilevski, and C. Woodward. On Mesh-Independent Convergence of an Inexact Newton-Multigrid Algorithm. SIAM Journal on Scientific Computing, 25:570–590, 2003.

[14] Z. Cai, R. Lazarov, T. Manteuffel, and S. McCormick. First-Order System Least Squares for Second-Order Partial Differential Equations. SIAM J. Numer. Anal, 31:1785–1799, 1994.

[15] Z. Cai, T. Manteuffel, and S. McCormick. First-Order System Least Squares for Second-Order Partial Differential Equations. II,. SIAM J. Numer. Anal, 34:425–454, 1997.

[16] L. Chacon, D. A. Knoll, and J. M. Finn. An Implicit, Nonlinear Reduced Resistive MHD Solver. J. of Computational Physics, 178:15–36, 2002.

[17] L. Chacon, D. A. Knoll, and J. M. Finn. Nonlinear Study of the Curvature-Driven Parallel Velocity Shear-Tearing Instability. Physics of Plasmas, 9:1164–1176, 2002.

[18] F. F. Chen. Introduction to Plasma Physics. Plenum, New York, 1974.

[19] A. Codd. Elasticity-Fluid Coupled Systems and Elliptic Grid Generation (EGG) Based on First-Order System Least Squares (FOSLS). PhD thesis, University of Colorado at Boulder, 2001.

[20] A. Codd, T. Manteuffel, and S. McCormick. Multilevel First-Order System Least Squares for Nonlinear Elliptic Partial Differential Equations. SIAM J. Numer. Anal, 41:2197–2209, 2003.

[21] M. Costabel. A Coercive Bilinear Form for Maxwell's Equations. J. Math. Anal. and Appl., 157:527–541, 1991.

[22] H. DeSterck, T. Manteuffel, S. McCormick, J. Nolting, J. Ruge, and L. Tang. Efficiency-based h- and hp-refinement strategies for finite element methods. J. Num. Lin. Alg. Appl., 15:249–270, 2008.

[23] F. Filbet and E. Sonnendrücker. Comparison of Eulerian Vlasov Solvers. Computer Physics Communications, 150:247–266, June 2003.

[24] F. Filbet, E. Sonnendrücker, and P. Bertrand. Conservative Numerical Schemes for the Vlasov Equation. J. Comp. Phys, 172:166–187, April 2001.

[25] B. Fornberg. On the Instability of Leap-Frog and Crank-Nicolson Approximations of a Nonlinear Partial Differential Equation. Mathematics of Computation, 27(121):13, January 1973.

[26] V. Girault and P.A. Raviart. Finite Element Approximation of the Navier-Stokes Equations. Springer-Verlag, 1979.

[27] M. Hestenes and E. Steifel. Methods of Conjugate Gradients for Solving Linear Systems. J. Research NBS, 49:409–436, 1952.

[28] J.J. Heys, E. Lee, T. Manteuffel, and S. McCormick. An Alternative Least-Squares Formulation of the Navier-Stokes Equations with Improved Mass Conservation. J. Comp. Phys, 226(1):994–1006, September 2007.

[29] J. K. Hunter and B. Nachtergaele. Applied Analysis. World Scientific, 2001.

[30] B. Jiang. The Least-Squares Finite Element Method. Theory and Applications in Computational Fluid Dynamics and Electromagnetics. Springer-Verlag, 1998.

[31] A. N. Kaufman. Plasma Physics in Theory and Application. McGraw-Hill, New York, 1966.

[32] L.R. Petzold K.E. Brenan, S.L. Campbell. Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[33] D. A. Knoll and L. Chacon. Coalescence of Magnetic Islands, Sloshing, and the Pressure Problem. Physics of Plasmas, 13(1), 2006.

[34] N. A. Krall and A. W. Trivelpiece. Principles of Plasma Physics. McGraw-Hill, New York, 1973.

[35] S. MacLachlan. Improving Robustness in Multiscale Methods. PhD thesis, University of Colorado at Boulder, 2004.

[36] D. R. Nicholson. Introduction to Plasma Theory. John Wiley and Sons, New York, 1983.

[37] J. Nolting. Efficiency-based Local Adaptive Refinement for FOSLS Finite Elements. PhD thesis, University of Colorado at Boulder, 2008.

[38] C.W. Oosterlee, A. Schuller, and U. Trottenberg. Multigrid. Academic Press, 2000.

[39] B. Philip, L. Chacon, and M. Pernice. Implicit Adaptive Mesh Refinement for 2D Reduced Resistive Magnetohydrodynamics. J. Comp. Phys, 227(20):8855–8874, October 2008.

[40] N. A. Phillips. An Example of Non-Linear Computational Instability, pages 501–504. *The Atmosphere and the Sea in Motion* Rockefeller Institute, New York, 1959.

[41] R.D.Richtmyer and K.W.Morton. Difference Methods for Initial-Value Problems. Interscience, New York, 2 edition, 1967.

[42] R. D. Richtmyer. A Survey of Difference Methods for Non-Steady Fluid Dynamics. Technical note 63-2, National Center for Atmospheric Research, Boulder, CO, 1962.

[43] O. Roehrle. Multilevel First-Order System Least Squares for Quasilinear Elliptic Partial Differential Equations. PhD thesis, University of Colorado at Boulder, 2004.

[44] J. Ruge. Fospack users manual, version 1.0. unpublished, 2000.

[45] J. Ruge and K. Stüben. Algebraic Multigrid (AMG). Multigrid Methods (McCormick, S.F., ed), 1986.

[46] Y. Saad. Iterative Methods for Sparse Linear Systems. SIAM books, 2003.

[47] H.R. Strauss. Nonlinear, Three-Dimensional Magnetohydrodynamics of Noncircular Tokamaks. Physics of Fluids, 19:134–140, 1976.

[48] P. Ullrich. Dynamics and Properties of the Magnetohydrodynamics Equations. unpublished, 2005.

[49] C. Westphal. First-Order System Least Squares for Geometrically-Nonlinear Elasticity in Nonsmooth Domains. PhD thesis, University of Colorado at Boulder, 2004.

[50] X. Zhang. Multilevel schwarz methods. Numer. Math, 63:521–539, 1992.