First-Order Systems Least-Squares Finite Element Methods and Nested Iteration for Electromagnetic Two-Fluid Kinetic-Based Plasma Models

by

Christopher A. Leibs

B.S., Rose-Hulman Institute of Technology, 2009

M.S., University of Colorado, 2012

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

2014

This thesis entitled:
First-Order Systems Least-Squares Finite Element Methods and Nested Iteration for Electromagnetic Two-Fluid Kinetic-Based Plasma Models
written by Christopher A. Leibs
has been approved for the Department of Applied Mathematics

_____

Prof. Thomas A. Manteuffel

_____

Prof. Stephen F. 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.

Leibs, Christopher A. (Ph.D., Applied Mathematics)

First-Order Systems Least-Squares Finite Element Methods and Nested Iteration for Electromagnetic Two-Fluid Kinetic-Based Plasma Models

Thesis directed by Prof. Thomas A. Manteuffel

Efforts are currently being directed towards a fully implicit, electromagnetic, JFNK-based solver, motivating the necessity of developing a fluid-based, electromagnetic, preconditioning strategy [15]. The two-fluid plasma (TFP) model is an ideal approximation to the kinetic Jacobian. The TFP model couples both an ion and an electron fluid with Maxwell's equations. The fluid equations consist of the conservation of momentum and number density. A Darwin approximation of Maxwell is used to eliminate light waves from the model in order to facilitate coupling to non-relativistic particle models. We analyze the TFP-Darwin system in the context of a stand-alone solver with consideration of preconditioning a kinetic-JFNK approach.

The TFP-Darwin system is addressed numerically by use of nested iteration (NI) and a First-Order Systems Least Squares (FOSLS) discretization. An important goal of NI is to produce an approximation that is within the basis of attraction for Newton's method on a relatively coarse mesh and, thus, on all subsequent meshes. After scaling and modification, the TFP-Darwin model yields a nonlinear, first-order system of equations whose Fréchet derivative is shown to be uniformly $\mathcal{H}^1$-elliptic in a neighborhood of the exact solution. $\mathcal{H}^1$ ellipticity yields optimal finite element performance and linear systems amenable to solution with Algebraic Multigrid (AMG). To efficiently focus computational resources, an adaptive mesh refinement scheme, based on the accuracy per computational cost, is leveraged. Numerical tests demonstrate the efficacy of the approach, yielding an approximate solution within discretization error in a relatively small number of computational work units.

Dedication

I dedicate this thesis to my father, for his unrelenting enthusiasm towards science, my mother, for her loving and unconditional support of everything I do, and finally, my brother, for being a constant source of motivation and inspiration.

Acknowledgements

There are countless (okay, technically countable) people to thank and acknowledge for their help in completing this thesis. First, I need to acknowledge Tom Manteuffel for his constant guidance and generosity. Tom has been an inspiring advisor and I look forward to continuing to work with him in the coming years. He has made my graduate school experience, dare I say, enjoyable. Special thanks to Steve McCormick for always asking the tough questions that led me to further discoveries, to John Ruge for his expertise in helping debug code and answering my endless questions about FOSPACK, and to Marian Brezina for providing unparalleled technical support and lively conversations. An extra special thanks goes to Luis Chacón for passing on so much knowledge about plasma physics, and tolerating our naive understanding of such a complex field. I especially couldn't have completed this thesis without the support of the Grandview gang. In particular, thanks go to Christian Ketelsen, for knowing the right time to drag me away from my computer, and to Toby Jones, for always lending an ear. I would also like to acknowledge all of the people who have acted as advisors to me in the past: Luis Chacón, Ben Bergen, Dana Knoll, Al McPherson, David Moulton, and Travis Austin. Thank you to my friends at Los Alamos, Will Taitano, Jeff Willert, and Josh Payne. Many thanks and acknowledgments go to my friends at CU, who have provided me with a constant source of support. The biggest acknowledgment goes to my wonderful girlfriend. Without her, I never could have accomplished this. Finally, none of this would have been possible without all of the generous sponsors of the research [1] .

[1] This work was partially performed under the auspices of the U.S. Department of Energy under grant number DE-FC02-03ER25574, Los Alamos National Laboratory under Subcontract 216508, and the National Science Foundation under grant number CBET-1249858.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

Plasma is a state of matter in which a large number of atoms have dissociated into their constituent ions and electrons. The availability of this free charge allows the plasma to interact with electromagnetic fields. The fields dictate the motion of the particles and, simultaneously, the motion of the particles alter the fields. This strongly coupled, self-consistent interaction is difficult to simulate. Many computational techniques have been used to simulate plasma [6,30]. The models are diverse and span a large range of physical scales (both spatial and temporal). The granularity of the model depends on the level of physical description required. For micro physics, one uses kinetic approaches (Vlasov, Boltzmann). When macro physics is under investigation, one must resort to fluid (moment-based) models. For accurate and predictable macro scale models, the effects of micro physics must be effectively be included. This leads to the need for efficient multiscale kinetic algorithms. Such algorithms demand implicit methods capable of bridging temporal scales.

Kinetic methods, in general, more accurately represent the physics of a plasma by capturing the effects of the well-known Vlasov-Maxwell system. One commonly used, efficient discretization is the particle-in-cell (PIC) method. Traditional PIC implementations employ explicit time stepping and suffer from a strict Courant-Friedrichs-Lewy (CFL) condition that limits time steps to be on the order of the electron plasma frequency ($\omega_{p,e}$). Another well-known difficulty is the finite-grid instability, which requires the computational mesh to be on the order of a Debye length (λ_D), making large computational domains infeasible. On the other hand, in the moment-based fluid (MB) approach, the plasma is treated as a fluid continuum and is coupled to Maxwell's equations

via charge density and current density. There are an abundance of flavors of MB methods, each of which is capable of targeting various levels of physical realism (Ideal MHD, Resistive MHD, Hall MHD, Extended MHD, Two-Fluid, etc.) [26]. In general, the fluid-like approach is much less memory intensive than a kinetic PIC simulation and allows for larger temporal and spatial scales. The MB methodology is traditionally used to explore slower plasma dynamics on larger computational domains.

Recently, a fully-implicit, electrostatic, one-dimensional, charge- and energy–conserving kinetic PIC algorithm was proposed and implemented [18]. The fully implicit nature of the algorithm is stable against temporal and spatial instabilities. This allows for a relaxed time step and larger computational domains, and it ameliorates many shortcomings of traditional PIC. Central to the algorithm is a Jacobian-Free Newton-Krylov [34] iteration that converges the Maxwell system at each time step, utilizing the particle system to supply charge density and current density to the Maxwell system. In turn, every evaluation of the residual in the Krylov method requires that the particle system be implicitly integrated. In this way, the field values and particle states are converged nonlinearly to a tight tolerance. This enhances the stability and accuracy properties as compared to prior implicit PIC implementations. The number of Krylov iterations required at each time step is often large. Even with advances in GPU computing, heterogeneous architectures, and clever implementations [19], the primary computational cost remains particle movement in each residual calculation. Thus, it is crucial to design a robust preconditioning method to reduce the number of Krylov iterations.

The use of physics-based preconditioning (PBP), in conjunction with JFNK, has proved fruitful in applications where disparate time scales are present [35]. Stiff hyperbolic PDE systems are transformed into a scalar parabolic PDE that targets fast wave behavior. The eigenvalues of the original system are, thus, effectively clustered, allowing implicit time-integration schemes to step over the stiff time scale while maintaining a reasonable number of Krylov iterations. Moreover, the scalar parabolic PDE is often amenable to fast multigrid methods [8], adding little to the overall computational cost.

A recent study demonstrates the effectiveness of using an electrostatic MB two-fluid plasma (TFP) model to accelerate the fore-mentioned JFNK-based kinetic solver [20]. The preconditioning process draws heavily from PBP, building a reduced PDE for the electron momentum that is parabolic in nature. In this way, the electrostatic moment model provides an inexpensive and easily inverted approximation to the electrostatic kinetic Jacobian. This enables the use of large implicit time steps while effectively bounding the number of Krylov residual evaluations independent of ion-electron mass ratios. Speedups of approximately three orders of magnitude versus explicit PIC were demonstrated.

Efforts are currently being directed towards a fully implicit, electromagnetic, JFNK-based kinetic solver, motivating the necessity of developing a suitable MB electromagnetic preconditioning strategy [18]. The introduction of an electromagnetic model makes using a PBP to reduce the MB system to a scalar elliptic PDE significantly more difficult. Recent work has been performed investigating such an approach [16, 17]. Such preconditioners require physical intuition and some amount of experimentation to construct. Instead, a more mathematically rigorous alternative is used that requires little to no physics insight, but still generates systems amenable to fast multigrid methods. We propose the use of First-Order System Least-Squares (FOSLS) and Nested Iteration (NI) to approximately invert a full electromagnetic (Darwin) two-fluid plasma system as a preconditioner.

The FOSLS methodology has been successfully applied to a large variety of problems [10]. First-order systems of PDEs are recast as the minimization of a functional. The resulting weak formulation admits a bilinear form that, if well-posed in the proper Hilbert space, allows the use of standard finite element spaces to generate a discrete system amenable to multigrid methods (Spaces for which effective multigrid methods exist include products of $\mathcal{H}^1$, $\mathcal{H}(\text{Div})$, and $\mathcal{H}(\text{Curl})$. Recently, the FOSLS approach was successfully applied to incompressible, resistive MHD [2]. MHD models take an average between an electron fluid and ion fluid to track bulk plasma behavior. The charge density is assumed to be zero throughout the domain (i.e., quasineutral). The model leads naturally to the addition of a resistive Ohm's law that, when used to eliminate the electric field in

the MHD equations, is amenable to the construction of an H^1-elliptic system. In the TFP context, no such Ohm's law can be supplied; well-posedness is sought via other mechanisms.

One of the main strategies for nonlinearities that result from a FOSLS discretization is nested iteration (NI). With NI, most of the work is done on coarse grids where nonlinear iteration is inexpensive. The approximation to the solution on a coarse grid is interpolated to a finer grid and used as an initial guess. The process is continued to progressively finer grids until a desired error tolerance or mesh is reached. In the context of a nonlinear system, an important goal of NI is to produce an approximation that is within the basis of attraction for Newtons method on a relatively coarse mesh and, thus, on all subsequent meshes. Often, by the time the finest grid is reached, very few nonlinear iterations are required. (It was shown in [22, 23] that, if the system is $\mathcal{V}$-elliptic in $\mathcal{H}^{1+\epsilon}$, for a sufficiently fine grid, only one Newton step and a fixed number of multigrid V-cycles are necessary to achieve an approximation to within discretization error on that grid.) In resistive MHD applications, it was demonstrated that the additional work performed on the coarse grids is a very small fraction of the total work on the finest grid, significantly reducing the total amount of overall work needed [2].

The goal of this thesis is to explore, in-depth, the viability of such an NI-FOSLS approach to TFP systems. The MB-TFP system is analyzed in the context of a stand-alone FOSLS solver with consideration of preconditioning a kinetic-JFNK approach. During JFNK iteration, accumulated moment information from the particle system would normally be used to both linearize and supply a closure to the moment-based preconditioner. In the stand-alone context we, instead, assume a simple isothermal closure in the stress tensor. The performance of the nonlinear system as a fully independent solver is examined. To this end, the linearized approximations in the Newton iteration are more complex than would be produced in preconditioning a kinetic-JFNK iteration. Thus, the results presented here strongly imply that the NI-FOSLS approach would be successful as a preconditioner for PIC-based methods.

The remainder of the thesis is organized as follows: In Chapter 2, an introduction to the basics of plasma physics is presented to bring the reader up to speed with terminology, notation, and

assumptions. Various plasma models are introduced in Chapter 3, including the Darwin approximation that removes light waves from the model, making it possible to couple to non-relativistic PIC simulations. Chapter 4 presents the FOSLS methodology for nonlinear PDEs, Nested Iteration, and a brief introduction to Algebraic Multigrid (AMG). In Chapter 5, the FOSLS formulation for TFP is developed and proved to be $\mathcal{H}^1$-elliptic. Crucial to this proof are Section 5.2, where the Darwin approximation is scaled and shown to be $\mathcal{H}^1$-elliptic, and Section 5.3, where the momentum density and number density equations are modified with an additional constraint and scaled to also provide $\mathcal{H}^1$-ellipticity. The main result follows in Section 5.5, where the full nonlinear coupling between the fluid model and Maxwell is described and the full system is proven to be uniformly $\mathcal{H}^1$-elliptic in a neighborhood of the exact solution. Numerical results for the NI-FOSLS approach applied to several test problems are presented in Chapter 6. Finally, discussion and future work appear in Chapter 7.

Chapter 2

Plasma Physics Background

Plasma is a state of mater in which atoms have dissociated into their constituent ions and electrons. The availability of this free charge allows the plasma to interact with electromagnetic fields. The fields dictate the motion of the particles and, simultaneously, the motion of the particles alter the fields. Plasma appears in a large variety of applications (e.g., Earth's magnetosphere, fusion reactors, propulsion systems, space craft re-entry and, solar weather); it is becoming increasingly important to be able to model plasma systems quickly and accurately.

This chapter introduces the basic concepts from plasma physics, the underlying assumptions used for the remainder of this thesis, and the Vlasov-Maxwel/Darwin system, which acts as the bedrock from which our models will be derived in Chapter 3.

2.1 Distribution Functions

The most complete description of plasma dynamics is in a 6-dimensional phase space, which consists of position, $\mathbf{r} = (x, y, z)$, and velocity, $\mathbf{v} = (u, v, w)$. Ideally, one would track every particle of each species as a point in phase space. Unfortunately, the number of particles, in even a modest plasma, far exceeds our current computational limits. For example, magnetically confined fusion plasmas often have densities that can exceed $10^{20}\ \frac{\text{particles}}{\text{m}^3}$ [6].

Instead, it is more convenient to represent each species, denoted by the subscript α, of the plasma with a distribution function,

$$f_\alpha(\mathbf{r}, \mathbf{v}, t) : \mathbb{R}^6 \times \mathbb{R} \to \mathbb{R} \qquad \alpha = 1, \ldots, S\,. \tag{2.1}$$

Each distribution function maps a point in phase space, $(\mathbf{r}, \mathbf{v}) \in \mathbb{R}^6$, at a given time, $t \in \mathbb{R}$, to a probability scaled by the total number of particles. To solidify the meaning of f_α, consider a plasma contained in a spatial region, $\Omega_\mathbf{r}$, and a velocity region, $\Omega_\mathbf{v}$. Then a distribution is defined such that

$$N_\alpha(t) = \int_{\Omega_\mathbf{v}} \int_{\Omega_\mathbf{r}} f_\alpha(\mathbf{r}, \mathbf{v}, t) \, \mathrm{d}\mathbf{r} \, \mathrm{d}\mathbf{v} \,, \tag{2.2}$$

where $N_\alpha(t)$ is the total number of particles of species α in the subset of phase space denoted by $\Omega_\mathbf{r} \times \Omega_\mathbf{v}$. Particles are assumed to be neither created nor destroyed. In other words, $N_\alpha(t) = N_\alpha$ remains constant through time. It is assumed that, for all species, α, $f_\alpha(\mathbf{r}, \mathbf{v}, t)$ is smooth and bounded, and has compact support given by $\mathrm{supp}(f_\alpha) = \Omega_\mathbf{r} \times \Omega_\mathbf{v}$. The smoothness is justified by assuming a large number of particles per unit volume, as well as assuming an ensemble-averaged distribution function [12]. The boundedness follows from a physical limit on the magnitude of density, and the compact support is due to the finite extent of the plasma volume and the absence of particles with infinite velocities.

Velocity moments of distribution functions play an important role in plasma theory and the derivation of the fluid equations. Each velocity moment (henceforth, moment) represents a different physical quantity. Denote the m-th moment of species α as

$$\psi_\alpha^m(\mathbf{r}, t) = \langle \mathbf{v}^m, f_\alpha(\mathbf{r}, \mathbf{v}, t) \rangle = \int_{\Omega_\mathbf{v}} \mathbf{v}^m f_\alpha(\mathbf{r}, \mathbf{v}, t) \, \mathrm{d}\mathbf{v} \,, \tag{2.3}$$

where $\mathbf{v}^m$ is a rank m tensor. For example, $\mathbf{v}^2 = \mathbf{v}\mathbf{v}^T$ is the outer product, producing a 3×3 matrix. See Table 2.1 for a list of the physical meaning of the first three moments. A more convenient nomenclature is adopted. The first moment, $\mathbf{p}_\alpha$, is referred to as the momentum density, and the second moment, $\underline{\mathbf{S}}_\alpha$, is referred to as the stress tensor (or total stress tensor). The moments depend only on space and time. The velocity variable has been integrated out. The moments are a more intuitive way to describe a plasma than the 7 dimensional phase space.

Moment	Variable	Physical Quantity
ψ^0_α	$n_\alpha(\mathbf{r},t)$	Number Density
ψ^1_α	$\mathbf{p}_\alpha(\mathbf{r},t)$	Momentum Number Density
ψ^2_α	$\underline{\mathbf{S}}_\alpha(\mathbf{r},t)$	Momentum Flux Density Tensor

Table 2.1: The first three velocity moments as defined by (2.3): Number Density, Momentum Number Density, Momentum Flux Density Tensor.

An often-used peripheral variable is the bulk plasma velocity:

$$\mathbf{u}_\alpha(\mathbf{r},t) = \frac{\mathbf{p}_\alpha(\mathbf{r},t)}{n_\alpha(\mathbf{r},t)}\,. \tag{2.4}$$

While not strictly a moment, the bulk velocity is easy to conceptualize and is used in many derivations. The reader should associate $\mathbf{u}_\alpha$ with the standard fluid velocity observed in classical fluid mechanics. From here on out, $\mathbf{p}_\alpha$ is used in favor of $\mathbf{u}_\alpha$ when ever possible due to its natural analog and symmetry with n_α in Maxwell's equations (Section 2.3).

2.2 The Vlasov Equation

The force on particle p of charge q_p at location $\mathbf{r}_p$ with velocity $\mathbf{v}_p$ is given by the well know Lorentz force,

$$\mathbf{F}_p = q_p\left[\mathbf{E}(\mathbf{r}_p) + \mathbf{v}_p \times \mathbf{B}(\mathbf{r}_p)\right]\,, \tag{2.5}$$

where $\mathbf{E}$ is a known electric field and $\mathbf{B}$ is a known magnetic field. Assume that all particles of the same species contained in an infinitesimal volume of phase space, $d\mathbf{r}d\mathbf{v}$, centered at $(\mathbf{r}_p, \mathbf{v}_p)$, experience the same force as the force on particle p. This observation, coupled with a simple conservation argument, leads to the realization that $\frac{\mathrm{d}f_\alpha}{\mathrm{d}t} = \frac{\delta_c f_\alpha}{\delta t}$ [26]. The notation $\frac{\delta_c f_\alpha}{\delta t}$ represents the variation in the distribution function due to collisions between particles and is often referred to as the collisional derivative. By writing the distribution function in component form and expanding the total time derivative, it is seen that

$$\frac{\mathrm{d}f_\alpha}{\mathrm{d}t} = \frac{\partial f_\alpha}{\partial t} + \frac{\partial f_\alpha}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial f_\alpha}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} + \frac{\partial f_\alpha}{\partial z}\frac{\mathrm{d}z}{\mathrm{d}t} + \frac{\partial f_\alpha}{\partial u}\frac{\mathrm{d}u}{\mathrm{d}t} + \frac{\partial f_\alpha}{\partial v}\frac{\mathrm{d}v}{\mathrm{d}t} + \frac{\partial f_\alpha}{\partial w}\frac{\mathrm{d}w}{\mathrm{d}t}\,. \tag{2.6}$$

Collecting derivatives, $\mathbf{v} = \left[\frac{\mathrm{d}x}{\mathrm{d}t}, \frac{\mathrm{d}y}{\mathrm{d}t}, \frac{\mathrm{d}z}{\mathrm{d}t}\right]^T$, $\mathbf{a} = \left[\frac{\mathrm{d}u}{\mathrm{d}t}, \frac{\mathrm{d}v}{\mathrm{d}t}, \frac{\mathrm{d}w}{\mathrm{d}t}\right]^T$, $\nabla f_\alpha = \left[\frac{\partial f_\alpha}{\partial x}, \frac{\partial f_\alpha}{\partial y}, \frac{\partial f_\alpha}{\partial z}\right]^T$, and $\nabla_{\mathbf{v}} f_\alpha = \left[\frac{\partial f_\alpha}{\partial u}, \frac{\partial f_\alpha}{\partial v}, \frac{\partial f_\alpha}{\partial w}\right]^T$, and substituting the relationship $\mathbf{F} = m\mathbf{a}$, it easily follows that

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} \left[\mathbf{E} + \mathbf{v} \times \mathbf{B}\right] \cdot \nabla_{\mathbf{v}} f_\alpha = \frac{\delta_c f_\alpha}{\delta t} \,. \tag{2.7}$$

Equation (2.7) is known as the Boltzmann equation. Only collisionless plasmas are considered by letting $\frac{\delta_c f}{\delta t} \to 0$. The collisionless version of the Boltzmann equation is known as the Vlasov equation:

$$\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} \left[\mathbf{E} + \mathbf{v} \times \mathbf{B}\right] \cdot \nabla_{\mathbf{v}} f_\alpha = 0 \,. \tag{2.8}$$

Systems that obey (2.7) or (2.8) are often referred to as having a "kinetic description". The Vlasov (Boltzmann) equation describes how electric and magnetic fields alter the state of the distribution function, and ultimately the charge density and current density. In turn, the electric and magnetic fields depend upon these densities through Maxwell's equations.

2.3 Maxwell's Equations

Crucial to the time evolution of a plasma is the interaction of charged particles with the electric field, $\mathbf{E}$, and magnetic field, $\mathbf{B}$. The location and velocities of charged particles produces an electromagnetic field while, simultaneously, the electromagnetic field alters the acceleration of the particles. Maxwell's equations describe fields produced in the presence of a known charge density and current density.

Given a plasma consisting of multiple species, α, the charge density, ρ, and current density, $\mathbf{j}$, are defined as

$$\rho(\mathbf{r}, t) = \sum_\alpha q_\alpha n_\alpha(\mathbf{r}, t) \,, \tag{2.9}$$

$$\mathbf{j}(\mathbf{r}, t) = \sum_\alpha q_\alpha \mathbf{p}_\alpha(\mathbf{r}, t) \,. \tag{2.10}$$

It should be noted that the charge density and current density can both be expanded as moments of the distribution function defined in Table 2.1:

$$\rho(\mathbf{r}, t) = \sum_\alpha q_\alpha \int_{\Omega_{\mathbf{v}}} f_\alpha(\mathbf{r}, \mathbf{v}, t) \, \mathrm{d}\mathbf{v} \,, \tag{2.11}$$

$$\mathbf{j}(\mathbf{r},t) = \sum_{\alpha} q_\alpha \int\limits_{\Omega_{\mathbf{v}}} \mathbf{v} f_\alpha(\mathbf{r},\mathbf{v},t)\,\mathrm{d}\mathbf{v}\,. \tag{2.12}$$

Without derivation, Maxwell's equations are

$$\nabla \times \mathbf{E}(\mathbf{r},t) + \frac{\partial \mathbf{B}(\mathbf{r},t)}{\partial t} = \mathbf{0}\,, \tag{2.13}$$

$$\epsilon_0 \nabla \cdot \mathbf{E}(\mathbf{r},t) = \rho(\mathbf{r},t)\,, \tag{2.14}$$

$$-\epsilon_0 \frac{\partial \mathbf{E}(\mathbf{r},t)}{\partial t} + \frac{1}{\mu_0}\nabla \times \mathbf{B}(\mathbf{r},t) = \mathbf{j}(\mathbf{r},t)\,, \tag{2.15}$$

$$\nabla \cdot \mathbf{B}(\mathbf{r},t) = 0\,, \tag{2.16}$$

where the constants, ϵ_0 and μ_0, are the permittivity of free space and the permeability of free space, respectively. Equation (2.13) is Faraday's law, (2.14) is Gauss' law, (2.15) is Ampère's law, and (2.16) is the solenoidal constraint on $\mathbf{B}$. For a particularly accessible derivation and clear explanation of physical meaning, see [47]. Maxwell's equations are written in a more convenient form by defining the differential matrix,

$$\mathbb{M} = \begin{bmatrix} \nabla\times & \partial_t \\ \epsilon_0\nabla\cdot & 0 \\ -\epsilon_0\partial_t & \frac{1}{\mu_0}\nabla\times \\ 0 & \nabla\cdot \end{bmatrix}, \tag{2.17}$$

electromagnetic field vector,

$$\mathbf{Z} = [\mathbf{E}(\mathbf{r},t), \mathbf{B}(\mathbf{r},t)]^T\,,$$

and source vector,

$$\mathbf{S} = [\mathbf{0}, \rho(\mathbf{r},t), \mathbf{j}(\mathbf{r},t), 0]^T\,.$$

Maxwell's equations now simply read

$$\mathbb{M}\,\mathbf{Z} = \mathbf{S}\,. \tag{2.18}$$

In Section 5.2, the properties of (2.18) are analyzed with respect to a FOSLS formulation under the Darwin approximation. We now have all the ammunition needed to define the overarching plasma system that guides both kinetic and and MB methods.

2.4 Vlasov-Maxwell System

The Vlasov-Maxwell (VM) system is produced by coupling the Vlasov equation (2.8) and Maxwell's equations (2.13) - (2.16) by means of the moment expansions of charge and current densities (2.11) and (2.12). For an S species plasma, this gives

$$\begin{aligned}
\tfrac{\partial f_\alpha}{\partial t} + \mathbf{v}\cdot\nabla f_\alpha + \tfrac{q_\alpha}{m_\alpha}\left[\mathbf{E}+\mathbf{v}\times\mathbf{B}\right]\cdot\nabla_v f_\alpha &= 0\,, \qquad (\alpha = 1,\ldots,S)\,,\\
\nabla\times\mathbf{E} + \tfrac{\partial \mathbf{B}}{\partial t} &= \mathbf{0}\,,\\
\epsilon_0\nabla\cdot\mathbf{E} &= \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} f_\alpha \,\mathrm{d}\mathbf{v}\,,\\
-\epsilon_0\tfrac{\partial \mathbf{E}}{\partial t} + \tfrac{1}{\mu_0}\nabla\times\mathbf{B} &= \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} \mathbf{v} f_\alpha \,\mathrm{d}\mathbf{v}\,,\\
\nabla\cdot\mathbf{B} &= 0\,.
\end{aligned} \tag{2.19}$$

This 7-dimensional (i.e., $\mathbf{r}$, $\mathbf{v}$, and t) system of integro-differential equations is the highest-level model that we consider. The system of integro-differential equations is not computationally tractable for large spatial or time scales, due to requiring a 7D discretization. The VM system is made more manageable for larger scales by casting the equations into a purely differential setting. Two common methods of approaching (2.19) are explored in Chapter 3.

2.5 The Darwin Approximation

The VM system supports a myriad of waves, the fastest being the light wave, which travels with wave speed $c = \frac{1}{\sqrt{\epsilon_0\mu_0}}$. This is easily demonstrated in a vacuum ($\rho = 0$ and $\mathbf{j} = 0$) by taking a curl of (2.13) and, a time derivative of (2.15), and eliminating the $\nabla\times\partial_t\mathbf{B}$. This process results in

$$\frac{1}{c^2}\partial_{tt}\mathbf{E} - \Delta\mathbf{E} = \mathbf{0}\,, \tag{2.20}$$

where $\nabla\times\nabla\times\mathbf{E}$ was reduced to $-\Delta\mathbf{E}$ using the fact that, in a vacuum, $\nabla\cdot\mathbf{E} = 0$. Solving Maxwell's equations on a discrete grid produces light waves that travel less than c due to the numerical dispersion relation. Thus, when coupling a discrete Maxwell solver to a particle simulation, it is possible that the particles can travel faster than the numerical speed of light. This unnatural

numerical artifact causes an effect known as numerical Cherenkov radiation, whereby computational particles artificially radiate (i.e., lose energy) [41].

In non-relativistic kinetic simulations, the speed of light is spurious in PIC simulations, and it may lead to enhanced numerical noise. Overcoming these instabilities can be accomplished by the injection of artificial dissipation, but only at the expense of energy conservation [20]. These issues can be circumvented by taking the asymptotic limit, $c \to \infty$, or, alternatively, $\epsilon_0 \to 0$. Unfortunately, this limit forces quasineutrality:

$$\rho = \lim_{\epsilon_0 \to 0} \epsilon_0 \nabla \cdot \mathbf{E} = 0 \,. \tag{2.21}$$

When coupling to a non-relativistic kinetic simulation, this is insufficient; charge separation effects are required. For these reasons, a Darwin model is preferred. The model is an approximation that decomposes fields into their solenoidal and irrotational components in order to analytically project out light waves while maintaining charge separation in Gauss' law [32].

The Darwin approximation is rigorously constructed by asymptotic expansion of a Lagrangian formulation of Maxwell's equations with $v/c \ll 1$ [36]. To more intuitively build the approximation, consider separating vector fields into their solenoidal (divergence-free) and irrotational (curl-free) components: $\mathbf{E} = \mathbf{E}_r + \mathbf{E}_s$, $\mathbf{B} = \mathbf{B}_s$, and $\mathbf{j} = \mathbf{j}_r + \mathbf{j}_s$. The subscript s denotes solenoidal components (i.e., $\nabla \cdot \mathbf{E}_s = 0$) and the subscript r denotes irrotational components (i.e., $\nabla \times \mathbf{E}_r = 0$). It should be noted that the magnetic field is naturally solenoidal. Expanding equations (2.13) - (2.16) into these components leads to

$$\begin{aligned} \epsilon_0 \nabla \cdot \mathbf{E}_r &= \rho \,, \\ \nabla \cdot \mathbf{B} &= 0 \,, \\ -\tfrac{1}{c^2}\partial_t \mathbf{E}_s - \mu_0 \mathbf{j}_s + \nabla \times \mathbf{B} &= \mu_0 \mathbf{j}_r + \tfrac{1}{c^2}\partial_t \mathbf{E}_r \,, \\ \nabla \times \mathbf{E}_s + \partial_t \mathbf{B} &= \mathbf{0} \,. \end{aligned} \tag{2.22}$$

The third equation contains an entirely solenoidal field on the left-hand side and an entirely irrotational field on the right-hand side. It is clear that the solenoidal component of the electric field is what supports the propagation of light waves. These waves are projected out of the system

by taking the asymptotic limit that the speed of light wave propagation is infinite, $\frac{1}{c^2}\partial_t \mathbf{E}_s \to \mathbf{0}$ (see [36, 44] for a detailed discussion). What remains is the Darwin model:

$$\begin{aligned} \epsilon_0 \nabla \cdot \mathbf{E}_r &= \rho\,, \\ \nabla \cdot \mathbf{B} &= 0\,, \\ -\epsilon_0 \partial_t \mathbf{E}_r + \tfrac{1}{\mu_0} \nabla \times \mathbf{B} &= \mathbf{j}\,, \\ \nabla \times \mathbf{E}_s + \partial_t \mathbf{B} &= 0\,. \end{aligned} \tag{2.23}$$

Coupling (2.23) with (2.8) via moment equations gives us our final overarching system of equations, the Vlasov-Darwin model:

$$\begin{aligned} \tfrac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \tfrac{q_\alpha}{m_\alpha} \left[(\mathbf{E}_r + \mathbf{E}_s) + \mathbf{v} \times \mathbf{B} \right] \cdot \nabla_v f_\alpha &= 0 \qquad (\alpha = 1, \ldots, S)\,, \\ \nabla \times \mathbf{E}_r &= \mathbf{0} \\ \epsilon_0 \nabla \cdot \mathbf{E}_r &= \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} f_\alpha \, \mathrm{d}\mathbf{v}\,, \\ -\epsilon_0 \partial_t \mathbf{E}_r + \tfrac{1}{\mu_0} \nabla \times \mathbf{B} &= \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} \mathbf{v} f_\alpha \, \mathrm{d}\mathbf{v}\,, \\ \nabla \cdot \mathbf{B} &= 0 \\ \nabla \times \mathbf{E}_s + \partial_t \mathbf{B} &= \mathbf{0} \\ \nabla \cdot \mathbf{E}_s &= 0\,. \end{aligned} \tag{2.24}$$

To reiterate, the final Darwin approximation does not support the propagation of light waves, and the model is only accurate in non-relativistic regimes. The key property that separates the Darwin model from a standard, quasineutral Maxwell model with a dropped displacement current is that it supports charge separation effects in Gauss' law. This is desirable when coupling to a TFP where a distinct number densities, for each species, must be tracked independently.

Chapter 3

Plasma Models

There are many levels of physical descriptions of a plasma. Kinetic models address micro physics effects, while fluid-like models are more convenient for the larger macro physics. Both of these levels are addressed by considering (1) a kinetic particle-in-cell method, which amounts to using the method of characteristics in a 6-dimensional phase space to directly approximate solutions to the Vlasov equation, and (2) a moment-based fluid model that couples self-consistently to Maxwell's field equations via linear combinations of the moment variables introduced in Section 2.1.

Each approach stems from the same VM system (2.19) but differs significantly in its approach to discretization. Described in broad strokes, the methods differ in when discretizations and moment integrals are performed. The PIC method first discretizes the distribution function, and moment integrals are taken only when coupling to Maxwell through the charge density and current density. In contrast, the MB fluid method examines moment integrals of the entire Vlasov equation to produce new primitive fluid-like variables that couple directly to Maxwell without need for integration; the resulting system of PDEs is then discretized (See Figure 3.1 for a coarse description).

Recently, a fully implicit, nonlinear, kinetic PIC algorithm was developed in the electrostatic and electromagnetic limits [17, 18, 20]. The kinetic PIC algorithm is implicitly coupled to a MB model, effectively marrying the the micro and macro levels to build a true scale-bridging algorithm. The end of this chapter is dedicated to a brief overview of this new method, with special considera-

tion as to how inversion of a MB two-fluid plasma model would provide an excellent preconditioner to these state-of-the-art implicit algorithms.

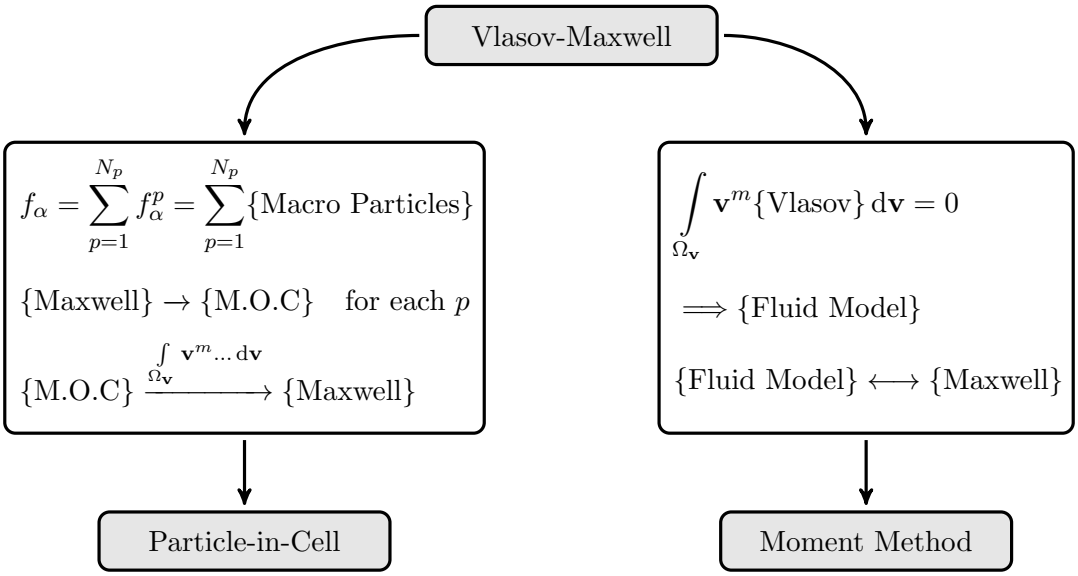

Figure 3.1: The general idea of both the PIC and MB methods. In PIC, the distribution functions are discretized into macro-particles. The force generated from the electromagnetic fields are used in the method of characteristics (M.O.C) to integrate the particles forward in time. The results from the M.O.C are then injected into Maxwell via moments. In MB, moments of the Vlasov equation generate a fluid model. The fluid model is then coupled self consistently to Maxwell.

3.1 Particle-in-Cell Method

In PIC, the distribution function is assumed to take the form of a linear combination of so-called macro-particles. Each macro-particle is moved via a simple set of kinematic ODEs, ultimately approximating the method of characteristics of the Vlasov equation (2.8). The carefully chosen shape of the macro-particles in phase-space make the moment integrals in (2.19) reduce to a simple sum that can be rapidly evaluated. The electromagnetic fields can then be interpolated back to the macro-particle positions in order to compute the Lorentz force. The process is repeated to move the particles forward in time using either an explicit time step or, more recently, a fully implicitly

iterative method [18].

Consider a thought experiment where every particle in a plasma is tracked. This, of course, is not computationally attainable. But, nevertheless, assign to each particle an index $p = 1, \ldots, N_T$, where $N_T = \sum_\alpha N_\alpha$ is the total number of particles. The distribution function of species α is assumed to take the form

$$f_\alpha(\mathbf{r}, \mathbf{v}, t) = \sum_{p \in \mathcal{S}_\alpha} \delta(\mathbf{r} - \mathbf{r}_p(t)) \delta(\mathbf{v} - \mathbf{v}_p(t)) \,, \tag{3.1}$$

where δ is the Dirac delta function and $\mathcal{S}_\alpha$ represents the set of all particle indices belonging to a species α. Observe that (2.8) is formally linear in f_α, motivating the use of method of characteristics on each particle [38]. That is, given the initial position and velocity of a particle, the particle trajectory is described by

$$\begin{aligned} \frac{\partial \mathbf{r}_p(t)}{\partial t} &= \mathbf{v}_p(t) \,, \\ \frac{\partial \mathbf{v}_p(t)}{\partial t} &= \frac{q_p}{m_p} \left[\mathbf{E}(\mathbf{r}_p(t), t) + \mathbf{v}_p \times \mathbf{B}(\mathbf{r}_p(t), t) \right] . \end{aligned}$$

Unfortunately, as the particles, and thus the distribution functions, evolve forward in time, the electromagnetic fields dependence on charge and current density must too be considered. Substituting (3.1) in to (2.14) and (2.15) gives

$$\epsilon_0 \nabla \cdot \mathbf{E} = \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} \sum_{p \in \mathcal{S}_\alpha} \delta(\mathbf{r} - \mathbf{r}_p) \delta(\mathbf{v} - \mathbf{v}_p) \, \mathrm{d}\mathbf{v} = \sum_{p=1}^{N_T} q_p \delta(\mathbf{r} - \mathbf{r}_p) \,, \tag{3.2}$$

and

$$-\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \frac{1}{\mu_0} \nabla \times \mathbf{B} = \sum_{\alpha=1}^{S} q_\alpha \int_{\Omega_\mathbf{v}} \mathbf{v} \sum_{p \in \mathcal{S}_\alpha} \delta(\mathbf{r} - \mathbf{r}_p) \delta(\mathbf{v} - \mathbf{v}_p) \, \mathrm{d}\mathbf{v} = \sum_{p=1}^{N_T} q_p \mathbf{v}_p \delta(\mathbf{r} - \mathbf{r}_p) \,. \tag{3.3}$$

As a reminder, S is the number of species, whereas $\mathcal{S}_\alpha$ is the set of particle indices for species type α. The delta function in velocity space integrates away leaving only a summation of spatial delta functions in (3.2), and particle velocities multiplied by spatial delta functions in (3.3).

Combining the above models and simplifying, a hypothetical plasma model is assembled, in

which the behavior of every particle is captured. The particles follow the kinematic equations

$$\begin{aligned} \partial_t \mathbf{r}_p(t) &= \mathbf{v}_p(t)\,, & p = 1, \ldots, N_T\,, \\ \partial_t \mathbf{v}_p(t) &= \tfrac{q_p}{m_p}\left[\mathbf{E}\left(\mathbf{r}_p(t), t\right) + \mathbf{v}_p(t) \times \mathbf{B}\left(\mathbf{r}_p(t), t\right)\right], & p = 1, \ldots, N_T\,, \end{aligned} \tag{3.4}$$

and the fields are updated by

$$\begin{aligned} \nabla \times \mathbf{E}\left(\mathbf{r}, t\right) + \partial_t \mathbf{B}\left(\mathbf{r}, t\right) &= \mathbf{0}\,, \\ \epsilon_0 \nabla \cdot \mathbf{E}\left(\mathbf{r}, t\right) &= \sum_{p=1}^{N_T} q_p \delta(\mathbf{r} - \mathbf{r}_p(t))\,, \\ -\epsilon_0 \partial_t \mathbf{E}\left(\mathbf{r}, t\right) + \tfrac{1}{\mu_0} \nabla \times \mathbf{B}\left(\mathbf{r}, t\right) &= \sum_{p=1}^{N_T} q_p \mathbf{v}_p(t) \delta(\mathbf{r} - \mathbf{r}_p(t))\,, \\ \nabla \cdot \mathbf{B}\left(\mathbf{r}, t\right) &= 0\,. \end{aligned} \tag{3.5}$$

Although verbose, each variable's dependence on $\mathbf{r}$ and t is explicitly stated to make clear how a discretization would be performed. The above equations are often referred to as a Klimontovich model. The PIC method is closely related to (3.4) and (3.5). The main difference is that the measure-zero spatial delta function is replaced with a more smoother shape function, $\beta_\gamma(\mathbf{r})$, with $\operatorname{supp} \beta_\gamma \neq 0$. This is discussed in detail in the next section.

3.1.1 Shape Functions

In practice (3.4) and (3.5) are impossible to implement due to current constraints on computational resources. The requirement of modeling every particle is relaxed to, instead, modeling a large number of clusters of particles. The clusters of particles are called macro-particles and are assumed to move together as a cohesive unit through phase-space. In reality, this should not be the case: the particles that comprise each macro-particle should experience different forces and, as the macro-particles move through the simulation, should experience dispersive effects. For a heuristic argument as to why this may be neglected, see [6].

Each macro-particle is given a finite spatial shape but remains a Dirac delta function in velocity space. Redefine $p = 1, \ldots, N_m$ as the macro-particle index, where N_m is the number of macro-particles and N_p is the number of particles contained in each macro-particle ($N_T = N_m N_p$).

Each distribution function now takes the form

$$f_\alpha = \sum_{p=1}^{N_m} N_p \beta_\gamma(\mathbf{r} - \mathbf{r}_p)\delta(\mathbf{v} - \mathbf{v}_p)\,.$$

The shape function, $\beta_\gamma(\xi)$, is often chosen to abide by the following [38]:

(1) The supp(β_γ) is a bounded, simply connected region on the order of the size of the discrete mesh spacing.

(2) The integral is unity, $\int_{-\infty}^{\infty} \beta_\gamma(\xi)\,\mathrm{d}\xi = 1$

(3) The shape function is radially symmetric, that is, $\beta_\gamma(\xi - \eta) = \beta_\gamma(\eta - \xi)$

With some amount of foresight, the shape functions are chosen to be constructed from B-splines. These functions posses several nice properties that are utilized in [18] to achieve charge conservation. In 1D, the first few B-splines are defined as follows:

$$b_0(\xi) = \begin{cases} 1 & : |\xi| < \frac{1}{2}\,, \\ 0 & : \text{otherwise}\,. \end{cases} \tag{3.6}$$

$$b_1(\xi) = \begin{cases} 1+\xi & : -1 < \xi \leq 0\,, \\ 1-\xi & : 0 < \xi \leq 1\,, \\ 0 & : \text{otherwise}\,. \end{cases} \tag{3.7}$$

$$b_2(\xi) = \begin{cases} \frac{1}{8}(2\xi+3)^2 & : -\frac{3}{2} < \xi \leq -\frac{1}{2}\,, \\ \frac{3}{4} - \xi^2 & : -\frac{1}{2} < \xi \leq \frac{1}{2}\,, \\ \frac{1}{8}(3-2\xi)^2 & : \frac{1}{2} < \xi < \frac{3}{2}\,, \\ 0 & : \text{otherwise}\,. \end{cases} \tag{3.8}$$

Define the B-spline induced 1D shape function of order γ as

$$\beta_\gamma(x) = \frac{1}{\Delta_p} b_\gamma\left(\frac{x}{\Delta_p}\right)\,, \tag{3.9}$$

where Δ_p denotes the length-scale of the support of the macro-particle in one dimension. In multiple dimensions, the shape function is simply the tensor product between 1D shape functions:

$$\beta_\gamma(\mathbf{r} - \mathbf{r}_p) = \frac{1}{\Delta_{p,x}\Delta_{p,y}\Delta_{p,z}} b_\gamma\left(\frac{x - x_p}{\Delta_{p,x}}\right) b_\gamma\left(\frac{y - y_p}{\Delta_{p,y}}\right) b_\gamma\left(\frac{z - z_p}{\Delta_{p,z}}\right)\,. \tag{3.10}$$

A rigorous definition of shape functions is left for the curious reader to explore (see [6]). The minor change to the shape function leads to the overlying system of equations that govern PIC methods:

$$\begin{aligned} \partial_t \mathbf{r}_p(t) &= \mathbf{v}_p(t), & p = 1, \ldots, N_m\,, \\ \partial_t \mathbf{v}_p(t) &= \tfrac{q_p}{m_p}\left[\mathbf{E}\left(\mathbf{r}_p(t), t\right) + \mathbf{v}_p(t) \times \mathbf{B}\left(\mathbf{r}_p(t), t\right)\right], & p = 1, \ldots, N_m\,, \end{aligned} \tag{3.11}$$

and,

$$\begin{aligned} \nabla \times \mathbf{E}\left(\mathbf{r}, t\right) + \partial_t \mathbf{B}\left(\mathbf{r}, t\right) &= \mathbf{0}\,, \\ \epsilon_0 \nabla \cdot \mathbf{E}\left(\mathbf{r}, t\right) &= \rho(\mathbf{r}, t) = \sum_{p=1}^{N_m} q_p \beta_\gamma(\mathbf{r} - \mathbf{r}_p(t))\,, \\ -\epsilon_0 \partial_t \mathbf{E}\left(\mathbf{r}, t\right) + \tfrac{1}{\mu_0} \nabla \times \mathbf{B}\left(\mathbf{r}, t\right) &= \mathbf{j}(\mathbf{r}, t) = \sum_{p=1}^{N_m} q_p \mathbf{v}_p(t) \beta_\gamma(\mathbf{r} - \mathbf{r}_p(t))\,, \\ \nabla \cdot \mathbf{B}\left(\mathbf{r}, t\right) &= 0\,. \end{aligned} \tag{3.12}$$

Note that the sums have been transformed into sums over macro-particles. The order of the shape function and discretization methods dictate the particular flavor of PIC.

3.1.2 Discretization

In PIC, there are two simultaneous discretizations. The first we have already seen; the distribution function is assumed to take the form for a summation of macro-particles that move with Lagrangian variables, $\mathbf{r}_p$ and $\mathbf{v}_p$. The second is for Maxwell's equations, where a discrete mesh is used. We must be precise in how the particle discretization and field discretization communicate.

Consider a simple finite difference discretization of Maxwell's equations. While more sophisticated spatial discretization can be used, the arguments that follow do not change significantly. At a given mesh point, i, and time, t, the charge and current densities are well defined by

$$\begin{aligned} \rho_i(\mathbf{r}_i, t) &= \sum_{p=1}^{N_m} q_p \beta_\gamma(\mathbf{r}_i - \mathbf{r}_p(t))\,, \\ \mathbf{j}_i(\mathbf{r}_i, t) &= \sum_{p=1}^{N_m} q_p \mathbf{v}_p \beta_\gamma(\mathbf{r}_i - \mathbf{r}_p(t))\,, \end{aligned} \tag{3.13}$$

where $\mathbf{r}_i$ represents the location of mesh point i. In a finite difference setting, the case of defining a field value at a Lagrangian position, $\mathbf{r}_p$, is as not straightforward. To simplify the matter, it is assumed that the fields "reach" to the particles in a symmetric fashion to how particles are

interpolated to mesh locations in (3.13). That is, the same symmetric shape function is used to compute the values of $\mathbf{E}(\mathbf{r}_p(t),t)$ and $\mathbf{B}(\mathbf{r}_p(t),t)$ in the equations of motion. Define

$$\mathbf{E}(\mathbf{r}_p(t),t) \;=\; \sum_{i\in\mathcal{M}_p} \mathbf{E}_i(t)\hat{\beta}_\gamma(\mathbf{r}_i-\mathbf{r}_p(t))\Delta_{p,x}\Delta_{p,y}\Delta_{p,z}, \tag{3.14}$$

where

$$\mathcal{M}_p = \{i \,|\, \mathbf{r}_i \in \operatorname{supp}\beta_\gamma(\mathbf{r}-\mathbf{r}_p)\}$$

is the index of all mesh points within the support of macro-particle p. The $\Delta_{p,x}\Delta_{p,y}\Delta_{p,z}$ term is included to bring the height of β_γ back to unity. The interpolation of $\mathbf{B}(\mathbf{r}_p(t),t)$ is defined in a similar way.

All that remains is to choose a time discretization. In classical PIC, a leap-frog method is used to explicitly time step. Let the superscript k denote a variable at $t=k\delta t$. The particles are updated with

$$\begin{aligned}
\tfrac{\mathbf{r}_p^{k+1}-\mathbf{r}_p^k}{\delta t} &= \mathbf{v}_p^{k+1/2} \\
\tfrac{\mathbf{v}_p^{k+1/2}-\mathbf{v}_p^{k-1/2}}{\delta t} &= \tfrac{q_p}{m_p}\sum_i \left[\mathbf{E}_i^{k+1/2}\beta_\gamma(\mathbf{r}_i-\mathbf{r}_p^k) + \tfrac{1}{2}(\mathbf{v}_p^{k+1/2}+\mathbf{v}_p^{k-1/2})\times\mathbf{B}_i^{k+1/2}\beta_\gamma(\mathbf{r}_i-\mathbf{r}_p^k)\right],
\end{aligned} \tag{3.15}$$

and the fields are updated with

$$\begin{aligned}
&\begin{bmatrix} \frac{1}{\delta t} & 0 \\ 0 & 0 \\ 0 & \frac{-\epsilon_0}{\delta t} \\ 0 & 0 \end{bmatrix}
\left(\begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}^{k+1/2} - \begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}^{k-1/2} \right) +
\begin{bmatrix} 0 & \nabla\times \\ 0 & \nabla\cdot \\ \frac{1}{\mu_0}\nabla\times & 0 \\ \nabla\cdot & 0 \end{bmatrix}^h
\begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}^{k-1/2} \\
&\qquad = \begin{bmatrix} \mathbf{0} \\ \sum_{p=1}^{N_m} q_p\beta_\gamma(\mathbf{r}_i-\mathbf{r}_p^k) \\ \sum_{p=1}^{N_m} q_p\mathbf{v}_p^k\beta_\gamma(\mathbf{r}_i-\mathbf{r}_p^k) \\ 0 \end{bmatrix},
\end{aligned} \tag{3.16}$$

where the superscript h denotes a basic finite difference discretization. This iteration is summarized in Figure 3.2.

When PIC is integrated in this way, there a two major conditions required in order to achieve stability. The time-step, δt, must be on the order of the extremely fast plasma frequency ($\frac{2\pi}{\omega_{p,e}}$) and the mesh spacing, Δ_x, Δ_y, and Δ_z, must be on the order of the relatively small Debye length (λ_D). These restrictions make both long time scale and large physical domains very costly to simulate. Although costly, PIC captures kinetic effects that fluid models are incapable of detecting.

Recent advances in implicit PIC methods have relaxed many of the restrictions of classical PIC methods, while still capturing relevant kinetic physics [18]. To do justice to these powerful, new, implicit PIC methods, we must first delve deeper into moment-based methods.

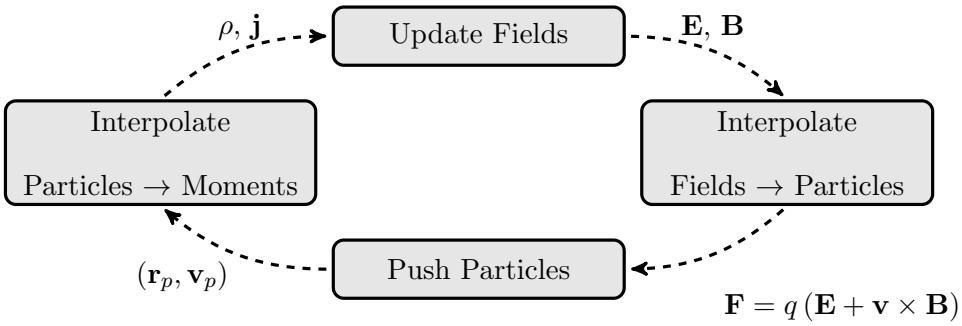

Figure 3.2: Classic PIC, where a leap-frop approach is used to first push the particles. The particle information is then accumulated to into the moments (ρ and $\mathbf{j}$). The moment information is the used to update the fields with Maxwell's equations. Finally, the field data is interpolated back to the particle positions. The process continues until a desired time-step is reached.

3.2 Moment Method

In the MB method, moments of the Vlasov equation (2.8) are taken to produce a system of fluid-like equations in number densities, n_α, and momentum densities, $\mathbf{p}_\alpha$. These quantities then couple directly back into the Maxwell system, (2.13) - (2.16), through charge and current density. This process eliminates all moment integrals, effectively casting the VM system into a purely differential fashion.

The reader should keep in mind that, although derivations for an arbitrary number of species

are provided, later, only two species, ions, i, and electrons, e, are considered. A mass-weighted averaging of the fluid equations recovers the traditional magnetohydrodynamic (MHD) equations. Instead, we focus on the system where no such averaging is performed and each species of the plasma has its own set of fluid-like equations.

3.2.1 The Zeroth Moment

We begin by examining the zeroth moment of (2.8):

$$\int_{\Omega_{\mathbf{v}}} \left[\frac{\partial f_\alpha}{\partial t} + \mathbf{v} \cdot \nabla f_\alpha + \frac{q_\alpha}{m_\alpha} \left[\mathbf{E} + \mathbf{v} \times \mathbf{B} \right] \cdot \nabla_{\mathbf{v}} f_\alpha \right] \mathrm{d}\mathbf{v} = 0 \,. \tag{3.17}$$

The first term on the left-hand side becomes

$$\int_{\Omega_{\mathbf{v}}} \frac{\partial f_\alpha}{\partial t} \,\mathrm{d}\mathbf{v} = \frac{\partial}{\partial t} \int_{\Omega_{\mathbf{v}}} f_\alpha \,\mathrm{d}\mathbf{v} = \frac{\partial n_\alpha}{\partial t} \,,$$

where pulling a ∂_t through the integral is justified because the velocity domain, $\Omega_{\mathbf{v}}$, is independent of time. The second term reduces to

$$\int_{\Omega_{\mathbf{v}}} \mathbf{v} \cdot \nabla f_\alpha \,\mathrm{d}\mathbf{v} = \int_{\Omega_{\mathbf{v}}} \nabla \cdot (\mathbf{v} f_\alpha) \,\mathrm{d}\mathbf{v} = \nabla \cdot \int_{\Omega_{\mathbf{v}}} \mathbf{v} f_\alpha \,\mathrm{d}\mathbf{v} = \nabla \cdot \mathbf{p}_\alpha \,,$$

where the identity, $\mathbf{v} \cdot \nabla f_\alpha = \nabla \cdot (\mathbf{v} f_\alpha)$, is realized by remembering that, in phase-space, $\mathbf{v}$ is considered to be an independent variable. The divergence can be pulled through the velocity integral for a similar reason.

In the final term, let $\mathbf{F} = \frac{q_\alpha}{m_\alpha} \left[\mathbf{E} + \mathbf{v} \times \mathbf{B} \right]$, and let $V(\nu)$ be the volume of a sphere of radius ν in velocity space centered at the origin. Take ν large enough such that $\Omega_{\mathbf{v}} \subset V(\nu)$. Then it holds that

$$\int_{\Omega_{\mathbf{v}}} \mathbf{F} \cdot \nabla_{\mathbf{v}} f_\alpha \,\mathrm{d}\mathbf{v} = \int_{V(\nu)} \mathbf{F} \cdot \nabla_{\mathbf{v}} f_\alpha \,\mathrm{d}\mathbf{v} = \int_{V(\nu)} \nabla_{\mathbf{v}} \cdot (\mathbf{F} \cdot f_\alpha) \,\mathrm{d}\mathbf{v} = \int_{\partial V(\nu)} (\mathbf{F} \cdot f_\alpha) \cdot \,\mathrm{d}\sigma = 0 \,,$$

where the last integral vanishes due to the velocity support of the distribution function being contained entirely in $V(\nu)$. The above chain of equalities uses the fact that

$$\nabla_{\mathbf{v}} \cdot (\mathbf{E} + \mathbf{v} \times \mathbf{B}) = \nabla_{\mathbf{v}} \cdot \mathbf{E} + \mathbf{B} \cdot \nabla_{\mathbf{v}} \times \mathbf{v} - \mathbf{v} \cdot \nabla_{\mathbf{v}} \times \mathbf{B} = \mathbf{0} \,.$$

Collecting the above expansions gives the zeroth moment of (2.7):

$$\partial_t n_\alpha(\mathbf{r},t) + \nabla \cdot \mathbf{p}_\alpha(\mathbf{r},t) = 0, \qquad \alpha = 1,\ldots,S\,. \tag{3.18}$$

This equation is referred to as the continuity equation. It states that the number density of species, α, is changing in time can only be due to a flux in the momentum density.

3.2.2 The First Moment

The first moment of (2.7) is

$$\int_{\Omega_\mathbf{v}} \mathbf{v}\left[\frac{\partial f_\alpha}{\partial t} + \mathbf{v}\cdot\nabla f_\alpha + \frac{q_\alpha}{m_\alpha}\left[\mathbf{E}+\mathbf{v}\times\mathbf{B}\right]\cdot\nabla_\mathbf{v} f_\alpha\right]\,\mathrm{d}\mathbf{v} = \mathbf{0}\,. \tag{3.19}$$

Starting with the first term, we get

$$\int_{\Omega_\mathbf{v}} \mathbf{v}\frac{\partial f_\alpha}{\partial t}\,\mathrm{d}\mathbf{v} = \frac{\partial}{\partial t}\int_{\Omega_\mathbf{v}} \mathbf{v} f_\alpha\,\mathrm{d}\mathbf{v} = \frac{\partial \mathbf{p}_\alpha}{\partial t}\,,$$

where a time derivative was pulled through both $\mathbf{v}$ and the integral because neither are time dependent.

The second term contains the product $\mathbf{v}^2$, which should be interpreted as the outer product $\mathbf{v}\mathbf{v}^T$, as detailed in Section 2.1. Again, using tricks similar to the second term in the continuity derivation,

$$\int_{\Omega_\mathbf{v}} \mathbf{v}\mathbf{v}^T\cdot\nabla f_\alpha\,\mathrm{d}\mathbf{v} = \int_{\Omega_\mathbf{v}} \nabla\cdot(\mathbf{v}\mathbf{v}^T f_\alpha)\,\mathrm{d}\mathbf{v} = \nabla\cdot\int_{\Omega_\mathbf{v}} \mathbf{v}\mathbf{v}^T f_\alpha\,\mathrm{d}\mathbf{v} = \nabla\cdot\underline{\mathbf{S}}_\alpha\,.$$

The last term on the left-hand side is subtle and special care must be taken. The integral reduces to

$$\frac{q_\alpha}{m_\alpha}\int_{\Omega_\mathbf{v}} (\mathbf{v}\mathbf{F}^T)\cdot\nabla_\mathbf{v} f_\alpha\,\mathrm{d}\mathbf{v} = \frac{q_\alpha}{m_\alpha}\int_{V(\nu)} (\mathbf{v}\mathbf{F}^T)\cdot\nabla_\mathbf{v} f_\alpha\,\mathrm{d}\mathbf{v}\,,$$

with $\mathbf{F} = \mathbf{E}+\mathbf{v}\times\mathbf{B}$ and $V(\nu)$ as described previously. Rearranging the integrand gives

$$\frac{q_\alpha}{m_\alpha}\int_{V(\nu)} (\mathbf{v}\mathbf{F}^T)\cdot\nabla_\mathbf{v} f_\alpha\,\mathrm{d}\mathbf{v} = \frac{q_\alpha}{m_\alpha}\int_{V(\nu)} \nabla_\mathbf{v}\cdot(\mathbf{v}\mathbf{F}^T f_\alpha)\,\mathrm{d}\mathbf{v} - \frac{q_\alpha}{m_\alpha}\int_{V(\nu)} \nabla_\mathbf{v}\cdot(\mathbf{v}\mathbf{F}^T) f_\alpha\,\mathrm{d}\mathbf{v}\,.$$

Using the divergence theorem and the fact that f_α vanishes on the boundary of $V(\nu)$, the first term on the right is zero. To reduce the second term on the right, observe that

$$\mathbf{v}\mathbf{F}^T = \begin{pmatrix} u\left(-wB_y + vB_z + E_x\right) & u\left(wB_x - uB_z + E_y\right) & u\left(-vB_x + uB_y + E_z\right) \\ v\left(-wB_y + vB_z + E_x\right) & v\left(wB_x - uB_z + E_y\right) & v\left(-vB_x + uB_y + E_z\right) \\ w\left(-wB_y + vB_z + E_x\right) & w\left(wB_x - uB_z + E_y\right) & w\left(-vB_x + uB_y + E_z\right) \end{pmatrix},$$

and take the velocity divergence of each row in order to conclude that

$$\nabla_{\mathbf{v}} \cdot (\mathbf{v}^T\mathbf{F}^T) = \begin{pmatrix} \left(-wB_y + vB_z + E_x\right) \\ \left(wB_x - uB_z + E_y\right) \\ \left(-vB_x + uB_y + E_z\right) \end{pmatrix} = \mathbf{F}\,.$$

Using the above identity, the forcing term reduces as follows:

$$-\frac{q_\alpha}{m_\alpha}\int\limits_{V(\nu)} \nabla_{\mathbf{v}} \cdot (\mathbf{v}\mathbf{F}^T) f_\alpha \,\mathrm{d}\mathbf{v} = -\frac{q_\alpha}{m_\alpha}\int\limits_{V(\nu)} \mathbf{F} f_\alpha \,\mathrm{d}\mathbf{v} = -\frac{q_\alpha}{m_\alpha} n_\alpha \mathbf{E} - \frac{q_\alpha}{m_\alpha}\mathbf{p}_\alpha \times \mathbf{B}\,.$$

Collecting all of the above simplifications gives the conservation of momentum equation:

$$\partial_t \mathbf{p}_\alpha + \nabla \cdot \underline{\mathbf{S}}_\alpha - \frac{q_\alpha}{m_\alpha} n_\alpha \mathbf{E} - \frac{q_\alpha}{m_\alpha}\mathbf{p}_\alpha \times \mathbf{B} = \mathbf{0} \qquad \alpha = 1, \ldots, S\,. \tag{3.20}$$

3.2.3 Closure

The observant reader would notice that, due to the $\mathbf{v} \cdot \nabla f_\alpha$ term in (2.8), each moment equation produces a variable of the next higher moment of the distribution function. The zeroth moment produced an expression in n_α and $\mathbf{p}_\alpha$. The first moment produced an expression in n_α, $\mathbf{p}_\alpha$, and $\underline{\mathbf{S}}_\alpha$. The pattern continues in this way: each moment equation adds an additional unknown variable. At some point, an equation of closure is required. That is, a function, $\mathcal{C}$, that relates the highest moment to some or all of the lower moments:

$$\psi_\alpha^n = \mathcal{C}(\psi_\alpha^0, \ldots, \psi_\alpha^{n-1})\,. \tag{3.21}$$

A closure is supplied in the stress tensor, $\underline{\mathbf{S}}_\alpha$. To do so, it is convenient to expand the stress tensor into two parts. This is done by defining the pressure tensor as

$$\underline{\mathbf{P}}_\alpha = \int\limits_{\Omega_{\mathbf{v}}} (\mathbf{v} - \mathbf{u}_\alpha)(\mathbf{v} - \mathbf{u}_\alpha)^T \,\mathrm{d}\mathbf{v}\,. \tag{3.22}$$

The pressure tensor is computed in the same way as the stress tensor, but the velocities are shifted into the fluid frame of reference. Using this definition, the original stress tensor is expanded as

$$\underline{\mathbf{S}}_\alpha = \underline{\mathbf{P}}_\alpha + \frac{\mathbf{p}_\alpha \mathbf{p}_\alpha^T}{n_\alpha} \,. \tag{3.23}$$

Substituting the expanded definition of the stress tensor into (3.20) gives

$$\partial_t \mathbf{p}_\alpha + \nabla \cdot \left(\frac{\mathbf{p}_\alpha \mathbf{p}_\alpha^T}{n_\alpha} \right) + \nabla \cdot \underline{\mathbf{P}}_\alpha - \frac{q_\alpha}{m_\alpha} n_\alpha \mathbf{E} - \frac{q_\alpha}{m_\alpha} \mathbf{p}_\alpha \times \mathbf{B} = \mathbf{0} \quad \alpha = 1, \ldots, S \,. \tag{3.24}$$

Closures are often supplied to the pressure tensor instead of to the stress tensor. Many possible closures exist for the pressure tensor; we provide three simple descriptions (see [26] for a detailed discussion of each).

Cold Plasma In the cold plasma assumption, the temperature of the plasma is not large enough for the constituent of particles to produce pressure effects. The cold plasma closure is simply

$$\underline{\mathbf{P}}_\alpha = 0 \,. \tag{3.25}$$

Adiabatic Plasma The next level of complexity arises by assuming the plasma pressure behaves similar to that of an ideal gas, that is, that the pressure effects are isotropic:

$$\underline{\mathbf{P}}_\alpha = \underline{\mathbf{1}} P_\alpha \,,$$

where $\underline{\mathbf{1}}$ is the identity tensor and P_α is a scalar pressure that obeys

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{P_\alpha}{n_\alpha^\gamma} \right) = 0 \,.$$

Here, γ is well known from thermodynamics to be the heat capacity ratio. The simplest form of this relation is known as the isothermal ($\gamma = 1$) approximation,

$$P_\alpha = \frac{\kappa_B T_\alpha}{m_\alpha} n_\alpha = \frac{1}{2} v_{th_\alpha}^2 n_\alpha \,, \tag{3.26}$$

where κ_B is Boltzmann's constant and v_{th_α} is the thermal velocity.

Chew-Goldberger-Low In the Chew-Goldberger-Low model, different adiabatic assumptions are made in the direction parallel ($\|$) and perpendicular ($\perp$) to the magnetic field:

$$\hat{\underline{\mathbf{P}}}_\alpha = \begin{bmatrix} P_{s\perp} & 0 & 0 \\ 0 & P_{s\perp} & 0 \\ 0 & 0 & P_{s\|} \end{bmatrix},$$

where $\hat{\underline{\mathbf{P}}}_\alpha$ is written in a local coordinate system relative to the magnetic field. A local transformation of coordinates can be used to then find $\underline{\mathbf{P}}_\alpha$.

3.2.4 Multi-Fluid Plasma Equations

At this point, we pause and collect all of the above moment integral derivations and couple them to (2.13) - (2.16):

$$\begin{aligned} \partial_t \mathbf{p}_\alpha + \nabla \cdot \left(\frac{\mathbf{p}_\alpha \mathbf{p}_\alpha^T}{n_\alpha} \right) + \frac{T_\alpha}{m_\alpha} \nabla n_\alpha - \frac{q_\alpha}{m_\alpha} n_\alpha \mathbf{E} - \frac{q_\alpha}{m_\alpha} \mathbf{p}_\alpha \times \mathbf{B} &= \mathbf{0}, \quad \alpha = 1, \ldots, S, \\ \partial_t n_\alpha + \nabla \cdot \mathbf{p}_\alpha &= 0, \quad \alpha = 1, \ldots, S, \\ \nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} &= \mathbf{0}, \\ \epsilon_0 \nabla \cdot \mathbf{E} - \sum_{\alpha=1}^{S} q_\alpha n_\alpha &= 0, \\ -\epsilon_0 \frac{\partial \mathbf{E}}{\partial t} + \frac{1}{\mu_0} \nabla \times \mathbf{B} - \sum_{\alpha=1}^{S} q_\alpha \mathbf{p}_\alpha &= \mathbf{0}, \\ \nabla \cdot \mathbf{B} &= 0, \end{aligned} \tag{3.27}$$

where an isothermal closure was supplied to eliminate $\underline{\mathbf{P}}_\alpha$ from the system ($\nabla \cdot \underline{\mathbf{P}}_\alpha = \frac{T_\alpha}{m_\alpha} \nabla n_\alpha$). It should be noted that Boltzmann's constant, κ_B, is absorbed into the temperatures, T_α. The system of equations, (3.27), is labeled as the multi-fluid plasma (MFP) equations.

The VM system, (2.19), has been reformulated into a first-order PDE with unknowns in n_α, $\mathbf{p}_\alpha$ (for $\alpha = 1, \ldots, S$), $\mathbf{E}$, and $\mathbf{B}$. The phase space velocity dependence has been removed from the system, making all unknowns functions depend only on $\mathbf{r}$ and t. We soon turn focus to the analysis of the two-fluid plasma (TFP) system with $\alpha = i, e$ (ions and electrons). The main goal of this thesis is to explore the viability of an NI-FOSLS-based solver applied to the TFP equations. The above system is considered in an entirely standalone context, but the results have important

implications towards preconditioning a recently developed implicit PIC method [15]. In the next section, an overview of this new kinetic-based implicit method is provided, paying special attention to where the inversion of a discretization of (3.27) is important.

3.3 Implicit Particle-in-Cell

Recent developments towards a fully implicit electromagnetic PIC method have been made [15, 19, 20]. This new version of PIC has the very desirable property of being fully implicit and charge- and energy-conserving, and the potential to be GPU accelerated. The nonlinear, fully implicit nature of the algorithm requires the use of a JFNK solver (described in Appendix A.1) to converge the macro-particle system and Maxwell model simultaneously. Central to the JFNK approach is a nonlinear residual calculation. At each time step, the JFNK solver converges the current field values and particle states to a tight tolerance. This new implicit approach relaxes many of the restrictions that explicit PIC methods put on mesh spacing and time step size.

In this section, only an overview of the approach is provided to motivate the need for an MB preconditioner. We are lax in notation, and neglect to rigorously define shape function orders, discretization schemes, moment accumulation details, and particle sub-stepping, all of which are important to the success of the method [18]. The reader should come away with a rough idea of the main components that make a kinetic-JFNK solver tick.

A slight modification is made to (3.15) and (3.16) to make the system fully implicit. We modify the particle evolution equations to be

$$\begin{aligned} \frac{\mathbf{r}_p^{k+1}-\mathbf{r}_p^k}{\delta t} &= \frac{\mathbf{v}_p^{k+1}+\mathbf{v}_p^k}{2} \\ \frac{\mathbf{v}_p^{k+1}-\mathbf{v}_p^k}{\delta t} &= \frac{q_p}{m_p}\sum_i \left[\frac{\mathbf{E}_i^k+\mathbf{E}_i^{k+1}}{2}\hat{\beta}_\gamma(\mathbf{r}_i-\frac{\mathbf{r}_p^k+\mathbf{r}_p^{k+1}}{2})\right] \\ &+ \frac{q_p}{m_p}\sum_i \left[\frac{\mathbf{v}_p^k+\mathbf{v}_p^{k+1}}{2}\times\frac{\mathbf{B}_i^k+\mathbf{B}_i^{k+1}}{2}\hat{\beta}_\gamma(\mathbf{r}_i-\frac{\mathbf{r}_p^k+\mathbf{r}_p^{k+1}}{2})\right]. \end{aligned} \tag{3.28}$$

Given the vector of particle states, $\mathbf{r}^k = \left[\mathbf{r}_1^k, \ldots, \mathbf{r}_{N_m}^k\right]$ and $\mathbf{v}^k = \left[\mathbf{v}_1^k, \ldots, \mathbf{v}_{N_m}^k\right]$, the field values $\mathbf{E}^k$ and $\mathbf{B}^k$, and a current approximation of $\mathbf{E}^{k+1}$ and $\mathbf{B}^{k+1}$, define the forward solution of (3.28) for

particle states at $k+1$ as

$$\mathbf{Q}^{k+1} = \mathcal{E}\left(\mathbf{Z}^{k+1}\right) = \mathcal{E}\left(\mathbf{Z}^{k+1}\,;\,\mathbf{Q}^{k},\mathbf{Z}^{k}\right)\,, \tag{3.29}$$

where $\mathbf{Q}^k = \left[\mathbf{r}^k, \mathbf{v}^k\right]^T$ and $\mathbf{Z}^k = \left[\mathbf{E}^k, \mathbf{B}^k\right]^T$. System (3.28) is solved rapidly for each particle via a simple iterative method, such as Picard. This is where the use of modern computer architectures, such as GPUs, can be exploited to accelerate the above forward solution.

Equation (3.16) is modified in a similar fashion:

$$\begin{aligned}\begin{bmatrix} \frac{1}{\delta t} & 0 \\ 0 & 0 \\ 0 & \frac{-\epsilon_0}{\delta t} \\ 0 & 0 \end{bmatrix} \left(\begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}_i^{k+1} - \begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}_i^{k} \right) + \tfrac{1}{2} \begin{bmatrix} 0 & \nabla\times \\ 0 & \nabla\cdot \\ \frac{1}{\mu_0}\nabla\times & 0 \\ \nabla\cdot & 0 \end{bmatrix}^h \left(\begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}^{k} + \begin{bmatrix} \mathbf{B} \\ \mathbf{E} \end{bmatrix}^{k+1} \right) \\ - \begin{bmatrix} 0 \\ \sum_{p=1}^{N_m} q_p \beta_\gamma(\mathbf{r}_i - \frac{\mathbf{r}_p^k + \mathbf{r}_p^{k+1}}{2}) \\ \sum_{p=1}^{N_m} q_p \frac{\mathbf{v}_p^k + \mathbf{v}_p^{k+1}}{2} \beta_\gamma(\mathbf{r}_i - \frac{\mathbf{r}_p^k + \mathbf{r}_p^{k+1}}{2}) \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ 0 \\ \mathbf{0} \\ 0 \end{bmatrix}\,. \end{aligned} \tag{3.30}$$

Consider the case where $\mathbf{Q}^k$ and $\mathbf{Z}^k$ are known. Then, for given approximations, $\mathbf{Z}^{k+1}$ and $\mathbf{Q}^{k+1}$, define the residual of (3.30) as

$$G\left(\mathbf{Z}^{k+1}, \mathbf{Q}^{k+1}\right)\,, \tag{3.31}$$

where the dependence upon $\mathbf{Q}^k$ and $\mathbf{Z}^k$ is assumed. Observe that the residual can be evaluated given only the previous values of $\mathbf{Z}^k$ and $\mathbf{Q}^k$ and a current guess for $\mathbf{Z}^{k+1}$. That is, $\mathbf{Q}^{k+1}$ can be acquired via (3.29):

$$G\left(\mathbf{Z}^{k+1}\right) \equiv G\left(\mathbf{Z}^{k+1}, \mathcal{E}\left(\mathbf{Z}^{k+1}\right)\right)\,. \tag{3.32}$$

It should be clearly stated that every call to the residual function, $G(\mathbf{Z}^{k+1})$, implicitly requires a call to $\mathcal{E}(\mathbf{Z}^{k+1})$. The GPU architecture is well tailored to tackle these types of particle integration.

The nature of GPU memory structure allows for both the particle integration and accumulation in (3.30) to be carried out efficiently [43]. The particle integration step is be carried out by

all GPU threads simultaneously and further accelerated by storing electromagnetic field data using the very fast read-only texture memory. The accumulation of particle states onto mesh points is performed in a hierarchical way, exploiting the local and shared GPU memory spaces to reduce the thread contention when writing back into the global memory space [21]. But, even with advances in GPU computing, heterogeneous architectures, and clever implementations [19], the cost of $\mathcal{E}\left(\mathbf{Z}^{k+1}\right)$ remains costly.

Nevertheless, the dimension of the problem has been reduced to a more manageable size, bringing the application of a Newton-type method back into the realm of possibilities. Unfortunately, due to the complicated evolution operator, $\mathcal{E}$, which, within itself, involves an iterative method to converge, the explicit formation of the Jacobian of $G(\mathbf{Z}^{k+1})$ is nearly impossible. To bypass this, a Jacobian-Free Newton Krylov (JFNK) technique is employed. In short, the JFNK approach (see Appendix A.1) forms a Krylov subspace for an approximation of the Jacobian by successive application of the function G. Applying a preconditioner to the JFNK iteration can greatly reduce the dimension of the Krylov subspace and, in turn, the number of expensive GPU function evaluations.

A recent study demonstrates the effectiveness of using an electrostatic MB two-fluid plasma (TFP) model to precondition a JFNK-based kinetic solver [20]. The electrostatic MB model provides an inexpensive and easily inverted approximation to the electrostatic kinetic Jacobian. This enables the use of larger implicit time steps, while bounding the number of Krylov iterations independent of the ion-to-electron mass ratio. Speedups of approximately three orders of magnitude versus explicit PIC were demonstrated.

Efforts are currently being directed towards a fully implicit, electromagnetic, JFNK-based kinetic solver, motivating the necessity of developing a suitable MB electromagnetic preconditioning strategy [15]. Currently, the use of a physics-based preconditioners are used to fill this roll [17]. We, instead, explore the more mathematically rigorous alternative of using a First-Order System Least-Squares (FOSLS) discretization to approximately invert the full electromagnetic (Darwin) two-fluid plasma system as a preconditioner.

3.4 Two-Fluid Plasma Model As a Preconditioner

To use the TFP-Darwin system as a preconditioner, defined as P^{-1}, for the kinetic JFNK solver, we must define the process of applying such a system to the kinetic residual. When a preconditioned JFNK method is applied to (3.32), a Krylov subspace is built for the linear system

$$JP^{-1}\delta\mathbf{W} = -G(\mathbf{Z}^{k+1})\,, \tag{3.33}$$

where J is the approximate Jacobian of G. In the formulation of the Krylov subspace, P^{-1} must be applied to a residual vector, $\mathbf{R}$, and return an update, $\delta\mathbf{Z}$.

The TFP-Darwin system is

$$\begin{aligned}
\partial_t \mathbf{p}_i + \nabla\cdot\underline{\mathbf{S}}_i - \tfrac{q_i}{m_i} n_i\mathbf{E} - \tfrac{q_i}{m_i}\mathbf{p}_i\times\mathbf{B} &= \mathbf{0}\,,\\
\partial_t n_i + \nabla\cdot\mathbf{p}_i &= 0\,,\\
\partial_t \mathbf{p}_e + \nabla\cdot\underline{\mathbf{S}}_e - \tfrac{q_e}{m_e} n_e\mathbf{E} - \tfrac{q_e}{m_e}\mathbf{p}_e\times\mathbf{B} &= \mathbf{0}\,,\\
\partial_t n_e + \nabla\cdot\mathbf{p}_e &= 0\,,\\
\nabla\times\mathbf{E}_r &= \mathbf{0}\,,\\
-q_i n_i - q_e n_e + \epsilon_0\nabla\cdot\mathbf{E}_r &= 0\,,\\
-q_i\mathbf{p}_i - q_e\mathbf{p}_e - \epsilon_0\partial_t\mathbf{E}_r + \tfrac{1}{\mu_0}\nabla\times\mathbf{B} &= \mathbf{0}\,,\\
\nabla\cdot\mathbf{B} &= 0\,,\\
\nabla\times\mathbf{E}_s + \partial_t\mathbf{B} &= \mathbf{0}\,,\\
\nabla\cdot\mathbf{E}_s &= 0\,,
\end{aligned} \tag{3.34}$$

where the closures for $\underline{\mathbf{S}}_\alpha$ are outlined in Section 3.2.3. To see how the TFP system can be used as

a preconditioner, consider a simple implicit time discretization:

$$\begin{aligned}
\tfrac{1}{\delta t}\mathbf{p}_i + \nabla\cdot\underline{\mathbf{S}}_i - \tfrac{q_i}{m_i}n_i\mathbf{E} - \tfrac{q_i}{m_i}\mathbf{p}_i\times\mathbf{B} + \{\ldots\} &= \mathbf{0}\,,\\
\tfrac{1}{\delta t}n_i + \nabla\cdot\mathbf{p}_i + \{\ldots\} &= 0\,,\\
\tfrac{1}{\delta t}\mathbf{p}_e + \nabla\cdot\underline{\mathbf{S}}_e - \tfrac{q_e}{m_e}n_e\mathbf{E} - \tfrac{q_e}{m_e}\mathbf{p}_e\times\mathbf{B} + \{\ldots\} &= \mathbf{0}\,,\\
\tfrac{1}{\delta t}n_e + \nabla\cdot\mathbf{p}_e + \{\ldots\} &= 0\,,\\
\nabla\times\mathbf{E}_r &= \mathbf{0}\,,\\
-q_in_i - q_en_e + \epsilon_0\nabla\cdot\mathbf{E}_r &= 0\,,\\
-q_i\mathbf{p}_i - q_e\mathbf{p}_e - \epsilon_0\tfrac{1}{\delta t}\mathbf{E}_r + \tfrac{1}{\mu_0}\nabla\times\mathbf{B} + \{\ldots\} &= \mathbf{0}\,,\\
\nabla\cdot\mathbf{B} &= 0\,,\\
\nabla\times\mathbf{E}_s + \tfrac{1}{\delta t}\mathbf{B} + \{\ldots\} &= \mathbf{0}\,,\\
\nabla\cdot\mathbf{E}_s &= 0\,,
\end{aligned} \tag{3.35}$$

where the notation $\{\ldots\}$ contains all previous known time step information, and all unknowns are taken to be at the current time step. Linearizing the system gives

$$\begin{aligned}
\tfrac{1}{\delta t}\delta\mathbf{p}_i + \nabla\cdot\delta\underline{\mathbf{S}}_i - \tfrac{q_i}{m_i}n_{i,0}\delta\mathbf{E} - \tfrac{q_i}{m_i}\mathbf{E}_0\delta n_i - \tfrac{q_i}{m_i}\mathbf{p}_{i,0}\times\delta\mathbf{B} + \tfrac{q_i}{m_i}\mathbf{B}_0\times\delta\mathbf{p}_{i,0} &= -\mathbf{R}_i\,,\\
\tfrac{1}{\delta t}\delta n_i + \nabla\cdot\delta\mathbf{p}_i &= -R_i\,,\\
\tfrac{1}{\delta t}\delta\mathbf{p}_e + \nabla\cdot\delta\underline{\mathbf{S}}_e - \tfrac{q_e}{m_e}n_{e,0}\delta\mathbf{E} - \tfrac{q_e}{m_e}\mathbf{E}_0\delta n_e - \tfrac{q_e}{m_e}\mathbf{p}_{e,0}\times\delta\mathbf{B} - \tfrac{q_e}{m_e}\mathbf{B}_0\times\delta\mathbf{p}_{e,0} &= -\mathbf{R}_e\,,\\
\tfrac{1}{\delta t}\delta n_e + \nabla\cdot\delta\mathbf{p}_e &= -R_e\,,\\
\nabla\times\delta\mathbf{E}_r &= -\mathbf{R}_{\mathbf{E}_r}\,,\\
-q_i\delta n_i - q_e\delta n_e + \epsilon_0\nabla\cdot\delta\mathbf{E}_r &= -R_{\mathbf{E}_r}\,,\\
-q_i\delta\mathbf{p}_i - q_e\delta\mathbf{p}_e - \epsilon_0\tfrac{1}{\delta t}\delta\mathbf{E}_r + \tfrac{1}{\mu_0}\nabla\times\delta\mathbf{B} &= -\mathbf{R}_{\mathbf{B}}\,,\\
\nabla\cdot\delta\mathbf{B} &= -R_{\mathbf{B}}\,,\\
\nabla\times\delta\mathbf{E}_s + \tfrac{1}{\delta t}\mathbf{B} &= -\mathbf{R}_{\mathbf{E}_s}\,,\\
\nabla\cdot\delta\mathbf{E}_s &= -R_{\mathbf{E}_s}\,,
\end{aligned} \tag{3.36}$$

where subscript 0 denotes the current linearization and the right-hand sides are the negative residual of (3.35). Observe that the residuals are computable given only a current value of $\mathbf{Z}^{k+1}$. The logic follows from the same reasoning used in (3.32); moment variables are computable via forward particle solve. The preconditioning process of applying P^{-1} is carried out as follows:

(1) Given the current field values, $\mathbf{Z}^{k+1}$, and particle states, $\mathbf{Q}^{k+1}$, compute the residual of (3.34).

(2) Solve the system of equations (3.36) for updates.

(3) Return the update to $\delta\mathbf{Z}$, discarding the fluid variable updates.

(4) Proceed with the JFNK step, calling the particle evolution operator with the preconditioned values of $\mathbf{Z}^{k+1}$.

The fluid equations included in the TFP system act as a coarse approximation to the particle system. The solution process adjusts the field values accordingly. This, in turn, greatly reduces the number of Krylov iterations needed, shifting much of the load onto the MB preconditioner.

3.4.1 Kinetic Stress Tensor Closure

Until now, we have shrugged off the stress tensors that appear in (3.34) for the prconditioning process. There are several ways to address this. The stress tensor could be expanded into the convective derivative plus pressure tensor, as was carried out in Section 3.2.3:

$$\nabla \cdot \underline{\mathbf{S}}_\alpha = \nabla \cdot \underline{\mathbf{P}}_\alpha + \nabla \cdot \frac{\mathbf{p}_\alpha \mathbf{p}_\alpha^T}{n_\alpha} .$$

Upon linearization, the convective derivative is eliminated and a closure is supplied in pressure. A second option is to use the particle states to accumulate a density normalized stress tensor [49]:

$$\hat{\underline{\mathbf{S}}}_\alpha = \frac{\underline{\mathbf{S}}_\alpha}{n_\alpha} .$$

The normalized stress tensor is assumed be known for any given Krylov step, as it is entirely supplied by the particles. In this way, the particles are used to close the system. This modification transforms the stress tensor as

$$\nabla \cdot \underline{\mathbf{S}}_\alpha = \nabla \cdot \left(\hat{\underline{\mathbf{S}}}_\alpha n_\alpha \right) , \tag{3.37}$$

where n_α is the fluid unknown. Focus is now given to the process of solving such TFP systems in a stand alone context. All findings that are made can be directly applied to inversion of system (3.36).

3.5 Standalone Two-Fluid Plasma

To fully analyze the TFP-Darwin system, the MB fluid method is cast into a standalone context, independent of a kinetic PIC solver. The final TFP-Darwin system we consider is

$$\begin{aligned}
\partial_t \mathbf{p}_i + \tfrac{T_i}{m_i}\nabla n_i - \tfrac{q_i}{m_i} n_i \mathbf{E} - \tfrac{q_i}{m_i}\mathbf{p}_i \times \mathbf{B} &= -\nabla \cdot \left(\tfrac{\mathbf{p}_i \mathbf{p}_i^T}{n_i}\right), \\
\partial_t n_i + \nabla \cdot \mathbf{p}_i &= 0, \\
\partial_t \mathbf{p}_e + \tfrac{T_e}{m_e}\nabla n_e - \tfrac{q_e}{m_e} n_e \mathbf{E} - \tfrac{q_e}{m_e}\mathbf{p}_e \times \mathbf{B} &= -\nabla \cdot \left(\tfrac{\mathbf{p}_i \mathbf{p}_e^T}{n_e}\right), \\
\partial_t n_e + \nabla \cdot \mathbf{p}_e &= 0, \\
\nabla \times \mathbf{E}_r &= 0, \\
-\tfrac{q_i}{\epsilon_0} n_i - \tfrac{q_e}{\epsilon_0} n_e + \nabla \cdot \mathbf{E}_r &= 0, \\
-q_i \mu_0 \mathbf{p}_i - q_e \mu_0 \mathbf{p}_e - \epsilon_0 \mu_0 \partial_t \mathbf{E}_r + \nabla \times \mathbf{B} &= 0, \\
\nabla \cdot \mathbf{B} &= 0, \\
\nabla \times \mathbf{E}_s + \partial_t \mathbf{B} &= 0, \\
\nabla \cdot \mathbf{E}_s &= 0,
\end{aligned} \tag{3.38}$$

where an isothermal closure is used for the pressure tensor, (3.26). The convective derivatives that appear on the right-hand side are considered optional and, if included, are taken as known source terms, lagged in either time or mesh. Pure conductive boundary conditions are prescribed:

$$\begin{aligned}
\hat{\mathbf{n}} \cdot \nabla n_\alpha &= 0, \\
\mathbf{p}_\alpha &= \mathbf{0}, \\
\hat{\mathbf{n}} \times (\mathbf{E}_r + \mathbf{E}_s) &= \mathbf{0}, \\
\hat{\mathbf{n}} \cdot \mathbf{B} &= 0,
\end{aligned} \tag{3.39}$$

where $\hat{\mathbf{n}}$ denotes the outward boundary normal. To make system (3.38) well posed, stronger boundary conditions are imposed. The additive constraint on electric field is decomposed into separate constraints for both the irrotational and solenoidal components:

$$\begin{aligned}
\hat{\mathbf{n}} \times \mathbf{E}_r &= \mathbf{0}, \\
\hat{\mathbf{n}} \times \mathbf{E}_s &= \mathbf{0}.
\end{aligned} \tag{3.40}$$

The decomposed boundary condition is justified by realizing that the irrotational electric field must generated from an electrostatic potential, $\mathbf{E}_r = -\nabla\Phi$. The electrostatic potential is constant on the boundary and, thus, $\nabla\Phi \perp \hat{\mathbf{n}}$ on the boundary. It directly follows from (3.39) that $\hat{\mathbf{n}} \times \mathbf{E}_s = \mathbf{0}$ on the boundary as well. The problem will be solved on a simple two-dimensional domain and, thus, differential operators are projected into their two-dimensional counter parts as outlined in Appendix A.2.

The viability of the system as a standalone, nonlinear fluid solver in the context of NI-FOSLS is the focus of the remainder of the thesis. When used in the context of a kinetic-JFNK iteration within a PIC method, a linearization of (3.38) well approximates the kinetic-Jacobian [15, 20]. In other words, the above TFP system captures the required physics necessary to supply the kinetic JFNK approach with a robust preconditioner. A direct coupling to a kinetic-based simulation is out of the scope of this thesis and is reserved for future work. See Chapter 7 for a discussion of potential pitfalls.

Chapter 4

Numerical Methods

There are four main ingredients that work together to construct a solution technique for the TFP-Darwrin system: First-Order System Least Squares (FOSLS), Nested-Iteration (NI), Adaptive Mesh Refinment (AMR), and Algebraic Multigrid (AMG). The FOSLS methodology guides the discretization and linearization process, whereby the choice of a finite element basis gives rise to a discrete linear system. The NI process uses a sequence of nested meshes to reduce the overall computational cost of the nonlinear iterations [1]. An AMR algorithm is used during NI to adaptively select the next finer mesh. The process works by minimizing the overal accuracy per computational cost (ACE) [50]. Finally, each linear system that is generated during the solution process is tackled with a conjugate-gradient-accelerated AMG solver. All of these components working together are shown in Chapter 6 to solve the TFP-Darwin system using only a handful of work units on the finest mesh.

4.1 First-Order System Least Squares

The First-Order System Least-Squares (FOSLS) finite element methodology was initially applied to general convection-diffusion problems [10]. Since then, the approach has been successfully applied to a large variety of problems such as Navier-Stokes, Maxwell's equations, neutron transport, and, most relevantly, incompressible resistive MHD [2, 11, 39, 42].

The FOSLS approach boasts many advantages over its Galerkin counterpart. Namely, it avoids the constraint of choosing finite element spaces that must conform with the well-known

inf-sup condition of Ladyzhenskaya-Babuška-Brezzi (LBB condition). In many cases, the resulting weak formulation admits a bilinear form that, if $\mathcal{H}^1$ elliptic, can be used with standard finite element spaces to generate a discrete system amenable to multigrid methods. Moreover, since the weak form is derived from a minimization, the resulting linear system is symmetric positive definite (SPD). Finally, the FOSLS functional serves as an excellent local error indicator for adaptive mesh refinement (AMR).

Of course, nothing comes for free. Unless the system being considered is naturally first-order, more dependent variables must be introduced into the system to remove higher order operators. For example, a FOSLS formulation of the convection-diffusion equation requires four independent variables, as compared with one for the second-order scalar system. Often, the second-order systems under consideration are a result of reducing naturally first-order physics systems, and the application of FOSLS can restore the unknowns to their primitive forms. Solutions to FOSLS formulations do not exactly satisfy conservation equations. Instead, the equations are only minimized in an L^2 sense. Special methods can be used to project the solution back onto a conservative solution space [28, 29]. For example, if conservation of mass is of more importance to the physics than the accuracy of other equations, special steps can be taken to enhance conservation of mass. Last but not least, manipulating a system into an equivalent $\mathcal{H}^1$-elliptic formulation (as will be defined in (4.7)) is challenging.

In the FOSLS methodology, a PDE is first recast (if needed) into a differential first-order system and the solution is posed as a minimization by taking the L^2-norm of each equation. Denote the nonlinear first-order system as $\mathcal{F}(\mathbf{u}) = \mathbf{f}$, such that $\mathcal{F} : \mathcal{V} \to \left(L^2\right)^q$, and define the minimization by

$$\mathbf{u}_* = \arg\min_{\mathbf{u}\in\mathcal{V}} \|\mathcal{F}(\mathbf{u}) - \mathbf{f}\|^2 , \tag{4.1}$$

where $\mathbf{u}_*$ is the exact solution, $\mathcal{V}$ is an appropriate Hilbert space, and q represents the number of equations.

Minimization is achieved either by applying a Newton iteration directly to the nonlinear

functional, which is called a FOSLS-Newton method, or by linearizing the PDE first, establishing a Newton iteration, and applying a FOSLS to the linearized system. This latter approach is called a Newton-FOSLS method and is the approach considered here.

Toward that end, let $\mathbf{u}_\ell$ be the current approximation of the solution and consider the Taylor expansion,

$$\mathcal{F}(\mathbf{u}_\ell + \delta) = \mathcal{F}(\mathbf{u}_\ell) + \mathcal{F}'(\mathbf{u}_\ell)[\delta] + \frac{1}{2}\mathcal{F}''(\hat{\mathbf{u}}_\ell)[\delta, \delta]\,, \tag{4.2}$$

where $\mathcal{F}'(\mathbf{u}_\ell)[\mathbf{v}]$ is the first Fréchet derivative at $\mathbf{u}_\ell$ in direction $\mathbf{v}$ and $\mathcal{F}''(\hat{\mathbf{u}}_\ell)[\mathbf{v}, \mathbf{w}]$ is the second Fréchet derivative at $\mathbf{u}_\ell$ in directions $\mathbf{v}$ and $\mathbf{w}$. Here, $\hat{\mathbf{u}}_\ell = \mathbf{u}_\ell + \alpha\delta$ for some $\alpha \in (0, 1)$. The first Fréchet derivative, $\mathcal{F}'(\mathbf{u}_\ell) : \mathcal{V} \to L^2$, is a linear operator on $\mathcal{V}$. To simplify notation, denote $\mathbb{L}_\ell = \mathcal{F}'(\mathbf{u}_\ell)$ and the residual by $R_\ell = \mathbf{f} - \mathcal{F}(\mathbf{u}_\ell)$.

A Newton iteration is established by ignoring the second Fréchet derivative and solving for δ by minimizing the L^2-norm:

$$\begin{aligned} \delta_\ell &= \arg\min_{\delta \in \mathcal{V}} \|\mathbb{L}_\ell \delta - R_\ell\|^2, && (4.3)\\ \mathbf{u}_{\ell+1} &= \mathbf{u}_\ell + \delta_\ell\,. && (4.4) \end{aligned}$$

The nonlinear iteration process is illustrated in Figure 4.1. At each step of the Newton iteration, a quadratic functional is minimized, resulting in an equivalent weak variational form: find $\delta_\ell \in \mathcal{V}$ such that, $\forall \mathbf{v} \in \mathcal{V}$,

$$\langle \mathbb{L}_\ell \delta_\ell, \mathbb{L}_\ell \mathbf{v} \rangle = \langle R_\ell, \mathbb{L}_\ell \mathbf{v} \rangle\,, \tag{4.5}$$

where $\langle \cdot, \cdot \rangle$ denotes the standard L^2 inner product. Define the bilinear form

$$B_\ell(\mathbf{v}, \mathbf{w}) = \langle \mathbb{L}_\ell \mathbf{v}, \mathbb{L}_\ell \mathbf{w} \rangle\,. \tag{4.6}$$

Assume that B_ℓ is continuous and coercive in the $\mathcal{V}$-norm (also called $\mathcal{V}$-elliptic), that is, there exist constants $c_0, c_1 > 0$, dependent only on $\mathbf{u}_\ell$, such that

$$\begin{aligned} B_\ell(\mathbf{v}, \mathbf{w}) &\leq c_1 ||\mathbf{v}||_{\mathcal{V}} ||\mathbf{w}||_{\mathcal{V}}, \\ B_\ell(\mathbf{v}, \mathbf{v}) &\geq c_2 ||\mathbf{v}||^2_{\mathcal{V}}\,. \end{aligned} \tag{4.7}$$

Then B is an inner product on $\mathcal{V}$ and the Riesz representation theorem implies that there exists a unique solution, $\delta_\ell \in \mathcal{V}$, that satisfies (4.5). In Section 5.5, it is shown that, for the TFP system (3.38), such bounds exist uniformly for every $\mathbf{u}_\ell$ in a neighborhood of the exact solution, $\mathbf{u}_*$.

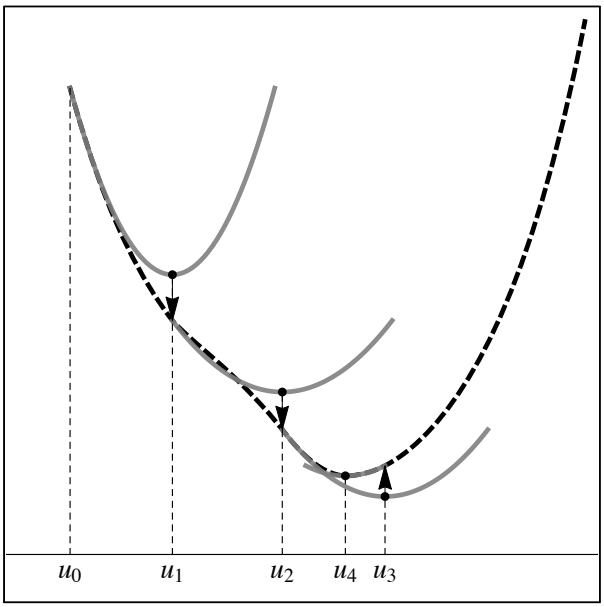

Figure 4.1: The dashed black line is a cartoon representation of a nonlinear functional. At each step of Newton's method, linearization produces a local quadratic functional, pictured as gray parabolas. Each quadratic functional is minimized. The nonlinear functional is linearized again around the found minimum and the process continues until a desired tolerence is reached.

Discretization is accomplished by restricting the minimization in (4.1) to a finite-dimensional subspace, $\mathcal{V}^h \subset \mathcal{V}$, resulting in a discrete nonlinear system. This is, in turn, solved by a Newton-FOSLS iteration by restricting the minimization in (4.3) and weak form in (4.5) to $\mathcal{V}^h$, resulting in a discrete weak form: given discrete approximation $\mathbf{u}_\ell^h$, find $\delta_\ell^h \in \mathcal{V}^h$ such that, $\forall\, \mathbf{v}^h \in \mathcal{V}^h$,

$$\langle \mathcal{F}'(\mathbf{u}_\ell^h)[\delta^h], \mathcal{F}'(\mathbf{u}_\ell^h)[\mathbf{v}^h] \rangle = \langle (\mathbf{f} - \mathcal{F}(\mathbf{u}_\ell^h)), \mathcal{F}'(\mathbf{u}_\ell^h)[\mathbf{v}^h] \rangle \,. \tag{4.8}$$

An important result that follows from (4.7) and (4.9) is that the functional is equivalent to measuring error in the $\mathcal{V}$-norm. This measure is locally computable and can be used efficiently to influence refinement strategies. This type of adaptive refinement is discussed further in Section 4.3.

Bounds (4.7) hold for any finite-dimensional subspace, which implies a unique solution to the discrete weak form (4.8). Choice of a finite element basis, $\mathcal{V}^h = \text{span}\,\{\phi_i\}_{i=1}^N$, in (4.5) admits a linear system, $A^h\mathbf{x}^h = \mathbf{b}$, where

$$\left(A^h\right)_{i,j} = \langle \mathcal{F}'(\mathbf{u}_\ell^h)[\phi_i], \mathcal{F}'(\mathbf{u}_\ell^h)[\phi_j]\rangle\,, \quad (\mathbf{b})_i = \langle (\mathbf{f} - \mathcal{F}(\mathbf{u}_\ell^h)), \mathcal{F}'(\mathbf{u}_\ell^h)[\phi_i]\rangle. \tag{4.9}$$

Given bounds (4.7), matrix A^h is symmetric positive definite. In the special case that the operator, $\mathcal{F}'(\mathbf{u}_\ell^h)$, is continuous and coercive in $\mathcal{V} = \mathcal{H}^1$, then an optimal multigrid algorithm exists for the generated linear system [10, 11]. If $\mathcal{V}$ is a product space of $\mathcal{H}^1$, $\mathcal{H}(\text{Div})$, and $\mathcal{H}(\text{Curl})$, optimal multigrid algorithms may also exist [3–5].

Ignoring boundary conditions, the least-squares weak form, (4.5), is equivalent to

$$\langle \mathbb{L}_\ell^*\mathbb{L}_\ell\delta, \mathbf{v}\rangle = \langle \mathbb{L}_\ell^* R_\ell, \mathbf{v}\rangle\,, \tag{4.10}$$

where the self-adjoint operator, $\mathbb{L}_\ell^*\mathbb{L}_\ell$, is known as the formal-normal. It is instructive to examine the structure of the formal-normal for FOSLS formulations. In many systems, scalings can be applied to $\mathbb{L}_\ell$ such that the formal-normal yields well-behaved, balanced, elliptic operators. The structure of the formal-normal acts as a litmus test of the effectiveness of using multigrid methods on the system.

4.2 Nested Iteration

Nested Iteration (NI) is employed as part of our solution technique. The main idea of NI is that the problem is solved on a sequence of successively finer meshes. The total cost of all the work performed on the coarse meshes is small, but generates an initial guess for the finest mesh that is close to the minimum of the nonlinear functional, thus requiring only a few additional Newton-FOSLS iterations.

To be more precise, let Ω^h represent a mesh with spacing h. Let $\mathbf{u}_i^h$ be the initial guess for a Newton-FOSLS iteration on Ω^h and $\mathbf{u}_f^h$ be the final iterate of the Newton-FOSLS process. An initial guess, $\mathbf{u}_i^H$, is made on a very coarse mesh, where computation is extremely cheap. Some

nonlinear solver is then employed to resolve the approximate discrete solution, $\mathbf{u}_f^H$. Mesh refinement is performed, and the discrete coarse mesh solution is interpolated into the finer mesh, $\Omega^{\frac{H}{2}}$. The interpolated solution is taken as the initial guess for the Newton-FOSLS iteration,

$$P\mathbf{u}_f^H \rightarrow \mathbf{u}_i^{\frac{H}{2}}\ , \tag{4.11}$$

where P represents the interpolation operator between meshes. The process is repeated on all subsequent levels, interpolating each approximate discrete solution into the next finer mesh as an initial guess. See Figure 4.2 for an illustration of NI. The process is stopped once a specified tolerance or desired mesh is reached. The goal of all of the work done on the coarse meshes is to produce an initial guess on the finest mesh that well approximates the minimum. In this way, the finest mesh Newton-FOSLS process only needs to perform a handful of iterations [22, 23].

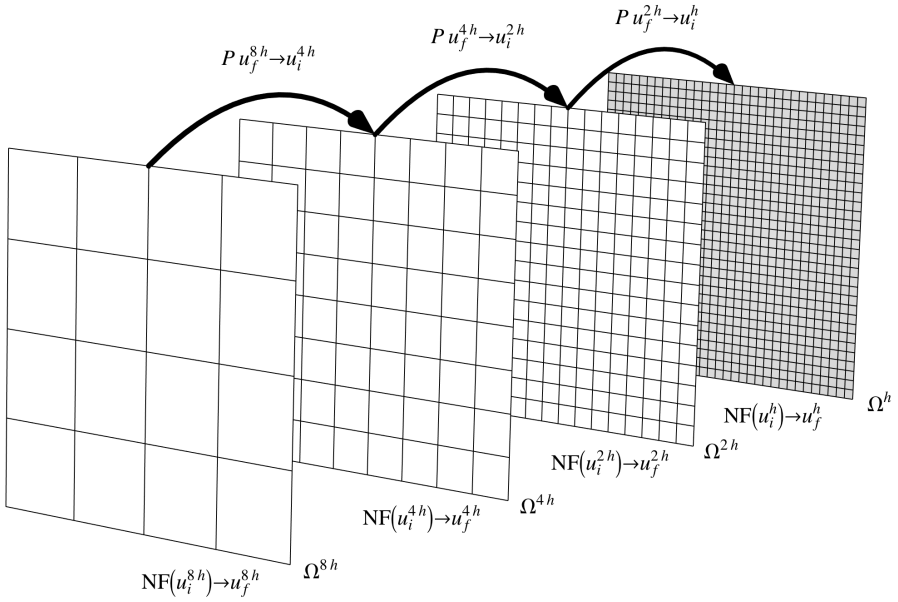

Figure 4.2: In the Nested Iteration process, a sequence of increasingly finer meshes is used to reduce the overall computational cost of a simulation.

To make the NI strategy as efficient as possible, it is desirable to reduce the amount of work done on the coarse meshes while still producing a suitable initial guess for the fine mesh problem. It

is not necessary to solve the discrete coarse problems exactly; the solution only needs to be within discretization error such that the interpolation into the next finer mesh moves the overall algorithm closer to the true solution, u_* [53].

Let $\{u_0^{2h}, u_1^{2h}, \ldots\}$ be a sequence of Newton-FOSLS iterates converging to the true discrete solution, u_*^{2h}, on mesh Ω^{2h}. At some point during the iteration, interpolating into the finer space, Ω^h, moves the approximate solution closer to the true solution, u_*, than continuing towards u_*^{2h}. Take u_ℓ^{2h} to be the current iterate and consider two strategies:

(a) Interpolate the current approximation, u_ℓ^{2h}, into grid Ω^h, $Pu_\ell^{2h} \to u_0^h$, and perform a Newton-FOSLS iteration to give u_1^h.

(b) First, take an additional iteration on grid Ω^{2h} to produce $u_{\ell+1}^{2h}$, interpolate into grid Ω^h, $Pu_{\ell+1}^{2h} \to \hat{u}_0^h$, and then perform a Newton-FOSLS iteration to give $\hat{u}_1^h$.

Figure 4.3 gives a geometric representation of a situation where the additional work in strategy (b) provides no additional benefit, that is, $\|u_1^h - u_*\| < \|\hat{u}_1^h - u_*\|$. The decision must be made carefully. One one hand, the iterations on mesh Ω^{2h} are cheaper than on mesh Ω^h but, on the other hand, the iterations are not heading towards the correct solution. Outlining a rigorous strategy for minimizing the number of linearization steps and AMG V-Cycles on the coarse meshes is out of the scope of this thesis, but is covered extensively in [1, 53] and its utility demonstrated in [2].

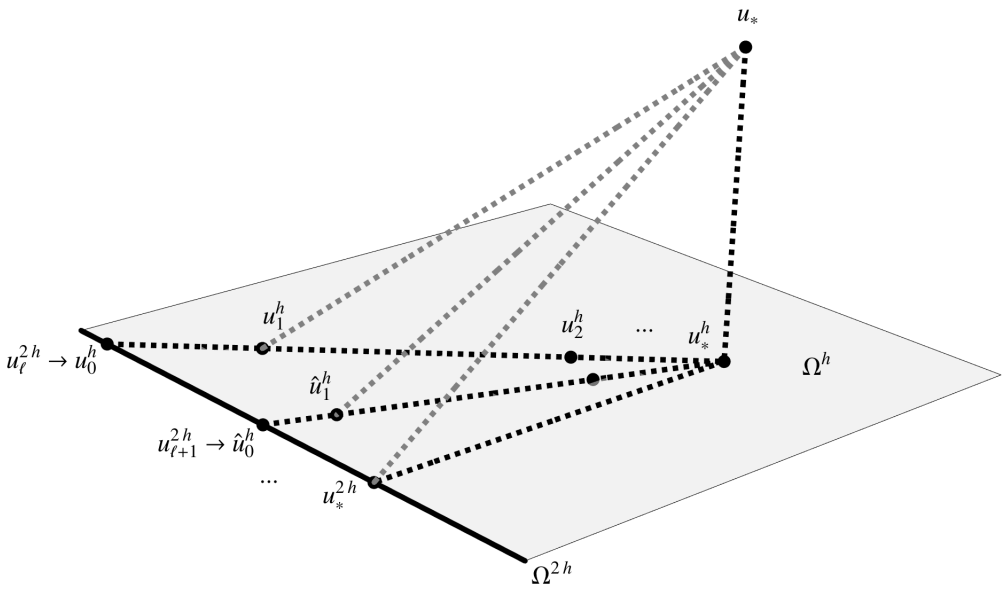

Figure 4.3: The black line represents the discrete Hilbert subspace of solutions on mesh Ω^{2h} and the gray plane represents the larger (fine mesh) discrete Hilbert subspace is solutions on mesh Ω^h. The true continuous solution is represented by u_*. It should be noted that this figure is not to scale: in reality, u^* should be much closer to u^h.

4.3 Adaptive Refinement

To further improve the efficiency of the NI-Newton-FOSLS algorithm, the FOSLS functional is used as a local error estimator, indicating where mesh refinement should be performed. Define the FOSLS functional as

$$\mathcal{G}_\ell(\delta^h; f) = \|\mathbb{L}_\ell^h \delta^h - R_\ell\|^2 , \tag{4.12}$$

with $\mathbb{L}_\ell^h = \mathcal{F}'(\mathbf{u}_\ell^h)$, the discrete analog of $\mathbb{L}_\ell$. A small touch of algebra gives

$$\mathcal{G}_\ell(\delta^h; f) = \|\mathbb{L}_\ell^h \delta^h - R_\ell\|^2 = \|\mathbb{L}_\ell^h \delta^h - \mathbb{L}_\ell^h \delta_*\|^2 = \|\mathbb{L}_\ell^h \mathbf{e}\|^2 , \tag{4.13}$$

where δ_* is the true solution to the discrete linear problem and $\mathbf{e} = \delta^h - \delta_*$ is the remaining error. Using the result of $\mathcal{V}$-ellipticity in (4.7) and the chain of equalities in (4.13) leads to the fact that

$$c_1\|\mathbf{e}\|_{\mathcal{V}} \leq \mathcal{G}_\ell(\delta^h; f) \leq c_2\|\mathbf{e}\|_{\mathcal{V}}\,. \tag{4.14}$$

That is, the FOSLS functional is an equivalent measure to the size of the error in the $\mathcal{V}$-norm. The upper bound implies that the FOSLS functional can be used as an a posteriori error measure. That is, when $\frac{\mathcal{G}_\ell(\delta^h;f)}{c_2}$ is large, so is the remaining error. Moreover, the functional is locally computable in each element, revealing where in the mesh more computational effort should be focused.

On a given mesh, when the NI-Newton-FOSLS strategy indicates that it is time to refine and move the solution into a finer space, the FOSLS functional is computed locally in each element. Instead of refining all elements (i.e., uniform refinement), only a fraction of the elements with the largest error are marked for refinement. The elements that are refined are picked using the Accuracy per Computational cost Efficiency (ACE) measure [24, 45]. In short, ACE balances the reduction in the error of refining elements with the induced computational overhead on the next level.

After performing the desired number of Newton-FOSLS iterations on mesh j, consider ordering the elements from largest error to smallest error, $e_1 \geq e_2 \geq e_3 \geq \ldots \geq e_N$, where N_j is the number of elements. The first rN elements are chosen to be refined, where $0 < r \leq 1$. Each refinement generates 2^d children elements, where d is the dimension. Thus, the number of elements on the next mesh, $j+1$, can be related to the refinement fraction, r, by

$$N_{j+1} \approx N_j \left(1 - r + 2^d r\right)\,. \tag{4.15}$$

The fraction of elements marked for refinement needs to be chosen carefully, being mindful of the AMG setup cost, $C(r)$, the error reduction factor, $\gamma(r)$, and the convergence factor, $\rho(r)$, on level $j+1$. Each of these values is estimated as a function of r. After predicting these components, the total amount of work it takes to solve the problem on the $j+1$ mesh, $W(r)$, can also be estimated as a function of r. The optimal ratio, r_*, is then calculated by minimizing the effective reduction factor

$$r_* = \arg\min_r \gamma(r)^{\frac{1}{W(r)}}\,. \tag{4.16}$$

The details of this process can be found in [50]. The ACE algorithm can be tuned for aggressive refinement strategies by allowing elements to be refined multiple times.

As an example of the ACE algorithm in action, consider solving the FOSLS formulation of the Poisson equation on the domain, $\Omega = [0,1] \times [0,1]$:

$$\begin{aligned} \mathbf{u} - \nabla p &= \mathbf{0}\,, \\ \nabla \cdot \mathbf{u} &= f(x,y)\,, \\ \nabla \times \mathbf{u} &= \mathbf{0}\,, \end{aligned} \tag{4.17}$$

subject to Dirichlet boundary conditions,

$$\begin{aligned} p &= 0\,, \qquad \partial\Omega\,, \\ \hat{\mathbf{n}} \times \mathbf{u} &= 0\,, \qquad \partial\Omega\,. \end{aligned} \tag{4.18}$$

As an experiment, take $f(x,y) = \exp\left(100(x-0.25)^2\right)\exp\left(100(y-0.25)^2\right)$ as seen in Figure 4.4. The forcing function has a sharp feature centered at $(0.25, 0.25)$.

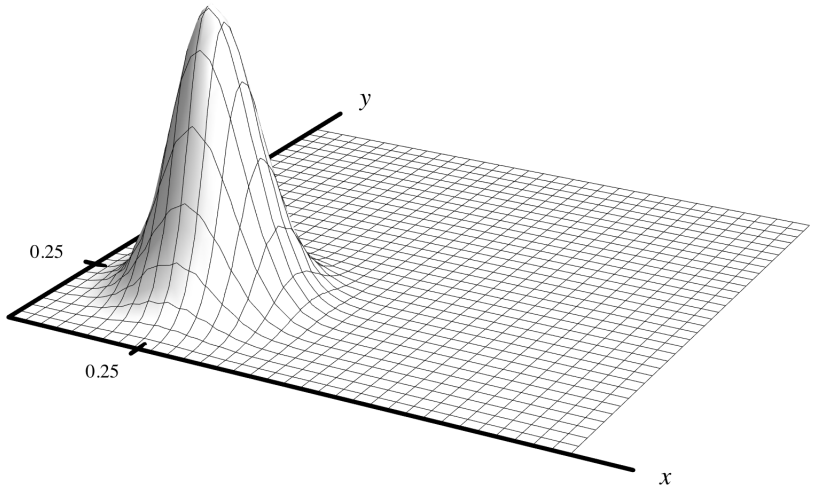

Figure 4.4: The source function $f(x,y) = \exp\left(100(x-0.25)^2\right)\exp\left(100(y-0.25)^2\right)$ used in (4.17) on domain $[0,1] \times [0,1]$

It is intuitive that during the NI process, very few mesh points are required to capture the behavior of the solution far from $(0.25, 0.25)$. On the other hand, near the steep gradient, more elements are required. Using an initial guess of zero for both p and $\mathbf{u}$, ACE refinement produces the sequence of nested meshes seen in Figure 4.5. An initial mesh with 64 elements was used and refined 3 times, producing a mesh with 2026 elements. Three levels of uniform refinement would produce a mesh with 4096 elements. Even though the ACE-produced mesh contains approximately half the number of elements, the computational efforts are better focused on the elements where large errors remain. ACE requires less computational resources to reduce the L^2-error to a comparable level as the same amount of resources used with uniform refinement.

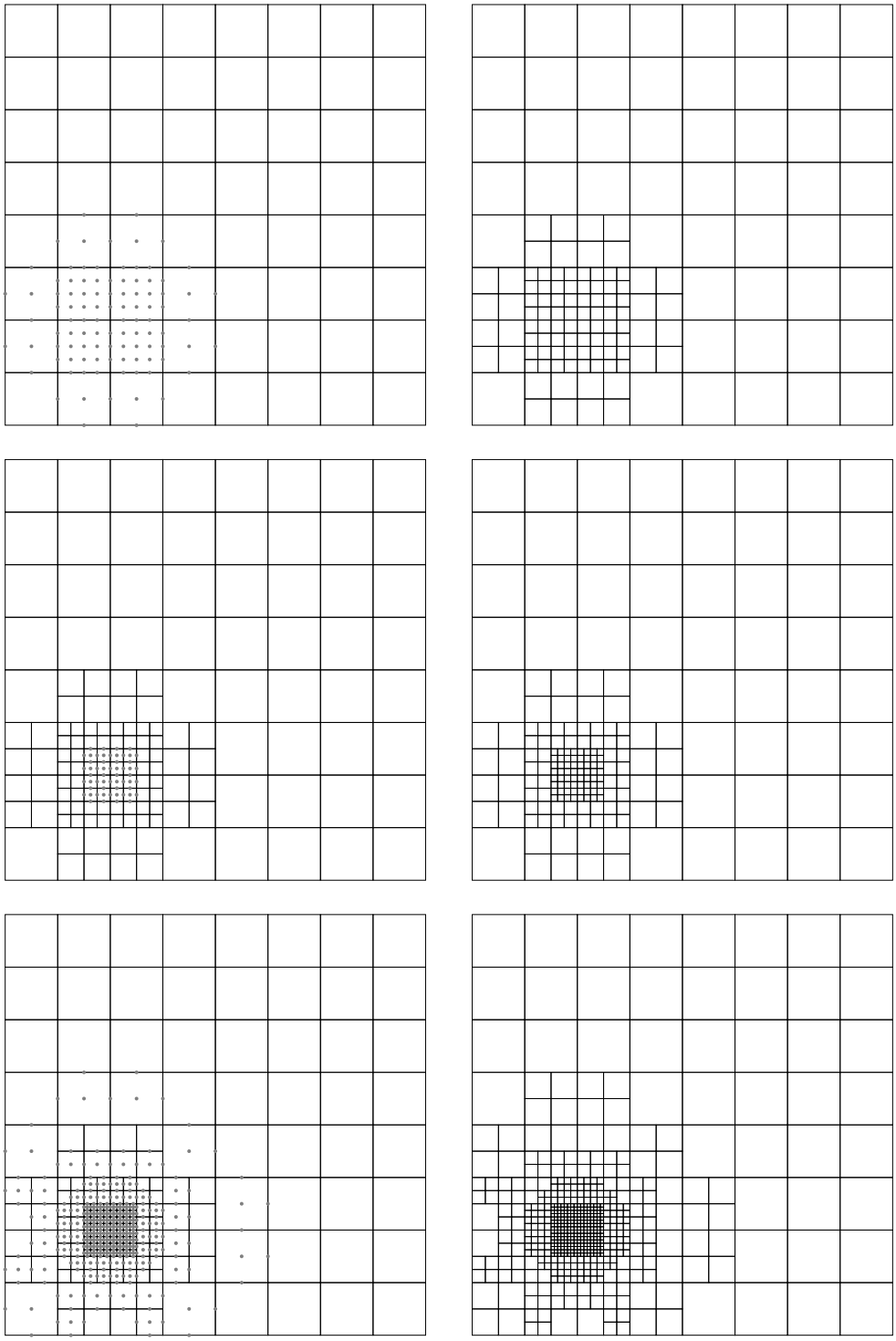

Figure 4.5: Three levels of ACE refinement used on problem (4.17) with an initial mesh of 8×8. The gray dots that appear in the left column indicate which finite element degrees of freedom are selected by ACE for refinement.

4.4 Multigrid

The linear systems generated by the discrete weak form, (4.9), at each step of the Newton-FOSLS process must be solved quickly and efficiently. The fact that the linear systems stem from a FOSLS discretization makes the use of multigrid (MG) methods promising. As was stated in Section 4.1, we strive to ensure that the first-order system be $\mathcal{H}^1$-elliptic in order to guarantee that standard finite element spaces generate a linear system amenable to MG methods. As shown later in Section 5.5, the TFP-Darwin system can be modified to produce a fully $\mathcal{H}^1$-elliptic system. In this section, we describe a standard Algebraic Multigrid (AMG) solver, as applied to the general linear system

$$Ax = b\,, \tag{4.19}$$

where $A \in \mathbb{R}^{N_f \times N_f}$ and $x, b \in \mathbb{R}^{N_f}$. Two key components of multigrid are a relaxation process (also called the smoother) and restriction/interpolation operators that transfer vectors between fine and coarse grids. The components of MG work together in a complimentary way. The relaxation process quickly removes the high frequency components from the error. The relaxation process stagnates when all that remains in the error is smooth; continuing to apply the smoother is ineffective. Instead, after only a few smoothing steps, the problem is restricted to a coarser grid, where the remaining smooth error can be easily approximated. The coarse-grid problem is solved using less work than its fine grid counterpart. The solution of the coarse-grid problem is interpolated back to the fine grid, where it is smoothed a few more times and used to correct the fine-grid solution.

Let x_* and x represent the true solution and approximate solution on the fine grid, respectively. Define the residual as

$$r = b - Ax\,, \tag{4.20}$$

and observe that the the linear system can also be expressed as

$$Ae = r\,, \tag{4.21}$$

where $e = x_* - x$ is the error. It is trivially true that $x_* = x + e$; if the error was known exactly,

the problem would be solved. Instead, we define a process that constructs the approximate error, $\hat{e}$, and iterates in order to converge to the true solution:

$$x \leftarrow x + \hat{e}\,. \tag{4.22}$$

Define the action of applying the smoother ν times as $\hat{e}_\nu = S_\nu^{-1} r$. The smoothing process is often taken to be a simple Jacobi or Gauss-Seidel sweep. In its current state, convergence of the iteration, $x \leftarrow x + S_\nu^{-1} r$, rapidly slows as all that remains of the error is smooth. The problem must be dealt with on a coarse-grid.

Let $R \in \mathbb{R}^{N_c \times N_f}$ and $P \in \mathbb{R}^{N_f \times N_c}$ identify the restriction and interpolation operators, respectively, where N_f is the size of the fine grid and N_c is the size of the coarse-grid. The smoothed residual is restricted to the coarse-grid, $r_\nu^c = RA\hat{e}_\nu$. The coarse-grid error is then found by solving

$$A^c e_\nu^c = r_\nu^c\,, \tag{4.23}$$

where $A^c \in \mathbb{R}^{N_c \times N_c}$ is a coarse-grid representation of A (to be discussed later). The coarse-grid error is then interpolated back to the fine grid, $\hat{e}_\nu^f = P\hat{e}_\nu^c$. It is assumed that, once interpolated, the coarse-grid error well approximates the fine grid error. But because the interpolation from coarse-grid to fine grid is not exact, an additional μ applications of the smoother are applied, giving us our final error correction for $\hat{e}$:

$$\hat{e} = S_\mu^{-1} A \hat{e}_\nu^f\,. \tag{4.24}$$

Bringing all of these components together gives us the definition of a single coarse-grid correction scheme with ν pre-smooths and μ post-smooths:

$$x \leftarrow x + \left[S_\mu^{-1} AP(A^c)^{-1} RAS_\nu^{-1}\right] r\,. \tag{4.25}$$

The process is summarized again by reading the entire operator from right to left. First ν smoothing steps are applied to the current residual, producing relatively smooth error. The application of A turns the smooth error into a residual. The operator R restricts the residual to the coarse-grid. Applying $(A^c)^{-1}$ gives the coarse-grid error, which is then interpolated back into the fine grid via

P. This new fine grid error is smoothed once more by being converted into a residual and smoothed μ more times with $S_\mu^{-1}A$. For an algorithmic description of the process, see Algorithm 1:

Input: Matrix, A, right-hand side vector, b, and initial guess x_i.
Output: An apporximate solution $x_f \approx A^{-1}b$
VCycle(A, b, x_i, ν, μ):
if GRID = COARSEST **then**
  $x_f = A^{-1}b$
else
  Relax ν times on $Ax_i = b$
  Construct P and R
  Restrict the residual $r^c = Rr = R(b - Ax_i)$
  Form $A^c = P^T AP$
  $x_f = x_i + \text{VCycle}(A^c, r^c, 0, \nu, \mu)$
  Relax μ times on $Ax_f = b$
end
return x_f

Algorithm 1: The V(ν, μ)-Cycle. Note that the relaxation process is taken to mean that the vector is updated in place.

We have neglected to state how the coarse-grid problem, $A^c e^c = r^c$, is solved. If Jacobi or Gauss-Seidel was, again, used to solve for e^c, we would get bottlenecked by the fact that the solution stalls after a handful of iterations. To overcome this, a coarse-grid correction to the coarse-grid can be used again! Coarse-grid corrections are applied recursively to solve each coarse-grid error equation until the grids become coarse enough that a direct solution is easily attained. Starting from the coarsest grid, the errors are interpolated back up, correcting the solution on each successively finer grid. This type of nesting of coarse-grid corrections is known as a multigrid V-Cycle, or often as a V(ν, μ)-Cycle, referring to the number of pre- and post-smoothing operations. A very common form of V-Cycle is the V(2,1)-Cycle, with Gauss-Seidel used for the smoother.

There are several ways to construct the coarse-grid operator, A^c, but for the purposes of this thesis, only the variational formulation is considered [8]. Assuming the interpolation operator is known, define the coarse-grid operator as

$$A^c = RAP\,, \tag{4.26}$$

where $R = P^T$. All that remains in order to complete the description of the V-Cycle algorithm is to

define the interpolation operator from the coarse to fine grid. The choice of P dictates the type of multigrid method. The process of choosing a good interpolation operator, P, is quite involved and is explored in depth in [8]. One of the most traditional types of multigrid is Algebraic Multigrid (AMG). The core idea of interpolation in AMG is to construct P such that algebraically smooth error is well represented. Define the algebraically smooth error vector, e_s, such that

$$S^{-1}Ae_s \approx 0\,. \tag{4.27}$$

That is, the act of relaxation no longer produces a substantial correction to the current iterate, indicating that the smoother has stalled. Whatever remains is considered "smooth" in the eye of the matrix. To interpolate a coarse-grid error to a fine grid error, define

$$(Pe^c)_i = \begin{cases} e_i^c & i \in C \\ \sum_{j \in C_i} \omega_{ij} e_j & i \in F \end{cases}, \tag{4.28}$$

where C is the collection of coarse-grid points, F is the collection of fine-grid points with the coarse-grid points removed, C_i is the set of coarse-grid points that strongly influence the point i, and ω_{ij} are the interpolation weights. The selection of the sets C, F, and C_i and weights, ω_{ij}, are determined by careful analysis of the matrix entries, and is outside the scope of this thesis [51].

Chapter 5

FOSLS Formulation of Two-Fluid Plasma

In this chapter, the TFP-Darwin system is modified in order to produce a system that is fully $\mathcal{H}^1$-elliptic. Each linear diagonal block is scaled to ensure optimal algebraic multigrid (AMG) performance. In the case of the fluid blocks, the system does not naturally admit $\mathcal{H}^1$ ellipticity and an additional equation must be introduced. Once the block diagonals are investigated, the off diagonal nonlinearities of the TFP-Darwin model are addressed in Theorem 5.5.1, proving the $\mathcal{H}^1$-ellipticity of the entire system.

5.1 TFP Blocks

Starting from (3.38), a basic implicit time discretization is assumed, resulting in $\partial_t \to \frac{1}{\delta t}$. All explicit time discretization terms are grouped into the right-hand side, effectively acting as source terms. The unknowns are interpreted to be at the current implicit time step. Implicit time discretization exposes the spatial operator that must be efficiently inverted. The nonlinear terms $n_\alpha \mathbf{E}_r$, $\mathbf{p}_\alpha \times \mathbf{B}$, and $n_\alpha \mathbf{E}_s$ are linearized according to the methods outlined in Chapter 4 (for example, $n_\alpha \mathbf{E}_r \to n_\alpha^0 \delta \mathbf{E}_r + \mathbf{E}_r^0 \delta n_\alpha + n_\alpha^0 \mathbf{E}_r^0$), and are addressed in Section 5.4. The TFP system,

(3.38), is written as a differential matrix operator:

$$\left[\begin{array}{cc|cc|ccc}
\frac{m_i}{\delta t} & T_i\nabla & 0 & 0 & -q_i n_i & -q_i \mathbf{p}_i \times & -q_i n_i \\
\nabla\cdot & \frac{1}{\delta t} & 0 & 0 & 0 & 0 & 0 \\
\hline
0 & 0 & \frac{m_e}{\delta t} & T_e\nabla & -q_e n_e & -q_e \mathbf{p}_e \times & -q_e n_e \\
0 & 0 & \nabla\cdot & \frac{1}{\delta t} & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & \nabla\times & 0 & 0 \\
0 & -\frac{q_i}{\epsilon_0} & 0 & -\frac{q_e}{\epsilon_0} & \nabla\cdot & 0 & 0 \\
-q_i\mu_0 & 0 & -q_e\mu_0 & 0 & -\frac{\epsilon_0\mu_0}{\delta t} & \nabla\times & 0 \\
0 & 0 & 0 & 0 & 0 & \nabla\cdot & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1}{\delta t} & \nabla\times \\
0 & 0 & 0 & 0 & 0 & 0 & \nabla\cdot
\end{array}\right]
\left[\begin{array}{c}
\mathbf{p}_i \\ n_i \\ \hline \mathbf{p}_e \\ n_e \\ \hline \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s
\end{array}\right]
=
\left[\begin{array}{c}
\mathbf{f}_{\mathbf{p}_i} \\ g_{n_i} \\ \hline \mathbf{f}_{\mathbf{p}_e} \\ g_{n_e} \\ \hline 0 \\ g_{\mathbf{E}_r} \\ \mathbf{f}_{\mathbf{B}} \\ 0 \\ \mathbf{f}_{\mathbf{E}_s} \\ 0
\end{array}\right], \tag{5.1}$$

subject to the pure conductive boundary conditions (3.39) and (3.40)

$$\begin{aligned}
\hat{\mathbf{n}}\cdot\nabla n_\alpha &= 0\,, \\
\mathbf{p}_\alpha &= \mathbf{0}\,, \\
\hat{\mathbf{n}}\times\mathbf{E}_r &= \mathbf{0}\,, \\
\hat{\mathbf{n}}\times\mathbf{E}_s &= \mathbf{0}\,, \\
\hat{\mathbf{n}}\cdot\mathbf{B} &= 0\,,
\end{aligned} \tag{5.2}$$

where $\hat{\mathbf{n}}$ denotes the outward unit normal to the boundary.

As presented in (5.1), the TFP system does not naturally admit $\mathcal{H}^1$-ellipticity. It can be shown that system (5.1) is instead $\mathcal{V}_0$-elliptic with $\mathbf{p}_\alpha \in \mathcal{H}\,(\mathrm{Div})$, $n_\alpha \in \mathcal{H}^1$, and $\mathbf{E}_r, \mathbf{B}, \mathbf{E}_s \in \left(\mathcal{H}^1\right)^3$. That is,

$$\mathcal{V}_0 = \mathcal{H}\,(\mathrm{Div}) \otimes \mathcal{H}^1 \otimes \mathcal{H}\,(\mathrm{Div}) \otimes \mathcal{H}^1 \otimes \left(\mathcal{H}^1\right)^3 \otimes \left(\mathcal{H}^1\right)^3 \otimes \left(\mathcal{H}^1\right)^3\,. \tag{5.3}$$

Posed in this space, momentum densities are not in $\mathcal{H}^1$. Special care is required to ensure multigrid algorithms yield optimal convergence. One approach is to use $\mathcal{H}\,(\mathrm{Div})$-conforming finite elements, for example, Raviart-Thomas elements, for the momentum densities and a non-standard multigrid

algorithm that employs distributed relaxation based on the support of the divergence-free basis elements [3, 4].

The alternative used in this thesis is to directly modify (5.1) by introducing an additional constraint, pulling $\mathbf{p}_\alpha$ back into $(\mathcal{H}^1)^3$. The assumption that $\mathbf{p}_\alpha \in (\mathcal{H}^1)^3$ is reasonable for most plasma systems of interest. Denote the differential matrix system in (5.1) by $\mathcal{F}(\mathbf{u}) = \mathbf{f}$ and identify the block differential matrix systems as

$$\mathcal{F}(\mathbf{u}) = \begin{bmatrix} \mathbb{A}_i & 0 & 0 \\ 0 & \mathbb{A}_e & 0 \\ \mathbb{L}_i & \mathbb{L}_e & \mathbb{D} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_e \\ \mathbf{u}_D \end{bmatrix} + \begin{bmatrix} 0 & 0 & \mathbb{U}_i(\mathbf{u}_i, \mathbf{u}_D) \\ 0 & 0 & \mathbb{U}_e(\mathbf{u}_e, \mathbf{u}_D) \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.4}$$

where the first term is linear and $\mathbb{U}_i(\mathbf{u}_i, \mathbf{u}_D)$ and $\mathbb{U}_e(\mathbf{u}_e, \mathbf{u}_D)$ are functions that contain the nonlinear upper right-hand blocks in (5.1).

In the following sections, the block systems are investigated. First, the Darwin block, $\mathbb{D}$, is scaled and shown to be $\mathcal{H}^1$-elliptic, independent of all physical constants. Then, the fluid blocks, $\mathbb{A}_i$ and $\mathbb{A}_e$, are modified with an additional equation, scaled, and shown to be $\mathcal{H}^1$-elliptic. The fluid blocks are solved with a variety of physical parameters to demonstrate the robustness of the AMG solver applied to the modified system. Finally, in Section 5.4, the Fréchet derivative of the entire system is addressed and the linearized system is shown to be uniformly $\mathcal{H}^1$-elliptic.

5.2 Darwin Block

The Darwin block, $\mathbb{D}$, is naturally $\mathcal{H}^1$-elliptic, but contains scalings that are dependent upon physical constants and the size of the time step. The isolated (uncoupled) Darwin system is

$$\begin{bmatrix} \nabla\times & 0 & 0 \\ \nabla\cdot & 0 & 0 \\ -\frac{\epsilon_0\mu_0}{\delta t}\mathbb{I} & \nabla\times & 0 \\ 0 & \nabla\cdot & 0 \\ 0 & \frac{1}{\delta t}\mathbb{I} & \nabla\times \\ 0 & 0 & \nabla\cdot \end{bmatrix} \begin{bmatrix} \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s \end{bmatrix} = \begin{bmatrix} 0 \\ g_{\mathbf{E}_r} \\ \mathbf{f}_{\mathbf{B}} \\ 0 \\ \mathbf{f}_{\mathbf{E}_s} \\ 0 \end{bmatrix}. \tag{5.5}$$

The formal-normal of $\mathbb{D}$ is

$$\mathbb{D}^*\mathbb{D} = \begin{bmatrix} \frac{\epsilon_0^2\mu_0^2}{\delta t^2}\mathbb{I} - \Delta & -\frac{\epsilon_0\mu_0}{\delta t}\nabla\times & 0 \\ -\frac{\epsilon_0\mu_0}{\delta t}\nabla\times & \frac{1}{\delta t^2}\mathbb{I} - \Delta & \frac{1}{\delta t}\nabla\times \\ 0 & \frac{1}{\delta t}\nabla\times & -\Delta \end{bmatrix}. \tag{5.6}$$

Depending on the value of the physical constants, the first-order off-diagonal curl operators could potentially dominate the system, leading to poor AMG convergence until the mesh becomes sufficiently fine. We remedy this by introducing a general left scaling, $\mathbb{R}_D = \mathrm{diag}\,[r_1, r_2, r_3, r_4, r_5, r_6]$, and general right scaling, $\mathbb{S}_D = \mathrm{diag}\,[s_1, s_2, s_3]$. Take note that the constants in the diagonal scaling matrices are expanded by an appropriately sized identity matrix, which should be obvious from the context of scaling either a vector or scalar quantity. The scaled system takes on the form

$$\mathbb{R}_D\,\mathbb{D}\,\mathbb{S}_D\,\left(\mathbb{S}_D^{-1}\mathbf{u}_D\right) = \mathbb{R}_D\,\mathbf{f}_D\,,$$

where $\mathbf{f}_D$ is the right-hand side of (5.5). Define $\hat{\mathbf{u}}_D = \left(\mathbb{S}_D^{-1}\right)\mathbf{u}_D$ to be the transformed unknown variables. The formal-normal of the new scaled linear system looks messy at first glance:

$$\left(\mathbb{R}_D\,\mathbb{D}\,\mathbb{S}_D\right)^*\left(\mathbb{R}_D\,\mathbb{D}\,\mathbb{S}_D\right) =$$

$$\begin{bmatrix} \frac{\epsilon_0^2\mu_0^2 r_3^2 s_1^2}{\delta t^2}\mathbb{I} - r_1^2 s_1^2\Delta + (r_1^2 s_1^2 - r_2^2 s_1^2)\nabla\nabla\cdot & -\frac{\epsilon_0\mu_0 r_3^2 s_1 s_2}{\delta t}\nabla\times & 0 \\ -\frac{\epsilon_0\mu_0 r_3^2 s_1 s_2}{\delta t}\nabla\times & \frac{r_5^2 s_2^2}{\delta t^2}\mathbb{I} - r_3^2 s_2^2\Delta + (r_3^2 s_2^2 - r_4^2 s_2^2)\nabla\nabla\cdot & \frac{r_5^2 s_2 s_3}{\delta t}\nabla\times \\ 0 & \frac{r_5^2 s_2 s_3}{\delta t}\nabla\times & -r_5^2 s_3^2\Delta + (r_5^2 s_3^2 - r_6^2 s_3^2)\nabla\nabla\cdot \end{bmatrix}.$$

Setting up a system of nonlinear equations allows us to solve for scalings such that the formal-normal is as "clean" as possible. We try to remove all $\nabla\nabla\cdot$ terms and make all other operators

have coefficients of unity. The nonlinear system to accomplish this is

$$\begin{aligned}
\tfrac{\epsilon_0^2\mu_0^2 r_3^2 s_1^2}{\delta t^2} &= 1, \\
r_1^2 s_1^2 &= 1, \\
r_1^2 s_1^2 - r_2^2 s_1^2 &= 0, \\
\tfrac{r_5^2 s_2^2}{\delta t^2} &= 1, \\
r_3^2 s_2^2 &= 1, \\
r_3^2 s_2^2 - r_4^2 s_2^2 &= 0, \\
r_5^2 s_3^2 &= 1, \\
r_5^2 s_3^2 - r_6^2 s_3^2 &= 0, \\
\tfrac{\epsilon_0\mu_0 r_3^2 s_1 s_2}{\delta t} &= 1, \\
\tfrac{r_5^2 s_2 s_3}{\delta t} &= 1\,.
\end{aligned} \tag{5.7}$$

The above system can be solved uniquely, up to the value of r_1, by demanding that all scalings be positive. For simplicity, select $r_1 = 1$, which, in turn, defines

$$\begin{aligned}
\mathbb{R}_D &= \operatorname{diag}\left[1, 1, \tfrac{\delta t}{\epsilon_0\mu_0}, \tfrac{\delta t}{\epsilon_0\mu_0}, \tfrac{\delta t^2}{\epsilon_0\mu_0}, \tfrac{\delta t^2}{\epsilon_0\mu_0}\right], \\
\mathbb{S}_D &= \operatorname{diag}\left[1, \tfrac{\epsilon_0\mu_0}{\delta t}, \tfrac{\epsilon_0\mu_0}{\delta t^2}\right].
\end{aligned} \tag{5.8}$$

These modifications generate the scaled Darwin system:

$$\widehat{\mathbb{D}} = \mathbb{R}_D\,\mathbb{D}\,\mathbb{S}_D = \begin{bmatrix}
\nabla\times & 0 & 0 \\
\nabla\cdot & 0 & 0 \\
-\mathbb{I} & \nabla\times & 0 \\
0 & \nabla\cdot & 0 \\
0 & \mathbb{I} & \nabla\times \\
0 & 0 & \nabla\cdot
\end{bmatrix}. \tag{5.9}$$

Carrying the same scaling through to the unknowns gives us the transformations: $\widehat{\mathbf{E}}_r = \mathbf{E}_r$, $\widehat{\mathbf{B}} = \frac{\delta t}{\epsilon_0\mu_0}\mathbf{B}$, and $\widehat{\mathbf{E}}_s = \frac{\delta t^2}{\epsilon_0\mu_0}\mathbf{E}_s$. The right-hand side terms are transformed in a similar way. With standard boundary conditions, it is straightforward to prove that this system is $\mathcal{H}^1$-elliptic, as seen by direct application of the Standard Compactness Argument, Lemma 5.5.1, and the coercivity of

Div-Curl blocks (see Lemma 5.5.3). The nature of what remains is clearly seen by considering the formal-normal of $\widehat{\mathbb{D}}$:

$$\widehat{\mathbb{D}}^*\widehat{\mathbb{D}} = \begin{bmatrix} \mathbb{I} - \Delta & -\nabla\times & 0 \\ -\nabla\times & \mathbb{I} - \Delta & \nabla\times \\ 0 & \nabla\times & -\Delta \end{bmatrix}. \tag{5.10}$$

The system is differentially diagonally dominant, and the diagonal blocks dominate the off-diagonal blocks, leading to a system that, for fine enough mesh, is nearly identical to a decoupled block vector Laplacian operator.

5.2.1 Darwin Block Numerical Test

To demonstrate how an AMG solver performs on the Darwin subsystem, two tests are performed. In the first, the AMG asymptotic convergence factor is estimated by selecting a zero right-hand side and using a random initial guess. A tight AMG[1] solver tolerance is prescribed in order to ensure the convergence factors settle into their asymptotic region. In the second test, a solution is built that is consistent with (5.2). The right-hand side us constructed by applying the Darwin block operator to the manufactured solution. Again, a random initial guess is used. A much looser AMG solver tolerance is used as outlined in Section 4.2. We demonstrate the convergence of the FOSLS functional and L^2-norm of the error (L^2-error). In both tests, uniform NI is used through 7 levels of refinement with quadratic elements. The coarsest mesh is solved using linear finite elements; these rates and results are not included. The linear finite element solution only serves as a cheap initial guess to bootstrap the NI process.

Let ρ denote the asymptotic convergence factor of AMG. The value of ρ is computed by allowing AMG V-cycles, on each level of refinement, to continue until a stable factor is reached. The asymptotic AMG convergence factors for both $\mathbb{D}$ and $\widehat{\mathbb{D}}$ are seen in Table 5.1. The system is non-dimensionalized by working in Alfvénic units as outlined in Appendix A.3. The domain is set to $[-1,1]\times[-1,1]$. Two different cases are considered. For system $\mathbb{D}_a$, set $\delta t = 0.1$, $\epsilon_0 = 0.3^2$,

[1] In the solution techniques employed in this section, it should be noted that the AMG solver is accelerated with an outer conjugate gradient loop.

and $\mu_0 = 1.0$ and, for system $\mathbb{D}_b$, set $\delta t = 0.01$, $\epsilon_0 = 0.3^2$, and $\mu_0 = 1.0$. The scaled system, $\mathbb{D}$, is independent of physical constants and has the same performance regardless of the values of δt, ϵ_0, and μ_0.

Lev.	$\rho(\mathbb{D}_a)$	$\rho(\mathbb{D}_b)$	$\rho(\widehat{\mathbb{D}})$
1	0.64	0.88	0.048
2	0.61	0.96	0.044
3	0.60	0.97	0.052
4	0.60	0.96	0.049
5	0.61	0.96	0.050
6	0.65	0.96	0.077

Table 5.1: The asymptotic AMG convergence factors for (5.5) and (5.9). The first column uses physical constants $\delta t = 0.1$, $\epsilon_0 = 0.1^2$, and $\mu_0 = 1.0$, the second column uses physical constants $\delta t = 0.01$, $\epsilon_0 = 0.1^2$, and $\mu_0 = 1.0$, and the third column is the scaled system and is independent of physical constants. Level 1 is a 4×4 quadrilateral mesh and is refined uniformly 5 times using NI.

Without surprise, the scaled Darwin block performs significantly better than its unscaled counterpart. It should be noted, by examining the formal-normal of the original system (5.6), that the unscaled Darwin system is relatively well behaved for $\delta t \gg 1$. The value of the permittivity constant (in Alfvénic units) is quite mild, $0 < \epsilon_0 < 1$, and only appears in the numerator of first-order off-diagonal terms in the formal-normal. It is the time step, δt, that has the potential to complicate AMG performance. This is seen clearly in the performance difference between $\rho(\mathbb{D}_a)$ and $\rho(\mathbb{D}_b)$. As $\delta t \to 0$, the off-diagonal curl operators dominate. Reasonable AMG factors could be recovered with enough h-refinement, as the second-order Laplacian operators would begin to dominate.

In the next test, a manufactured solution is constructed with a combination of trig functions:

$$\begin{aligned}
E_{r,x} &= \cos(2\pi x/L_x)\sin(\pi y/L_y), \\
E_{r,y} &= \cos(2\pi x/L_x)\cos(\pi y/L_y), \\
E_{r,z} &= \cos(2\pi x/L_x)\sin(\pi y/L_y), \\
B_x &= \cos(2\pi y/L_y)\sin(2\pi x/L_x), \\
B_y &= \cos(2\pi x/L_x)\sin(\pi y/L_y), \\
B_z &= \cos(2\pi x/L_x)\cos(\pi y/L_y), \\
E_{s,x} &= \cos(4\pi x/L_x)\sin(\pi y/L_y), \\
E_{s,y} &= \cos(4\pi x/L_x)\cos(\pi y/L_y), \\
E_{s,z} &= \cos(4\pi x/L_x)\sin(\pi y/L_y)\,.
\end{aligned} \tag{5.11}$$

The true solutions, denoted $\mathbf{u}^*$, are run through the original operator (before scaling) in order to generate a right-hand side. Figure 5.1 shows that the FOSLS functional for both the scaled, $\widehat{\mathbb{D}}$, and unscaled, $\mathbb{D}_b$, Darwin systems decreases as $O(h^2)$ with mesh refinement. The $\mathcal{H}^1$-ellipticity of the Darwin system yields enhanced L^2 convergence using the well-known Aubin-Nitsche duality argument [40]. That is, the L^2-norm of the error converges one order faster than the functional norm of the error. In this case, the L^2-norm of the error decreases as $O(h^3)$.

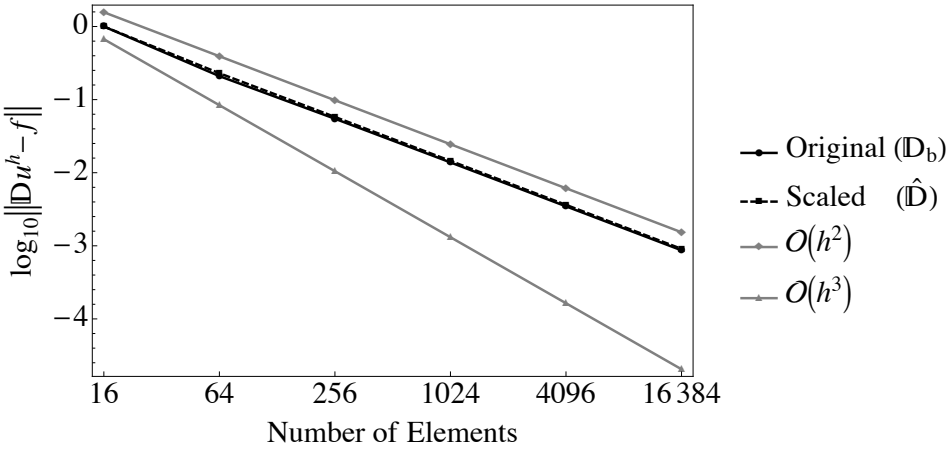

Figure 5.1: The FOSLS functional against the number of elements for the Darwin system. The results are normalized by the first value. The linear system for each level was solved with AMG to a tolerance of 0.5×10^{-2}. Both the scaled ($\hat{\mathbb{D}}$) and unscaled ($\mathbb{D}_b$) functionals decrease with $O(h^2)$.

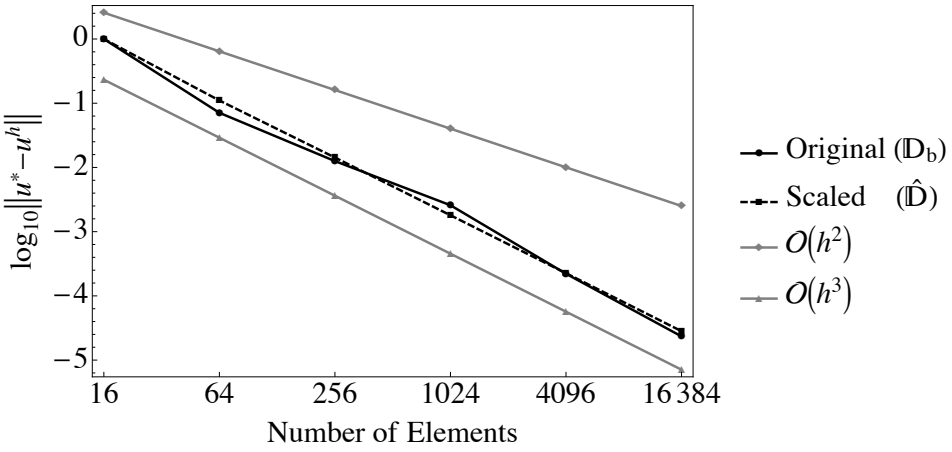

Figure 5.2: The L^2-norm of the error against the number of elements for the Darwin system. The results are normalized by the first value. The linear system for each level were solved with AMG to a tolerance of 0.5×10^{-2}. Both the scaled ($\hat{\mathbb{D}}$) and unscaled ($\mathbb{D}_b$) L^2 errors decrease with $O(h^3)$ as predicted by theory.

Even though the scaled and unscaled systems converge with the same order, the total amount of work performed in the two cases differs substantially. Define a Work Unit (WU) to be the cost of one matrix-vector multiplication on the finest level. The amount of work performed on each level

of the NI process is recorded in Table 5.2. In both the scaled and unscaled cases, the amount of work performed on the coarse grids of the NI process accounts for only a small percent of the the total work. The unscaled system costs approximately 11 times as much.

Lev.	NNZ	WU($\mathbb{D}_b$)	%	WU($\widehat{\mathbb{D}}$)	%
1	3.9×10^4	0.07	0.03	0.01	0.05
2	1.4×10^5	0.79	0.38	0.06	0.31
3	5.4×10^5	2.75	1.35	0.21	1.16
4	2.1×10^6	24.70	14.60	0.91	4.99
5	8.2×10^6	29.61	12.18	3.21	17.66
6	3.3×10^7	144.90	71.44	13.75	75.81
total	—	**202.82**	100	**18.15**	100

Table 5.2: The number of nonzeros in the linear operator (NNZ) and Work Units (WU) on each level of the NI process. A WU is defined as the cost of one matrix-vector multiplication on the finest level. The % column represents the percent of the total work performed on a given level. Tolerance for the AMG solver is set to 0.5×10^{-2}.

5.3 Fluid Blocks

The fluid blocks, $\mathbb{A}_\alpha$, are not naturally $\mathcal{H}^1$-elliptic. The root cause of this is due to the fact that there is no curl term associated with the momentum densities, $\mathbf{p}_\alpha$. The general form of a fluid block, $\mathbb{A}_\alpha$, is examined for arbitrary species α. The isolated fluid block is

$$\begin{bmatrix} \frac{m_\alpha}{\delta t} & T_\alpha \nabla \\ \nabla \cdot & \frac{1}{\delta t} \end{bmatrix} \begin{bmatrix} \mathbf{p}_\alpha \\ n_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{p}_\alpha} \\ g_{n_\alpha} \end{bmatrix}, \tag{5.12}$$

where the notation $T_\alpha = \frac{1}{2} m_\alpha v_{th_\alpha}^2$ is adopted for convenience. The system is modified by introducing the curl of the conservation of momentum equation as an additional equation:

$$\begin{bmatrix} \frac{m_\alpha}{\delta t} & T_\alpha \nabla \\ \frac{m_\alpha}{\delta t} \nabla \times & 0 \\ \nabla \cdot & \frac{1}{\delta t} \end{bmatrix} \begin{bmatrix} \mathbf{p}_\alpha \\ n_\alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{\mathbf{p}_\alpha} \\ \nabla \times \mathbf{f}_{\mathbf{p}_\alpha} \\ g_{n_\alpha} \end{bmatrix}. \tag{5.13}$$

Denote the differential matrix above by $\mathbb{C}_\alpha$. The modified system, $\mathbb{C}_\alpha$, can easily be shown to be fully $\mathcal{H}^1$-elliptic with appropriate boundary conditions. This is proven in Lemma 5.5.4 in Section 5.5. The additional curl constraint forces the momentum density into the smoother space, $\mathcal{H}^1$. This is a reasonable requirement to enforce on momentum density. The formal-normal of $\mathbb{C}_\alpha$ is

$$\mathbb{C}^* \mathbb{C} = \begin{bmatrix} \frac{m_\alpha^2}{\delta t^2} (\mathbb{I} - \Delta) + \left(\frac{m_\alpha^2}{\delta t^2} - 1 \right) \nabla \nabla \cdot & \left(\frac{m_\alpha T_\alpha}{\delta t} - \frac{1}{\delta t} \right) \nabla \\ - \left(\frac{m_\alpha T_\alpha}{\delta t} - \frac{1}{\delta t} \right) \nabla \cdot & \frac{1}{\delta t^2} - T_\alpha^2 \Delta \end{bmatrix}. \tag{5.14}$$

Notice that the upper left block diagonal may become dominated by the $\nabla \nabla \cdot$ term, depending on the size of the physical parameters. A similar approach to that in Section 5.2 is used to scale the block system. Let $\mathbb{R}_\alpha = \text{diag}\,[r_1, r_2, r_3]$ and $\mathbb{S}_\alpha = \text{diag}\,[s_1, s_2]$, where the components of the matrix are different from those used in the Darwin block scaling. Again, the constants in the diagonal scaling matrices are expanded by an appropriately sized identity matrix, which should be obvious from the context of scaling either a vector or scalar quantity. The scaled system takes on the form

$$\mathbb{R}_\alpha \mathbb{C}_\alpha \mathbb{S}_\alpha \left(\mathbb{S}_\alpha^{-1} \mathbf{u}_\alpha \right) = \mathbb{R}_\alpha \mathbf{f}_\alpha,$$

where $\mathbf{f}_\alpha$ is the right-hand side of (5.13). Define $\hat{\mathbf{u}}_\alpha = \left(\mathbb{S}_\alpha^{-1} \right) \mathbf{u}_\alpha$ to be the transformed unknown variables. The formal-normal of the transformed linear system is

$$(\mathbb{R}_\alpha \, \mathbb{C}_\alpha \, \mathbb{S}_D)^* \, (\mathbb{R}_D \, \mathbb{C} \, \mathbb{S}_D) =$$

$$\begin{bmatrix} \frac{m_\alpha^2 r_1^2 s_1^2}{\delta t^2} + \frac{m_\alpha^2 r_2^2 s_1^2}{\delta t^2} \Delta + \left(\frac{m_i^2 r_2^2 s_1^2}{\delta t^2} - r_3^2 s_1^2 \right) \nabla \nabla \cdot & \left(\frac{m_\alpha T_\alpha r_1^2 s_1 s_2}{\delta t} - \frac{r_3^2 s_1 s_2}{\delta t} \right) \nabla \\ - \left(\frac{m_\alpha T_\alpha r_1^2 s_1 s_2}{\delta t} - \frac{r_3^2 s_1 s_2}{\delta t} \right) \nabla \cdot & \frac{r_3^2 s_2^2}{\delta t^2} - T_\alpha s_1^2 s_2^2 \Delta \end{bmatrix}. \tag{5.15}$$

We, again, solve the nonlinear system of equations,

$$\begin{aligned} \tfrac{m_\alpha^2 r_2^2 s_1^2}{\delta t^2} &= 1\,, \\ \tfrac{m_i^2 r_2^2 s_1^2}{\delta t^2} - r_3^2 s_1^2 &= 0\,, \\ \tfrac{m_\alpha T_\alpha r_1^2 s_1 s_2}{\delta t} - \tfrac{r_3^2 s_1 s_2}{\delta t} &= 0\,, \\ T_\alpha s_1^2 s_2^2 &= 1\,, \end{aligned} \tag{5.16}$$

in order to produce a clean system. The system above can be solved uniquely up to the value of r_1 by forcing the all constants to be positive. Setting $r_1 = 1$ produces the left and right scalings

$$\begin{aligned} \mathbb{R}_\alpha &= \text{diag}\left[1, \tfrac{\delta t\sqrt{T_\alpha}}{\sqrt{m_\alpha}}, \sqrt{m_\alpha T_\alpha}\right]\,, \\ \mathbb{S}_\alpha &= \text{diag}\left[\tfrac{1}{\sqrt{m_\alpha T_\alpha}}, \tfrac{1}{T_\alpha}\right]\,. \end{aligned} \tag{5.17}$$

Applying these scalings to the fluid system balances the Div, Grad, Curl blocks:

$$\widehat{\mathbb{C}}_\alpha = \mathbb{R}_\alpha\, \mathbb{C}_\alpha\, \mathbb{S}_\alpha = \begin{bmatrix} \frac{\sqrt{m_\alpha}}{\delta t\sqrt{T_\alpha}} & \nabla \\ \nabla\times & 0 \\ \nabla\cdot & \frac{\sqrt{m_\alpha}}{\delta t\sqrt{T_\alpha}} \end{bmatrix}\,. \tag{5.18}$$

Carrying the same scaling through to the fluid unknowns gives us the transformations $\hat{\mathbf{p}}_\alpha = \sqrt{m_\alpha T_\alpha}\mathbf{p}_\alpha$ and $\hat{n}_\alpha = T_\alpha n_\alpha$. The right-hand side terms are transformed in a similar way. Consider how the scalings modify the formal-normal of $\widehat{\mathbb{C}}_\alpha$:

$$\widehat{\mathbb{C}}_\alpha^*\widehat{\mathbb{C}}_\alpha = \begin{bmatrix} \frac{m_\alpha}{\delta t T_\alpha} - \Delta & 0 \\ 0 & \frac{m_\alpha}{\delta t T_\alpha} - \Delta \end{bmatrix}\,. \tag{5.19}$$

The off-diagonal blocks in the formal-normal have been completely eliminated. For a large variety of standard boundary conditions, it is straightforward to prove that this system is $\mathcal{H}^1$ elliptic. That is, AMG V-Cycles are optimal for (5.18). Using numerical tests, we demonstrate the robustness of the scaled fluid block to variations in physical constants.

5.3.1 Fluid Block Numerical Test

The performance for the curl-modified fluid block, $\mathbb{C}_\alpha$, and the scaled block, $\widehat{\mathbb{C}}_\alpha$, are analyzed in a similar fashion to the Darwin block system in Section 5.2.1. We, again, non-dimensionalize the

system by working in Alfvénic units as outlined in Appendix A.3. The original fluid blocks, $\mathbb{A}_\alpha$, are not $\mathcal{H}^1$-elliptic and standard AMG performs poorly on their discrete systems. The results for these original systems are not presented. Here, the relevant physical constants to be varied are δt, m_α, and T_α. The asymptotic AMG factor, FOSLS functional, and L^2 error are computed for both $\mathbb{C}_\alpha$ and $\widehat{\mathbb{C}}_\alpha$ over the range of values listed in Table 5.3.

Constant	$\mathbb{C}_e^a$	$\mathbb{C}_e^b$	$\mathbb{C}_i^a$	$\mathbb{C}_i^b$
δt	0.01	0.1	0.01	0.1
m_α	5.44×10^{-4}	5.44×10^{-4}	1.0	1.0
T_α	100.0	10.0	5.0	0.5

Table 5.3: Numerical values for four realistic test cases using Alfvénic units. The values are characteristic of a typical electron fluid and ion fluid values. In all cases, the domain is set to $[0, 2] \times [0, 2]$.

The AMG convergence factors for electron scales and ion scales can be seen in Table 5.4 and Table 5.5, respectively. In both cases, the scaled forms show significant improvement over their unscaled versions. The scalings are robust against variation in the physical constants, achieving a peak AMG convergence factor of 0.17 for the electron blocks and 0.14 for the ion blocks.

Lev.	$\rho(\mathbb{C}_e^a)$	$\rho(\widehat{\mathbb{C}}_e^a)$	$\rho(\mathbb{C}_e^b)$	$\rho(\widehat{\mathbb{C}}_e^b)$
1	0.56	0.14	0.77	0.14
2	0.61	0.16	0.89	0.15
3	0.69	0.17	0.91	0.17
4	0.72	0.17	0.94	0.16
5	0.63	0.17	0.96	0.17
6	0.53	0.17	0.96	0.17

Table 5.4: AMG convergence factors for fluid system (5.13) and scaled fluid system (5.18) for electrons. Level 1 is a 4×4 quadrilateral mesh refined uniformly 6 times using NI with quadratic finite elements.

Lev.	$\rho(\mathbb{C}_i^a)$	$\rho(\widehat{\mathbb{C}}_i^a)$	$\rho(\mathbb{C}_i^b)$	$\rho(\widehat{\mathbb{C}}_i^b)$
1	0.36	0.06	0.27	0.08
2	0.60	0.06	0.40	0.07
3	0.75	0.10	0.49	0.11
4	0.72	0.12	0.52	0.13
5	0.86	0.14	0.52	0.13
6	0.92	0.14	0.51	0.13

Table 5.5: AMG convergence factors for fluid system (5.13) and scaled fluid system (5.18) for ions. Level 1 is a 4×4 quadrilateral mesh refined uniformly 6 times using NI with quadratic finite elements.

A manufactured solution is used to show the FOSLS functional and L^2 convergence properties of the fluid system. We, again, pick a solution constructed from simple trig functions that comply

with the conductive boundary conditions (5.2):

$$\begin{aligned} p_{s,x} &= \cos(2\pi x/L_x)\sin(\pi y/L_y)\,, \\ p_{s,x} &= \cos(2\pi x/L_x)\sin(\pi y/L_y)\,, \\ p_{s,x} &= \cos(4\pi x/L_x)\sin(\pi y/L_y)\,, \\ n_s &= 1+\tfrac{1}{4}\sin(2\pi x/L_x)\cos(2\pi y/L_y)\,. \end{aligned} \tag{5.20}$$

For the systems $\mathbb{C}_e^a$, $\mathbb{C}_i^a$, and $\mathbb{C}_i^b$, the convergence results closely follow theory. The FOSLS functionals decrease with $O(h^2)$ while L^2-errors gain an extra power and decrease as $O(h^3)$. In the case for the more difficult scalings in $\mathbb{C}_e^b$, the results are not as clean. The FOSLS functional is shown in Figure 5.3, where it is seen that both the scaled and unscaled systems still decrease with $O(h^2)$. But the L^2-norm of the error, shown in Figure 5.3, only converges with $O(h^2)$ for the unscaled system.

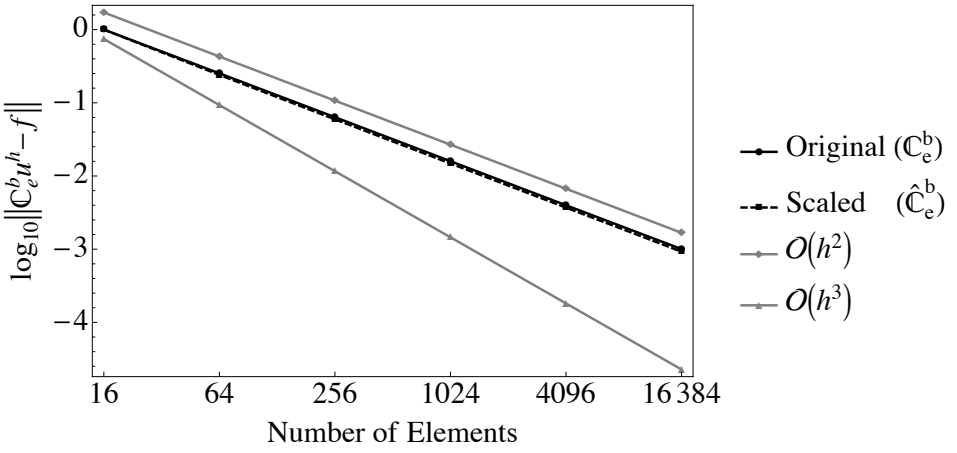

Figure 5.3: The FOSLS functional against the number of elements for the fluid system, $\mathbb{C}_e^b$. The results are normalized by the first value. The linear system for each level was solved with AMG to a tolerence of 0.5×10^{-2}. Both the scaled ($\hat{\mathbb{C}}_e^b$) and unscaled ($\mathbb{C}_e^b$) functionals decrease with $O(h^2)$.

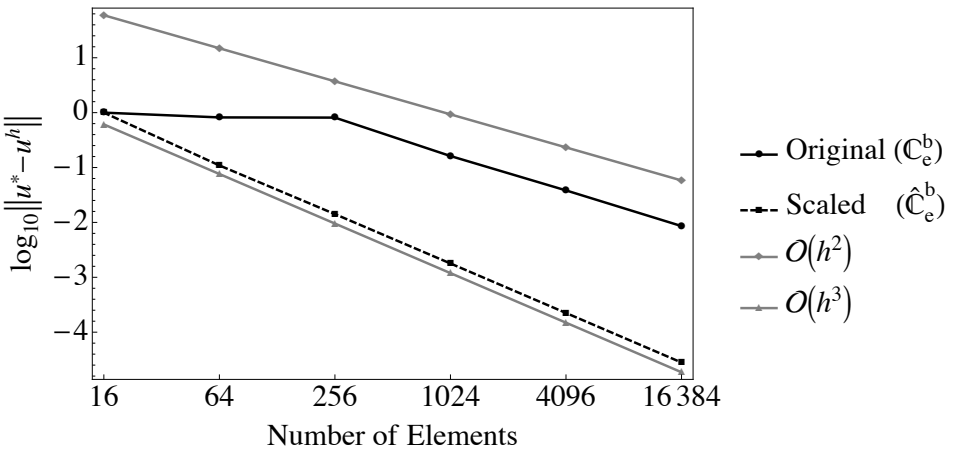

Figure 5.4: The L^2-norm of the error against the number of elements for the fluid system. The results are normalized by the first value. The linear system for each level were solved with AMG to a tolerence of 0.5×10^{-2}. The L^2 error for $\widehat{\mathbb{D}}$ decreases, as predicted by theory, with $O(h^3)$, while the L^2 error for the unscaled system, $\mathbb{C}_e^b$, decreases with only $O(h^2)$.

The lack of an enhanced L^2 convergence can be rationalized by careful examination of the formal-normal of the unscaled system, (5.14). In the upper diagonal block, the small value of m_e and relatively large value of δt cause the ratio $\frac{m_e^2}{\delta t^2} \approx 3 \times 10^{-5}$. Until enough h-refinement occurs, the upper diagonal block will be dominated by $\nabla\nabla\cdot$, creating a large near null-space component. Interestingly enough, in our 2D3V description, the divergence operator only acts upon the first two components of $\mathbf{p}_e$. If the L^2-error is viewed component-wise, the x and y components of $\mathbf{p}_e$ decrease with $O(h^2)$ and the z component of $\mathbf{p}_e$ and n_e decrease with $O(h^3)$. This is validated by Figure 5.5, where we see the predicted behavior. After enough h-refinement is performed, the x and y components of momentum should begin to decrease with $O(h^3)$, as theory predicts. This trend is seen near the end of the simulation. More refinement needs to be performed in order to pass more concrete judgment. Unfortunately, computer hardware limits were reached before such trends would be observed. This remains the subject of future work (See Chapter 7).

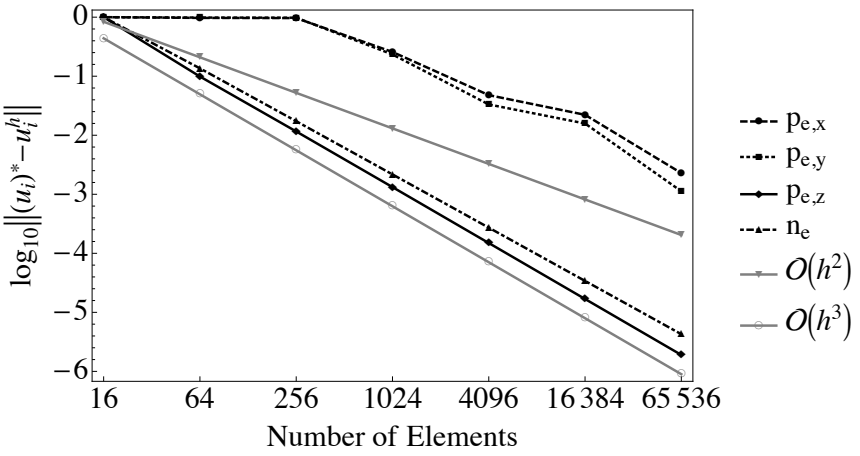

Figure 5.5: The L^2-error of each component against the number of elements for the fluid system. This is the same data as seen in Figure 5.4 separated into component form. The results are normalized by each components first value. The $p_{e,z}$ and n_e components decrease, as predicted by theory, with $O(h^3)$, while the $p_{e,z}$, and $p_{i,z}$, components only decrease with $O(h^2)$.

We compare the amount of work performed during the uniform NI process for each of the above mentioned cases. The electron blocks are seen in Table 5.6 and the ion blocks are seen in Table 5.7. The scaled operators perform significantly better. The ratio of total work performed on the unscaled systems to the total amount of work performed on the scaled systems are 3.56, 35.6, 7.57, and 1.88 for $\mathbb{C}_e^a$, $\mathbb{C}_e^b$, $\mathbb{C}_i^a$, and $\mathbb{C}_i^b$, respectively. The most gain is seen in the electron system, $\mathbb{C}_e^b$. Scaling the electron systems is crucial.

Lev.	NNZ	WU($\mathbb{C}_e^a$)	%	WU($\widehat{\mathbb{C}}_e^a$)	%	WU($\mathbb{C}_e^b$)	%	WU($\widehat{\mathbb{C}}_e^b$)	%
1	1.6×10^4	0.01	0.01	0.01	0.04	0.01	0.001	0.01	0.04
2	6.0×10^4	0.11	0.20	0.05	0.29	0.10	0.01	0.07	0.29
3	2.3×10^5	0.61	1.03	0.19	1.14	0.45	0.05	0.24	1.13
4	8.8×10^5	3.80	6.31	0.81	4.80	26.77	3.4	1.07	4.80
5	3.5×10^6	12.10	20.11	3.07	18.21	179.0	22.74	4.02	18.20
6	1.4×10^7	43.51	72.34	12.73	75.49	580.6	73.77	16.65	75.47
total	—	**60.14**	100	**16.87**	100	**786.9**	100	**22.06**	100

Table 5.6: The number of nonzeros in the linear operator (NNZ) and Work Units (WU) on each level of the NI process. A WU is defined as the cost of one matrix-vector multiplication on the finest level. The % column represents the percent of the total work performed on a given level. Tolerance for the AMG solver is set to 0.5×10^{-2}.

Lev.	NNZ	WU($\mathbb{C}_i^a$)	%	WU($\widehat{\mathbb{C}}_i^a$)	%	WU($\mathbb{C}_i^b$)	%	WU($\widehat{\mathbb{C}}_i^b$)	%
1	1.6×10^4	0.02	0.01	0.01	0.06	0.02	0.07	0.01	0.05
2	6.0×10^4	0.08	0.20	0.03	0.17	0.09	0.27	0.03	0.19
3	2.3×10^5	0.49	1.03	0.11	1.68	0.36	1.13	0.15	0.92
4	8.8×10^5	1.80	6.31	0.80	4.73	1.10	3.38	0.57	3.30
5	3.5×10^6	9.92	20.11	3.07	17.98	7.30	22.46	3.20	18.48
6	1.4×10^7	117.1	72.34	13.05	76.34	23.64	72.68	13.31	77.02
total	—	**129.41**	100	**17.09**	100	**32.52**	100	**17.28**	100

Table 5.7: The number of nonzeros in the linear operator (NNZ) and Work Units (WU) on each level of the NI process. A WU is defined as the cost of one matrix-vector multiplication on the finest level. The % column represents the percent of the total work performed on a given level. Tolerance for the AMG solver is set to 0.5×10^{-2}.

5.4 Full TFP System

Following the scalings and modifications outlined in Section 5.2 and Section 5.3 through the entire system, (5.1), yields

$$\hat{\mathcal{F}}(\hat{\mathbf{u}}) = \begin{bmatrix} \widehat{\mathbb{C}}_i & 0 & 0 \\ 0 & \widehat{\mathbb{C}}_e & 0 \\ \widehat{\mathbb{L}}_i & \widehat{\mathbb{L}}_e & \widehat{\mathbb{D}} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{u}}_i \\ \hat{\mathbf{u}}_e \\ \hat{\mathbf{u}}_D \end{bmatrix} + \begin{bmatrix} 0 & 0 & \widehat{\mathbb{U}}_i(\mathbf{u}_i, \mathbf{u}_D) \\ 0 & 0 & \widehat{\mathbb{U}}_e(\mathbf{u}_e, \mathbf{u}_D) \\ 0 & 0 & 0 \end{bmatrix}, \tag{5.21}$$

where $\widehat{\mathbb{C}}_\alpha$ and $\widehat{\mathbb{D}}$ are defined previously. As a reminder, $\hat{\mathbf{u}}_i = [\hat{\mathbf{p}}_i, \hat{n}_i]^T$, $\hat{\mathbf{u}}_e = [\hat{\mathbf{p}}_e, \hat{n}_e]^T$, and $\hat{\mathbf{u}}_D = \left[\widehat{\mathbf{E}}_r, \widehat{\mathbf{B}}, \widehat{\mathbf{E}}_s\right]^T$. The scaled linear lower off-diagonal blocks are

$$\widehat{\mathbb{L}}_\alpha = \begin{bmatrix} 0 & 0 \\ 0 & -\frac{q_\alpha}{\epsilon_0 T_\alpha} \\ -\frac{q_\alpha \delta t}{\epsilon_0 \sqrt{m_\alpha T_\alpha}} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}, \tag{5.22}$$

and the scaled nonlinear upper off-diagonal blocks are

$$\widehat{\mathbb{U}}_\alpha(\mathbf{u}_\alpha, \mathbf{u}_D) = \begin{bmatrix} -\frac{q_\alpha}{T_\alpha}\hat{n}_\alpha \widehat{\mathbf{E}}_r & -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t \sqrt{m_\alpha T_\alpha}} \hat{\mathbf{p}}_\alpha \times \widehat{\mathbf{B}} & -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t^2 T_\alpha}\hat{n}_\alpha \widehat{\mathbf{E}}_s \\ -\frac{q_\alpha \delta t}{\sqrt{m_\alpha T_\alpha}} (\nabla \hat{n}_\alpha) \times \widehat{\mathbf{E}}_r & -\frac{q_\alpha \epsilon_0 \mu_0}{m_\alpha} \nabla \times (\hat{\mathbf{p}}_\alpha \times \widehat{\mathbf{B}}) & -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t \sqrt{m_\alpha T_\alpha}} \nabla \times (\hat{n}_\alpha \widehat{\mathbf{E}}_s) \end{bmatrix}, \tag{5.23}$$

for $\alpha = i, e$. The unknowns and right-hand-side functions are also scaled appropriately, as outlined in previous sections.

The Fréchet derivative used in the Newton step (4.3) becomes

$$\hat{\mathcal{F}}'(\hat{\mathbf{u}})[\delta] = \begin{bmatrix} \widehat{\mathbb{C}}_i & 0 & 0 \\ 0 & \widehat{\mathbb{C}}_e & 0 \\ \widehat{\mathbb{L}}_i & \widehat{\mathbb{L}}_e & \widehat{\mathbb{D}} \end{bmatrix} \begin{bmatrix} \delta_i \\ \delta_e \\ \delta_D \end{bmatrix} + \begin{bmatrix} \widehat{\mathbb{U}}'_{ii} & 0 & \widehat{\mathbb{U}}'_{i_D} \\ 0 & \widehat{\mathbb{U}}'_{ee} & \widehat{\mathbb{U}}'_{e_D} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_i \\ \delta_e \\ \delta_D \end{bmatrix}, \tag{5.24}$$

where the action of the linearized terms are

$$\widehat{\mathbb{U}}'_{\alpha\alpha}\mathbf{u}_\alpha = \left[\begin{array}{c|c} -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t \sqrt{m_\alpha T_\alpha}} \widehat{\mathbf{B}} \times \circ & -(\frac{q_\alpha}{T_\alpha}\widehat{\mathbf{E}}_r + \frac{q_\alpha \epsilon_0 \mu_0}{\delta t^2 T_\alpha}\widehat{\mathbf{E}}_s)\circ \\ \frac{q_\alpha \epsilon_0 \mu_0}{m_\alpha} \nabla \times (\widehat{\mathbf{B}} \times \circ) & -\frac{q_\alpha \delta t}{\sqrt{m_\alpha T_\alpha}} (\nabla \circ) \times \widehat{\mathbf{E}}_r \; -\frac{q_\alpha \epsilon_0 \mu_0}{\delta t \sqrt{m_\alpha T_\alpha}} \nabla \times (\widehat{\mathbf{E}}_s \circ) \\ 0 & 0 \end{array}\right] \begin{bmatrix} \mathbf{p}_\alpha \\ n_\alpha \end{bmatrix},$$

$$\widehat{\mathbb{U}}'_{\alpha_D}\mathbf{u}_D = \left[\begin{array}{c|c|c} -\frac{q_\alpha}{T_\alpha}\hat{n}_\alpha \circ & -\frac{q_\alpha\epsilon_0\mu_0}{\delta t\sqrt{m_\alpha T_\alpha}}\hat{\mathbf{p}}_\alpha \times \circ & -\frac{q_\alpha\epsilon_0\mu_0}{\delta t^2 T_\alpha}\hat{n}_\alpha \circ \\ & & \\ \frac{q_\alpha\delta t}{\sqrt{m_\alpha T_\alpha}}\nabla\hat{n}_\alpha \times \circ & -\frac{q_\alpha\epsilon_0\mu_0}{m_\alpha}\nabla\times(\hat{\mathbf{p}}_\alpha \times \circ) & -\frac{q_\alpha\epsilon_0\mu_0}{\delta t\sqrt{m_\alpha T_\alpha}}\nabla\times(\hat{n}_\alpha \circ) \end{array}\right]\left[\begin{array}{c}\mathbf{E}_r\\ \mathbf{B}\\ \mathbf{E}_s\end{array}\right],$$

for $\alpha = i, e$. The placeholder, $\circ$, indicates the input to the operator and is adopted in order to clearly indicate the order of operations.

5.5 Uniform $\mathcal{H}^1$ Ellipticity

An important property of the operator, $\hat{\mathcal{F}}'(\hat{\mathbf{u}})$, is that it is uniformly coercive and continuous in a convenient norm for all $\hat{\mathbf{u}}$ in a neighborhood of the exact solution, $\hat{\mathbf{u}}_*$. This guarantees the existence of each Newton step once the approximation is sufficiently accurate. Below, it is shown that, under mild hypotheses, $\hat{\mathcal{F}}'(\mathbf{u})$ is uniformly $\mathcal{H}^1$-elliptic in a neighborhood of the exact solution. Define the following spaces:

$$\begin{aligned} L^2_0(\Omega) &= \left\{p \in L^2(\Omega) : \textstyle\int_\Omega p\,\mathrm{d}x = 0\right\}, \\ \mathcal{H}^1_0(\Omega) &= \left\{p \in L^2(\Omega) : \nabla p \in \left(L^2(\Omega)\right)^3 \text{ and } p = 0 \text{ on } \partial\Omega\right\}, \\ \mathcal{H}(\mathrm{Div},\Omega) &= \left\{\mathbf{v} \in \left(L^2(\Omega)\right)^3 : \nabla\cdot\mathbf{v} \in L^2(\Omega)\right\}, \\ \mathcal{H}(\mathrm{Curl},\Omega) &= \left\{\mathbf{v} \in \left(L^2(\Omega)\right)^3 : \nabla\times\mathbf{v} \in \left(L^2(\Omega)\right)^3\right\}, \\ \mathcal{H}_0(\mathrm{Div},\Omega) &= \left\{\mathbf{v} \in \mathcal{H}(\mathrm{Div},\Omega) : \hat{\mathbf{n}}\cdot\mathbf{v} = 0 \text{ on } \partial\Omega\right\}, \\ \mathcal{H}_0(\mathrm{Curl},\Omega) &= \left\{\mathbf{v} \in \mathcal{H}(\mathrm{Curl},\Omega) : \hat{\mathbf{n}}\times\mathbf{v} = \mathbf{0} \text{ on } \partial\Omega\right\}, \end{aligned} \tag{5.25}$$

where $\hat{\mathbf{n}}$ is the outward unit normal on the boundary. Let the open ball of radius r centered on $\hat{\mathbf{u}}_*$ be defined by

$$B_r(\hat{\mathbf{u}}_*) = \left\{\hat{\mathbf{u}} \in \mathcal{V} \;:\; \|\hat{\mathbf{u}} - \hat{\mathbf{u}}_*\|_{\mathcal{H}^1} < r\right\}. \tag{5.26}$$

It was shown in Section 5.3 that $\widehat{\mathbb{C}}_i$, $\widehat{\mathbb{C}}_e$, and $\widehat{\mathbb{D}}$ are coercive and continuous in $\mathcal{H}^1$. Here, it is further assumed that the block diagonals, $\widehat{\mathbb{C}}_i + \widehat{\mathbb{U}}'_{ii}$ and $\widehat{\mathbb{C}}_e + \widehat{\mathbb{U}}'_{ee}$, of linear system (5.24) are also $\mathcal{H}^1$-elliptic.

We begin by outlining a handful of lemmas that are used for the main coercivity result, Theorem 5.5.1. Lemmas are stated without proof and the reader is directed to [25] and [11] for

more rigorous accounts. The following lemma is what allows zeroth-order off-diagonal terms to be overlooked during the coercivity proof.

Lemma 5.5.1. Standard Compactness Argument (SCA)

Take $\mathcal{L} : \mathcal{V} \to \mathcal{W}$ as a one-to-one, bounded, linear operator. Assume further that $\mathcal{V}$ and $\mathcal{W}$ are Hilbert spaces such that $\mathcal{V}$ is compactly embedded in $\mathcal{W}$. If there exists a constant, $C > 0$, such that

$$||u||_{\mathcal{V}} \leq C\left(||\mathcal{L}u||_{\mathcal{W}} + ||u||_{\mathcal{W}}\right) \ \forall\ u \in \mathcal{V}\,,$$

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

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

To understand how to apply the SCA, let $\mathcal{L} : \mathcal{V} \to \mathcal{W}$ be a bounded linear operator, with $\mathcal{V}$ and $\mathcal{W}$ as defined above. Consider the case when $\mathcal{L}$ is of the form $\mathcal{L} = \mathcal{L}_0 + E$, where E consists entirely of zeroth-order off-diagonal terms and $\mathcal{L}_0$ is coercive in $\mathcal{W}$. By the coercivity of $\mathcal{L}_0$, there exists a constant, c, such that

$$||u||_{\mathcal{V}} \leq c||\mathcal{L}_0 u||_{\mathcal{W}}\,.$$

Using simple inequalities, it is seen that

$$||u||_{\mathcal{V}} \leq c||\mathcal{L}u - Eu||_{\mathcal{W}} \leq c||\mathcal{L}u||_{\mathcal{W}} + c||Eu||_{\mathcal{W}} \leq C\left(||\mathcal{L}u||_{\mathcal{W}} + ||u||_{\mathcal{W}}\right)\,.$$

Finally, invoking the SCA guarantees that there exists a constants, $\hat{C}$, such that

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

That is, the operator, $\mathcal{L}$, that contains the zeroth-order off-diagonal elements is coersive in $\mathcal{W}$ as well. This property is what grants us the ability to turn a blind eye to the $\widehat{\mathbb{L}}_\alpha$ blocks and the first row of the $\widehat{\mathbb{U}}'_{\alpha D}$ blocks during statement of Theorem 5.5.1.

The next three lemmas are the main work horse for proving coercivity. The Poincaré inequality is a powerful result that bounds the L^2-norm of a function above by a constant, dependent only upon the domain geometry, and the L^2-norm of the function's gradient. The Div-Curl and

Div-Grad-Curl coercivity follow from careful application of the Poincaré inequality. They are the extension of the Poincaré inequality that, given appropriate boundary conditions, grant coercivity to blocks of the form

$$\begin{bmatrix} \nabla\cdot \\ \nabla\times \end{bmatrix} \mathbf{F} \tag{5.27}$$

and

$$\begin{bmatrix} \mathbb{I} & \nabla \\ \nabla\times & 0 \\ \nabla\cdot & \mathbb{I} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \phi \end{bmatrix}, \tag{5.28}$$

where $\mathbf{F}$ represents a general vector quantity and ϕ represents a general scalar quantity. Notice that (5.27) is characteristic of a piece of the Darwin system and (5.28) is characteristic of the fluid blocks.

Lemma 5.5.2. Poincaré Inequality

For each $\phi \in L_0^2(\Omega)$, there exists a consants, $C > 0$, such that

$$\|\phi\|_0 \leq \sqrt{\text{diam}(\Omega)}\|\nabla\phi\|_0,$$

where $\text{diam}(\Omega)$ *is the diameter of* Ω.

Lemma 5.5.3. Div-Curl Coercivity

Take Ω to be a simply connected, bounded, convex domain (smooth or polyhedron). Then, for any $\mathbf{v} \in \mathcal{H}_0(Div,\Omega) \cap \mathcal{H}(Curl,\Omega)$ or $\mathbf{v} \in \mathcal{H}(Div,\Omega) \cap H_0(Curl,\Omega)$, there exists a $C \in \mathbb{R}^+$ such that

$$||\mathbf{v}||_1^2 \leq C\left(||\nabla\cdot\mathbf{v}||^2 + ||\nabla\times\mathbf{v}||^2\right).$$

This is equivalent to saying that

$$\mathcal{H}_0(Div,\Omega) \cap \mathcal{H}(Curl,\Omega) \subseteq \mathcal{H}^1(\Omega)$$

or

$$\mathcal{H}(Div,\Omega) \cap H_0(Curl,\Omega) \subseteq \mathcal{H}^1(\Omega).$$

Lemma 5.5.4. Div-Grad-Curl Coercivity

Take Ω to be a simply connected, bounded, convex domain (smooth or polyhedron). For any $\mathbf{v} \in \mathcal{H}_0(Div,\Omega) \cap \mathcal{H}(Curl,\Omega)$ and for $\alpha \in \mathcal{H}_0^1(\Omega)$, there exists a $C \in \mathbb{R}^+$ such that

$$||\alpha||_1^2 + ||\mathbf{v}||_1^2 \leq C\left(||\mathbf{v} + \nabla\alpha||^2 + ||\nabla \times \mathbf{v}||^2 + ||\nabla \cdot \mathbf{v} + \alpha||^2\right) .$$

Proof. Expand the first and last term on the right-hand side of the coercivity inequality:

$$||\mathbf{v} + \nabla\alpha||^2 + ||\nabla \cdot \mathbf{v} + \alpha||^2 = ||\mathbf{v}||^2 + ||\nabla\alpha||^2 + ||\nabla \cdot \mathbf{v}||^2 + ||\alpha||^2 + 2\left(\langle \mathbf{v}, \nabla\alpha\rangle + \langle \nabla \cdot \mathbf{v}, \alpha\rangle\right) .$$

Using integration by parts and applying the boundary conditions on $\mathbf{v}$ gives that

$$\langle \mathbf{v}, \nabla\alpha\rangle + \langle \nabla \cdot \mathbf{v}, \alpha\rangle = \int_{\partial\Omega} (\mathbf{v}\alpha) \cdot \hat{\mathbf{n}} \, \mathrm{d}s = 0 .$$

Direct application of Lemma 5.5.3 proves the results. □

The the following theorem is the main result of this section. It proves that the linearization of the scaled scaled TFP-Darwin system, (5.1), subject to boundary conditions (5.2), is elliptic in $\mathcal{H}^1$.

Theorem 5.5.1. *Let Ω be a convex domain with piecewise $\mathcal{C}^{1,1}$ boundary. Assume that there exist $r > 0$ and positive constants, $c_\alpha(r)$ and $C_\alpha(r)$, such that, for every $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$ and for every $\hat{\delta}_\alpha \in (\mathcal{H}^1)^4$,*

$$c_\alpha(r)\|\hat{\delta}_\alpha\|_{\mathcal{H}^1} \leq \|(\widehat{\mathbb{C}}_\alpha + \widehat{\mathbb{U}}'_{\alpha\alpha}(\hat{\mathbf{u}}))\hat{\delta}_\alpha\| \leq C_\alpha(r)\|\hat{\delta}_\alpha\|_{\mathcal{H}^1}, \tag{5.29}$$

for $\alpha = i, e$. Further, assume that $\hat{\mathcal{F}}'(\hat{\mathbf{u}})$ is injective for every $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$. Then there exist positive constants, $c(r)$ and $C(r)$, such that, for every $\hat{\mathbf{u}} \in B_r(\hat{\mathbf{u}}_*)$ and for every $\hat{\delta} \in (\mathcal{H}^1)^{17}$,

$$c(r)\|\hat{\delta}\|_{\mathcal{H}^1} \leq \|\hat{\mathcal{F}}'(\hat{\mathbf{u}})[\hat{\delta}]\| \leq C(r)\|\hat{\delta}\|_{\mathcal{H}^1}. \tag{5.30}$$

Proof. The upper bound in (5.30) is easily obtained by successive application of the triangle inequality and, for the sake of brevity, is not included. The proof of the lower bound requires more work, leaning heavily on all of the above lemmas.

Consider, first, the ellipticity of $\widehat{\mathbb{D}}$ (see (5.9)). This is proved by applying the SCA, Lemma 5.5.1, to remove the identity blocks. The coercivity and continuity constants of the remaining Div-Curl block are then established by direct application of Lemma 5.5.3. Let $c_D, C_D > 0$ be the constants such that

$$c_D||\hat{\mathbf{u}}_D||_{\mathcal{H}^1} \leq ||\widehat{\mathbb{D}}\hat{\mathbf{u}}_D|| \leq C_D||\hat{\mathbf{u}}_D||_{\mathcal{H}^1} \,. \tag{5.31}$$

Let $\hat{\mathbf{u}} \in B_r(\mathbf{u}_*)$. By assumption (5.29) and (5.31), the operator containing only the block diagonals,

$$\begin{bmatrix} \widehat{\mathbb{C}}_i + \widehat{\mathbb{U}}'_{ii}(\hat{\mathbf{u}}) & 0 & 0 \\ 0 & \widehat{\mathbb{C}}_e + \widehat{\mathbb{U}}'_{ee}(\hat{\mathbf{u}}) & 0 \\ 0 & 0 & \widehat{\mathbb{D}} \end{bmatrix},$$

is $\mathcal{H}^1$-elliptic with constants $c_{_B} = \min[c_\alpha(r), c_{_D}]$, $C_{_B} = \max[C_\alpha, C_{_D}]$.

It is clear that the upper blocks satisfy

$$\|\widehat{\mathbb{U}}_{\alpha_D}\hat{\delta}_{_D}\| \leq C_{\alpha_D}\|\hat{\delta}_{_D}\|_{\mathcal{H}^1} \leq C_{\alpha_D}/c_{_D}\|\widehat{\mathbb{D}}\hat{\delta}_{_D}\|, \tag{5.32}$$

for some constants, $C_{\alpha_D}(\hat{\mathbf{u}}) > 0$, for $\alpha = i, e$. Now, consider the system with the strictly lower-triangular blocks removed. It is straightforward to establish that

$$\left\| \begin{bmatrix} \widehat{\mathbb{C}}_i + \widehat{\mathbb{U}}'_{ii} & 0 & 0 \\ 0 & \widehat{\mathbb{C}}_e + \widehat{\mathbb{U}}'_{ee} & 0 \\ 0 & 0 & \widehat{\mathbb{D}} \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_{_D} \end{bmatrix} \right\| \leq \eta \left\| \begin{bmatrix} \widehat{\mathbb{C}}_i + \widehat{\mathbb{U}}'_{ii} & 0 & \widehat{\mathbb{U}}'_{i_D} \\ 0 & \widehat{\mathbb{C}}_e + \widehat{\mathbb{U}}'_{ee} & \widehat{\mathbb{U}}'_{e_D} \\ 0 & 0 & \widehat{\mathbb{D}} \end{bmatrix} \begin{bmatrix} \hat{\delta}_i \\ \hat{\delta}_e \\ \hat{\delta}_{_D} \end{bmatrix} \right\|,$$

with $\eta = (1 + \max[C_{\alpha_D}]/c_{_D})$. This proves the lower bound for the upper block triangular part of the system. The proof is completed by noting that the $\widehat{\mathbb{L}}_\alpha$ blocks only contain zeroth-order terms and by applying the SCA, Lemma 5.5.1. □

Full $\mathcal{H}^1$ellipticity implies that standard, $\mathcal{H}^1$-conforming, finite element spaces can be used and standard convergence bounds apply. In the linear case, enhanced L^2 convergence is also ensured [40]. That is, the L^2-norm of the error converges one order faster than the functional norm of the error. This is observed in the numerical tests that follow.

Chapter 6

Numerical Results

In this chapter, we begin by analyzing the spatial portion of the TFP-Darwin operator, looking at a fixed implicit time step and manufacturing a solution. The manufactured solutions are used to validate the theory developed in Chapter 5. Two are tested. The first is constructed from sine and cosine functions and is used as a baseline test. The second is built with a steep gradient in current density, simulating a more physically realistic problem. The FOSLS functional and L^2-error are shown for each case.

Next, an implicit time-stepping algorithm (BDF2) is demonstrated. A simple wave is set up and propagated. As the wave travels, gradients in the number density begin to develop, causing degraded performance of the unscaled system. The scaled version, however, remains unphased and the number of WUs per time step remains low.

Finally, the more realistic magnetic reconnection problem is simulated. Two current islands are placed next to one another. They are perturbed from an equilibrium solution and begin to attract. Upon collision, a current spike develops that characterizes a magnetic reconnection rate. The TFP-Darwin model stems from a the collisionless Vlasov equation and is, thus, unable to capture true reconnection. Needless to say, the performance of the solver under these realistic conditions is exceptional, requiring an average of approximately 70 WUs per time step.

6.1 Manufactured Solutions

In this section, performance results for the TFP-Darwin model, solved with NI, are presented. No time-stepping algorithm is implemented. Instead, focus is given to the system generated for a fixed implicit time step by manufacturing solutions and analyzing the NI-AMG solver performance and FOSLS convergence properties. Two different problems are analyzed. In the first, a simple smooth solution is constructed from sines and cosines. In the second, a solution is manufactured with a sharp gradient in the z-component of momentum density and a large circulation in the x and y components of the magnetic field. Large features in the fluid variables produce substantial nonlinearities in off-diagonal components, as appear in (5.23). It is noted that neither of these manufactured solutions are physical but, instead, are used to demonstrate how the nonlinear solver, NI, and AMG perform in an environment where the exact error can be measured.

The problems are solved on the computational domain $\Omega = [-1, 1] \times [-1, 1]$. The solution is taken to be periodic on the east and west boundaries and purely conductive on the north and south boundaries. The north and south boundaries are the same as in (5.2). The system is non-dimensionalized against ion mass m_i, fundamental charge e, ion inertial length d_i, ion number density n_0, and the ion cyclotron frequency Ω_i. (See Appendix A.3 for details of these Alfvénic units.) Unless otherwise specified, take $\frac{m_e}{m_i} = 5.44 \times 10^{-4}$, $T_i = 1$, $T_e = 20$, $\epsilon_0 = 0.3^2$, and $\mu = 1.0$. In all tests, a time step of $\delta t = 0.1$ is used.

The coarsest problem, Level 0, is a 4×4 mesh discretized with bilinear finite elements. A random initial guess is made and Newton-FOSLS is performed until the desired tolerance is reached. Using p-refinement, the initial guess for Level 1 is constructed from the approximate solution on Level 0, and standard NI-Newton-FOSLS, as outlined in Chapter 4, is continued until the finest level is reached.

6.1.1 Baseline Test

In this test a simple, smooth solution is constructed from sines and cosines. The number densities are taken to be small perturbations away from 1 and most of the momentum and field quantities are taken to be small perturbations away from 0. The exception is the quantities $p_{i,z}$, $p_{e,z}$, B_x, and B_y, which are $O(1)$. We prescribe the exact solution to be

$$\begin{aligned}
p_{i,x,y} &= \tfrac{1}{10}\sin(\pi x)\sin(\pi y)\ , \\
p_{i,z} &= \sin(\pi x)\sin(\pi y)\ , \\
n_i &= 1+\tfrac{1}{10}\cos(2\pi x)\cos(2\pi y)\ , \\
p_{e,x,y} &= -\tfrac{1}{10}\sin(\pi x)\sin(\pi y)\ , \\
p_{e,z} &= -\sin(\pi x)\sin(\pi y)\ , \\
n_e &= 1+\tfrac{1}{10}\cos(2\pi x)\cos(2\pi y)\ , \\
E_{r,x} &= \tfrac{1}{10}\cos(2\pi x)\sin(\pi y)\ , \\
E_{r,y} &= \tfrac{1}{10}\sin(\pi x)\cos(2\pi y)\ , \\
E_{r,z} &= \tfrac{1}{10}\sin(\pi x)\sin(\pi y)\ , \\
B_x &= \sin(\pi x)\cos(2\pi y)\ , \\
B_y &= \cos(2\pi x)\sin(\pi y)\ , \\
B_z &= \tfrac{1}{10}\cos(2\pi x)\cos(2\pi y)\ , \\
E_{s,x} &= \tfrac{1}{10}\cos(2\pi x)\sin(\pi y)\ , \\
E_{s,y} &= \tfrac{1}{10}\sin(\pi x)\cos(2\pi y)\ , \\
E_{s,z} &= \tfrac{1}{10}\sin(\pi x)\sin(\pi y)\ .
\end{aligned} \tag{6.1}$$

The asymptotic AMG convergence factors, ρ, for each level are seen in Table 6.1. These factors are computed by setting the AMG solver tolerance to 10^{-6}, effectively allowing AMG V-cycles to continue until a stable convergence factor is reached. This is done independently from the NI-Newton-FOSLS computations. Several Newton iterations are performed on each level, each with its own asymptotic factor. Denote ρ_i as the asymptotic convergence for each Newton iteration i. The factors hold between 0.50 - 0.60. A substantial quantity of Newton iterations are performed

on Level 0, but their cost relative to the rest of the iterations is negligible and, thus, not included in the table. Averaging over all V-cycles for every Newton iteration on Level 0 gives an average asymptotic convergence factor of 0.35.

Lev.	$\rho_1(\hat{\mathcal{F}}')$	$\rho_2(\hat{\mathcal{F}}')$
1	0.51	0.57
2	0.57	0.62
3	0.49	0.48
4	0.49	0.52
5	0.53	0.59
6	0.59	—

Table 6.1: Asymptotic AMG convergence factors for Problem (6.1) for the final, scaled, nonlinear TFP-Darwin system. A relative AMG solver tolerance of 10^{-6} was used. Level 1 is a 4×4 quadrilateral meshed discretized with quadratic elements. ρ_i denotes the AMG convergene factor produced for Newton iteration i. In this case, the maximum number of Newton steps required on all levels was 2.

The normalized nonlinear FOSLS functional values of $||\mathcal{F}(\mathbf{u}^h)||$ are seen in Figure 6.1, where $\mathbf{u}^h$ represents the discrete solution at each level. The values are normalized such that the initial FOSLS functional on Level 1 has a value of 1. Initially, the functional is decreasing at a rate of nearly $O(h^3)$ but, as refinement continues, it begins to approach $O(h^2)$.

The L^2-error of the solution after each Newton-FOSLS process is plotted in Figure 6.2. The values are normalized such that the the initial L^2-error on Level 1 has a value of 1. The error is decreasing at a rate of $O(h^3)$. As expected, the rate of L^2-convergence is one order higher than that of the FOSLS functional. This enhanced L^2 convergence rate confirms the result of Theorem 5.5.1; the linearized TFP-Darwin system is $\mathcal{H}^1$-elliptic.

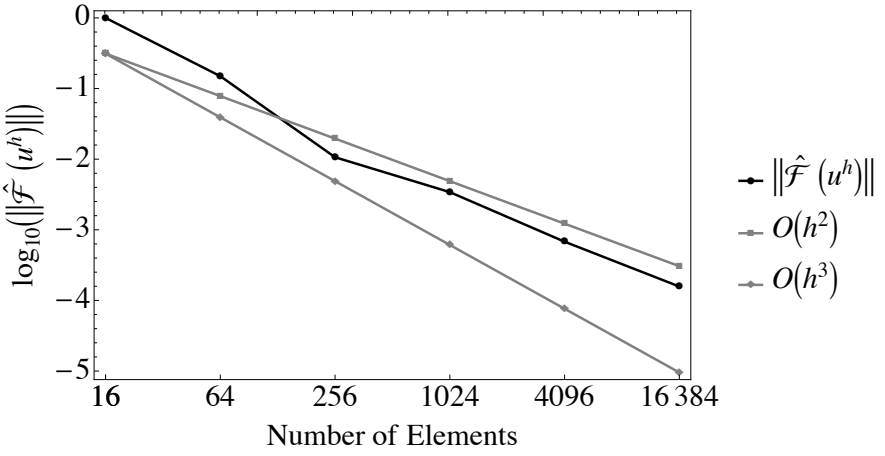

Figure 6.1: The nonlinear FOSLS functionals for Problem (6.1) through 6 levels of NI with uniform refinement. The tolerance of the AMG Solver is 0.5×10^{-1} and the tolerance of the Newton iteration is 10^{-1}. The value of h is defined as $\frac{1}{\sqrt{N_e}}$, where N_e is the number of elements. The values are normalized such that the initial, nonlinear FOSLS functional on Level 1 has a value of 1.

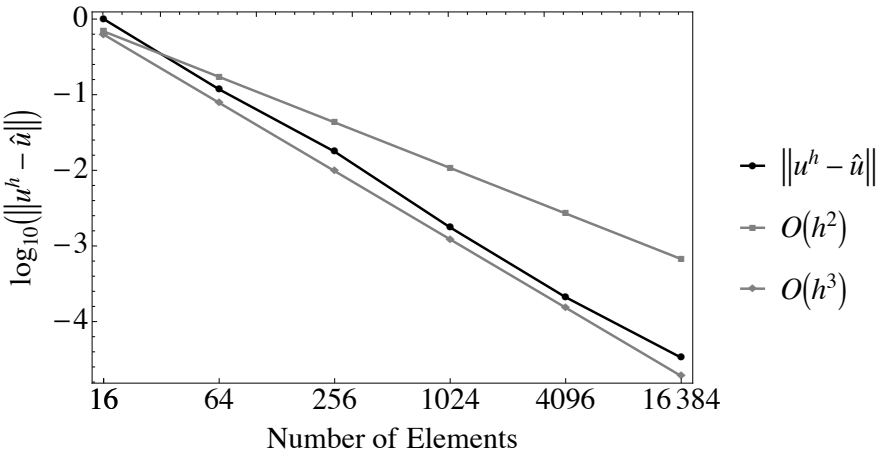

Figure 6.2: The final L^2-error on each level, through 6 levels of NI with uniform refinement for Problem (6.1). The tolerance of the AMG Solver is 0.5×10^{-1} and the tolerance of the Newton iteration is 10^{-1}. The solutions are normalized such that the error on the Level 1 is 1. Here, $\hat{u}$ represents the interpolant of the exact solution. The value of h is defined as $\frac{1}{\sqrt{N_e}}$, where N_e is the number of elements.

The benefit of NI is demonstrated in Table 6.2. Define a Work Unit (WU) to be the cost of

one matrix-vector multiplication on the finest level. It takes a total of 20.64 WU in order to bring the solution through the coarser grids and solve the nonlinear problem on Level 6. The 8 Newton iterations performed on Level 0 amounts to only 0.05% of the total work. Notice that the number of Newton iterations required on each successive level decreases such that, by the time the finest grid is reached, only 1 Newton iteration, using 3 AMG V-cycles, is required.

Lev.	Newton Iters.	Tot. V-Cycles	NNZ	WU
0	8	39	4.3×10^4	0.01
1	3	11	2.8×10^5	0.02
2	3	13	1.1×10^6	0.13
3	3	12	4.3×10^6	0.61
4	3	13	1.7×10^7	2.72
5	2	9	6.8×10^7	7.83
6	1	3	2.7×10^8	9.31
Total	—	—	—	20.64

Table 6.2: The number of Newton Iterations (Newton Iters.), total number of AMG V-cycles (Tot. V-Cycles), number of nonzeros in the linear operator (NNZ), and Work Units (WU) on each level of the NI process for Problem (6.1). A WU is defined as the cost of one matrix-vector multiplication on the finest level. Tolerance for the AMG solver is set to 0.5×10^{-1} and tolerance for the Newton iteration to 10^{-1}.

In this test, the components of the solution are smooth and, in turn, keep the first-order off-diagonal couplings small. The next test problem is used to demonstrate how the Newton-FOSLS-NI approach performs on problems with sharper features (i.e., steep gradients).

6.1.2 Sharp Current Density Test

In this test, all components of the solution remain the same as in problem (6.1) except for $p_{i,z}$, $p_{e,z}$, B_x, and B_y. For these unknowns, define

$$\begin{aligned} p_{i,z} &= \sin(\pi x)\sin(\pi y)\exp\left(-\tfrac{(x-0.5)^2}{2\sigma^2}\right)\exp\left(-\tfrac{(y-0.5)^2}{2\sigma^2}\right), \\ p_{e,z} &= -\sin(\pi x)\sin(\pi y)\exp\left(-\tfrac{(x-0.5)^2}{2\sigma^2}\right)\exp\left(-\tfrac{(y-0.5)^2}{2\sigma^2}\right), \\ B_x &= -(y-0.5)\sin(\pi x)\exp\left(-\tfrac{(x-0.5)^2}{2\sigma^2}\right)\exp\left(-\tfrac{(y-0.5)^2}{2\sigma^2}\right), \\ B_y &= (x-0.5)\sin(\pi y)\exp\left(-\tfrac{(x-0.5)^2}{2\sigma^2}\right)\exp\left(-\tfrac{(y-0.5)^2}{2\sigma^2}\right), \end{aligned} \tag{6.2}$$

with a value of $\sigma = 0.02$. The solution contains a current density ($j_z = q_i p_{i,z} + q_e p_{e,z}$) in the z-direction with a steep gradient and a corresponding strong circulation in the (x, y)-components of magnetic field. A sketch of these components are visualized in Figure 6.3.

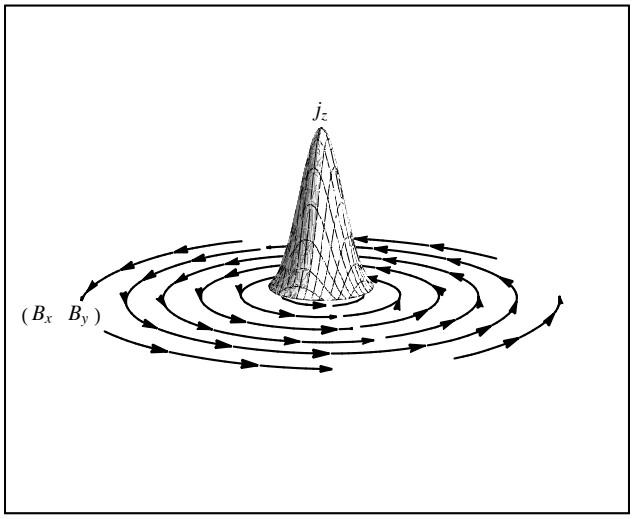

Figure 6.3: A sketch of the z-component of current density and (x, y)-components of magnetic field for Problem (6.2).

The asymptotic convergence factors are found in Table 6.3. The factors remain mostly unchanged from Problem (6.1), living in the range of 0.5 - 0.6. This is a good illustration of the robustness of the scalings designed in Chapter 5. Level 0 is not included in Table 6.3 because a large number of Newton iterations were performed. The average AMG convergence factor over all V-cycles on Level 0 is 0.45, and it is shown later that the amount of work performed is negligible.

Lev.	$\rho_1(\hat{\mathcal{F}}')$	$\rho_2(\hat{\mathcal{F}}')$
1	0.54	0.55
2	0.59	.60
3	0.45	—
4	0.47	—
5	0.55	—
6	0.54	—

Table 6.3: Asymptotic AMG convergence factors for the final scaled nonlinear TFP-Darwin system applied to Problem (6.2). A relative AMG solver tolerance of 10^{-6} was used. Level 1 is a 4×4 quadrilateral meshed discretized with quadratic elements. ρ_i denotes the factor produced for Newton iteration i.

The normalized nonlinear FOSLS functional values, $||\hat{\mathcal{F}}(\mathbf{u}^h)||_0$, are seen in Figure 6.4, with $\mathbf{u}^h$ defined previously. For the coarser grids, the FOSLS functional convergence rate is poor. This is likely due to the fact that the features present in the current density and magnetic fields are smaller than the resolution of the grid. After 3 levels of h-refinement (1024 elements), the functional approaches $O(h^2)$ behavior as theory predicts.

The L^2-error is seen in Figure 6.5. In a similar way to the nonlinear FOSLS functional, the L^2-error does not begin to achieve the predicted $O(h^3)$ convergence until 4 levels of h-refinement are performed.

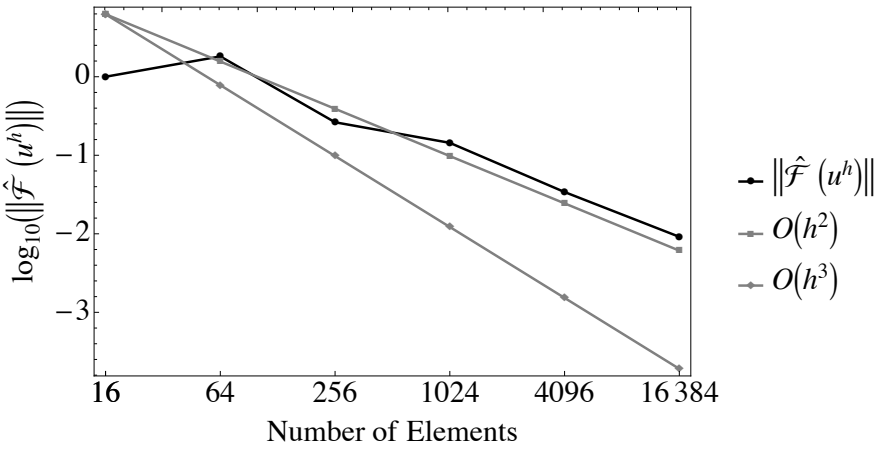

Figure 6.4: The nonlinear FOSLS functionals through 6 levels of NI with uniform refinement for Problem (6.2). Tolerance for the AMG solver is set to 0.5×10^{-1} and for the Newton iteration to 10^{-1}. The value of h is defined as $\frac{1}{\sqrt{N_e}}$. The values are normalized such that the initial nonlinear FOSLS functional on Level 1 has a value of 1.

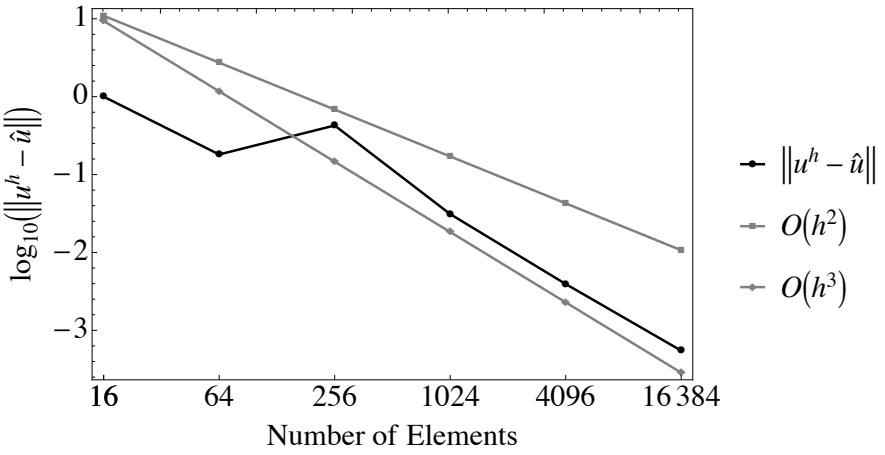

Figure 6.5: The final L^2-error on each level through 6 levels of NI with uniform refinement applied to Problem (6.2). Tolerance of the AMG solver is set to 0.5×10^{-1} and for the Newton iteration to 10^{-1}. The solutions are normalized such that the error on Level 0 is 1. Here, $\hat{u}$ represents the interpolant of the exact solution. The value of h is defined as $\frac{1}{\sqrt{N_e}}$, where N_e is the number of elements.

A summary of the required WUs can be found in Table 6.4. Again, more nonlinear iterations

are performed on the coarse grids where V-cycles and Newton linearization are cheap. By the time that the finest level is reached, only 2 Newton iterations and a total of 9 AMG V-cycles are required.

Both of the manufactured solutions, (6.1) and (6.2), demonstrate clearly the viability of the NI-FOSLS solver applied to the nonlinear TFP-Darwin system. We move now to demonstrate the time-stepping process by setting up a simple wave propagation.

Lev.	Newton Iters.	Tot. V-Cycles	NNZ	WU
0	7	34	4.3×10^4	0.01
1	3	11	2.8×10^5	0.02
2	4	17	1.1×10^6	0.17
3	1	3	4.3×10^6	0.13
4	2	7	1.7×10^7	1.38
5	1	3	6.8×10^7	2.21
6	2	9	2.7×10^8	31.33
Total	—	—	—	35.27

Table 6.4: The number of Newton Iterations (Newton Iters.), total number of AMG V-cycles (Tot. V-Cycles), number of nonzeros in the linear operator (NNZ), and Work Units (WU) on each level of the NI process for Problem (6.2). A WU is defined as the cost of a matrix-vector multiplication on the finest level. Tolerance for the AMG solver is set to 10^{-2} and for the Newton iteration to 10^{-2}.

6.2 Time-stepping algorithm

In this section, more realistic problems are simulated. In the first, a simple wave is propagated through a conductive tube. In the second, the much more difficult magnetic reconnection problem is tackled. In both experiments, a second-order, backward-difference formula (BDF2) is used. For

simple linear problems, BDF2 is unconditionally stable and there is no need to limit the size of a time step from the standpoint of temporal stability. The time step size does, however, affect the condition number of the matrices that result from discretization. This, combined with physical characteristics and scales under investigation, dictates the step sizes. It should be noted that the stability of BDF2 has been exchanged for well known dissipative behavior. A Crank-Nicolson scheme would not contain such dissipative effect, but would struggle from stability issues when applied directly to differential algebraic equations (DAEs).

In the case of the TFP equations, the FOSLS formulation leads to a system of DAEs and special consideration of the stability of BDF2 must be taken into account. As outlined for the MHD system in [1], the system is divided into two pieces:

$$\begin{aligned} \partial_t \mathbf{u} + L(\mathbf{u}, \mathbf{v}) &= \mathbf{0}\,, \\ G(\mathbf{u}, \mathbf{v}) &= \mathbf{0}\,, \end{aligned} \tag{6.3}$$

where the operator L contains all spatial components of time-dependent equations and operator G contains all time-independent constraint equations. In the context of the unmodified TFP-Darwin system, we have $\mathbf{u} = [\mathbf{p}_i, n_i, \mathbf{p}_e, n_e, \mathbf{E}_r, \mathbf{B}]^T$, $\mathbf{v} = \mathbf{E}_s$,

$$L(\mathbf{u}, \mathbf{v}) = \left[\begin{array}{cc|cc|ccc} 0 & \frac{T_i}{m_i}\nabla & 0 & 0 & -\frac{q_i}{m_i}n_i & -\frac{q_i}{m_i}\mathbf{p}_i\times & -\frac{q_i}{m_i}n_i \\ \nabla\cdot & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{T_e}{m_e}\nabla & -\frac{q_e}{m_e}n_e & -\frac{q_e}{m_e}\mathbf{p}_e\times & -q_e n_e \\ 0 & 0 & \nabla\cdot & 0 & 0 & 0 & 0 \\ \hline \frac{q_i}{\epsilon_0\mu_0}\mu_0 & 0 & \frac{q_e}{\epsilon_0\mu_0}\mu_0 & 0 & 0 & \frac{1}{\epsilon_0\mu_0}\nabla\times & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \nabla\times \end{array}\right] \left[\begin{array}{c} \mathbf{p}_i \\ n_i \\ \hline \mathbf{p}_e \\ n_e \\ \hline \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s \end{array}\right], \tag{6.4}$$

and

$$G(\mathbf{u},\mathbf{v}) = \left[\begin{array}{cc|ccc} 0 & 0 & \nabla\times & 0 & 0 \\ -\frac{q_i}{\epsilon_0} & -\frac{q_e}{\epsilon_0} & \nabla\cdot & 0 & 0 \\ 0 & 0 & 0 & \nabla\cdot & 0 \\ 0 & 0 & 0 & 0 & \nabla\cdot \end{array}\right] \left[\begin{array}{c} n_i \\ n_e \\ \hline \mathbf{E}_r \\ \mathbf{B} \\ \mathbf{E}_s \end{array}\right] . \tag{6.5}$$

While out of the scope of this thesis, it can be shown that the TFP-Darwin system falls into a specific category of DAEs such that many of the BDF formulas, including BDF2, are stable and convergent [7, 27].

6.2.1 Simple Wave Test

In this section, a simple wave is launched through the plasma to demonstrate the time-stepping scheme in a context that is easier to digest than the more complicated magnetic reconnection problem seen in the next section. This problem is not physical, and is used only as a stepping stone towards a more realistic problem. The scaled TFP system is highly effective at reducing the total amount of work over the lifetime of the simulation, producing computational speed ups of nearly 14 times over the unscaled version.

A small, Gaussian cluster of electrons and ions are centered at the origin and given an initial x-direction momentum. The initial conditions are

$$\begin{aligned} p^0_{i,x} &= 5\exp\left(-\tfrac{x^2}{\sigma^2}\right)\exp\left(-\tfrac{y^2}{\sigma^2}\right) , \\ n^0_i &= 2+\exp\left(-\tfrac{x^2}{\sigma^2}\right)\exp\left(-\tfrac{y^2}{\sigma^2}\right) , \\ p^0_{e,x} &= 5\exp\left(-\tfrac{x^2}{\sigma^2}\right)\exp\left(-\tfrac{y^2}{\sigma^2}\right) , \\ n^0_e &= 2+\exp\left(-\tfrac{x^2}{\sigma^2}\right)\exp\left(-\tfrac{y^2}{\sigma^2}\right) , \end{aligned} \tag{6.6}$$

and all other unknowns set to zero initially. In this setting, there is no initial charge density or current density. The parameters used, in Alfénic units, are $\delta t = 0.1$, $m_i = 1.0$, $m_e = 1.0/1836.15$, $T_i = 0.25$, $T_e = 1.0$, $\epsilon_0 = 0.3^2$, and $\mu_0 = 1.0$. A longer computational domain of $[-10, 10] \times [-5, 5]$ is used in order to give the waves ample space to propagate. The simulation is started on a coarse

mesh of 8×4, and 7 levels of NI with adaptive ACE refinement is carried out, as outlined in the previous sections.

The propagation of n_i, sampled at every $4\delta t$, is seen in Figure 6.6. Initially, the ion number density propagates forward, forming a steep gradient on the back side of the wave. At this point, a secondary backward-propagating wave is set into motion. The electron and ion waves naturally want to propagate at different speeds, but deviation from quasineutrality ($n_i = n_e$) results in a charge separation that drives an electric field that pulls the species back together.

The simulation is run for both the unscaled and scaled versions of the TFP system. The total amount of work performed for each time step is plotted in Figure 6.7. In the unscaled problem, the number of WUs steadily increases until $20\delta t$. We postulate that this is attributed to the large gradient that slowly develops until approximately $t = 20\delta t$, when the wave shape reaches nearly steady-state. The scaled problem experiences no such difficulty, and the number of WUs per time step remain nearly constant and much smaller than the WUs of the unscaled problem.

In both the scaled and unscaled cases, the NI process is crucial in reducing the number of nonlinear iteration on the finest level, but the scaled version of the operator greatly reduces the asymptotic AMG factors, resulting in considerably less WUs over the coarse of the simulation. The average WUs per time step for the unscaled system is 130 and, for the scaled system, is 9.4. Assuming roughly the same number of nonzero matrix entires on the finest level in both cases, this amounts to a speed up ratio of roughly 14.

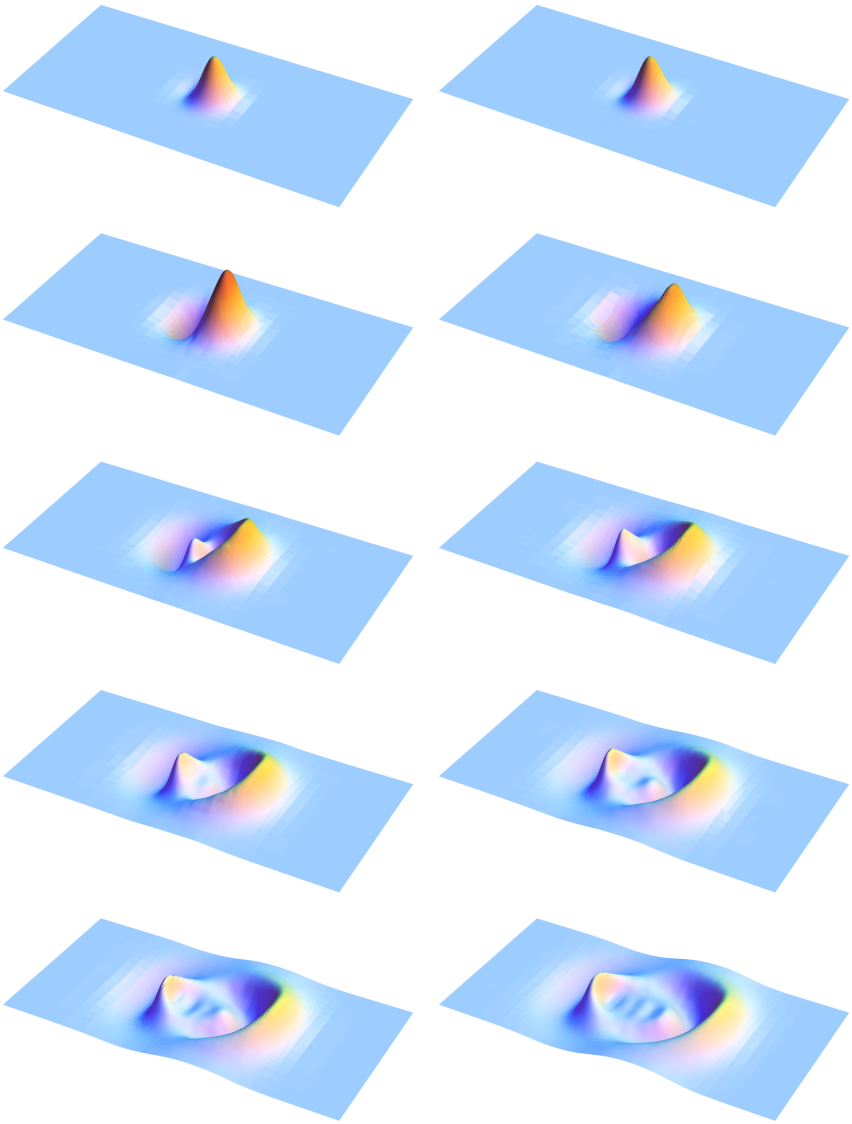

Figure 6.6: Surface plots of ion number density through time. The plots are sampled every $4\delta t$. The top left image is at $t = 4\delta t$ and the bottom right image is at the final time of $t = 40\delta t$.

The AMG factors computed during the simulation are not asymptotic, and should not be considered as the true performance of the solver. A much tighter AMG solver tolerance would be required in order to compute them, and estimates can be found in Section 6.2.1. Nonetheless, with the relatively loose solver tolerance of 0.5×10^{-1}, the average AMG factors over all time steps, levels, and V-Cycles is 0.98 for the unscaled problem and 0.76 for the scaled problem. At

these AMG factors, heuristically, to reduce the linear residual by the stopping tolerance would take approximately 148 V-Cycles for the unscaled system and 11 V-Cycles for the scaled system. This is a ratio of 13.45, very nearly the ratio of speed up seen in Figure 6.7.

The scaled system well outperforms its unscaled counterpart, remaining nearly constant over all time. The wave problem used above is still relatively simple; the gradient that develops is still small and constant throughout time. In the next section, the much more difficult and realistic problem of magnetic reconnection is addressed in order to truly scrutinize the NI-FOSLS development of the TFP-Darwin system.

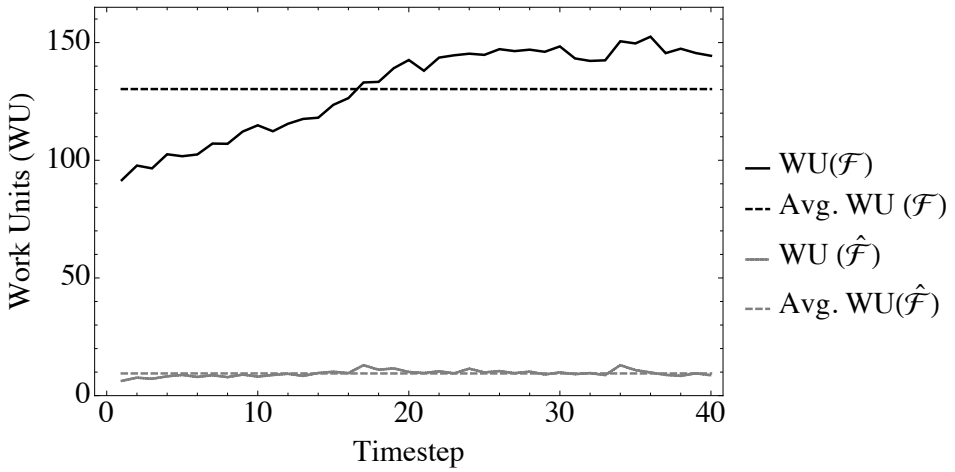

Figure 6.7: The Work Units (WU) for each time step of the simple wave propagation. The WUs are shown for both the unscaled ($\mathcal{F}$) and scaled ($\hat{\mathcal{F}}$) TFP system. The relative AMG tolerance is 0.5×10^{-1} and the relative Newton tolerance is 10^{-1}. The average WUs per time step for each simulation is plotted atop as dashed lines. The average for the unscaled system is 130.2 WUs, while the average for the scaled system is 9.4 WUs.

6.2.2 Magnetic Reconnection

In this section, the more realistic and challenging magnetic reconnection (MRC) problem is examined. Two current islands flow in the z-direction with a consistent magnetic field in the xy-plane. To balance the magnetic force ($\mathbf{p}_\alpha \times \mathbf{B}$), the number density is selected in a way such that

the pressure ($T_\alpha \nabla n_\alpha$) balances the momentum equation. This equilibrium is then perturbed and allowed to develop. The magnetic islands, in theory, should attract one another and produce the well-known magnetic reconnection effect [13, 46]. This is observed by a large spike in the current density at the so-called X-point, which lies between the two current density islands (see Figure 6.8). The initial conditions for the MRC problem are

$$\begin{aligned}
p^0_{i,x} &= -\frac{2\pi}{L}\frac{1-k^2}{\left(k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})\right)^2}\,, \\
n^0_i &= 1+\frac{1}{4T_i}\frac{1-k^2}{k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})}\,, \\
p_{e,x} &= \frac{2\pi}{L}\frac{1-k^2}{\left(k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})\right)^2}\,, \\
n^0_e &= 1+\frac{1}{4T_e}\frac{1-k^2}{k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})}\,, \\
B^0_x &= \frac{\sinh(\frac{4\pi y}{L})}{k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})}\,, \\
B^0_y &= \frac{k\sin(\frac{4\pi x}{L})}{k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})}\,,
\end{aligned} \tag{6.7}$$

where k controls the island width, L is the domain length and all other initial conditions are set to zero. The initial current density, $\mathbf{j} = \mathbf{p}_i - \mathbf{p}_e$, is directed into the page, with a consistent clockwise circulating magnetic field around each island. The computational domain is $\Omega = [-5, 5] \times [-5, 5]$. Only a subsection of the domain is visualized in Figure 6.8, as the current density rapidly drops to zero in the y-direction.

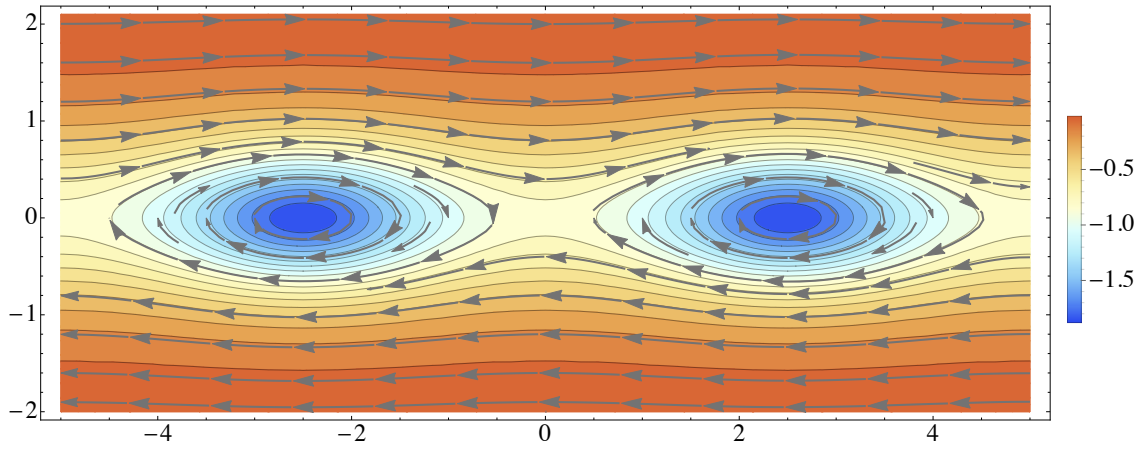

Figure 6.8: The magnetic reconnection initial condition for $k = 0.2$. The contour plot represents the z-component of the current density, and the stream plot is the x and y components of $\mathbf{B}$.

The IC (6.7) should satisfy an equilibrium solution. All time derivatives are set to zero in the TFP equations (3.38) and the IC is run through the remaining spatial operator. The nonzero residuals that remain are in Gauss' law and the z-component of Ampére's law:

$$\left(-\frac{q_i}{\epsilon_0}n_i - \frac{q_e}{\epsilon_0}n_e + \nabla\cdot\mathbf{E}_r\right)\Big|_{IC} \to -\frac{(1-k^2)(T_e-T_i)}{4\epsilon_0 T_e T_i\left(k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})\right)^2}\,, \tag{6.8}$$

and

$$(-q_i\mu_0\mathbf{p}_i - q_e\mu_0\mathbf{p}_e + \nabla\times\mathbf{B})_z|_{IC} \to \frac{4\pi}{L}\frac{(1-k^2)(\mu_0-1)}{\left(k\cos(\frac{4\pi x}{L})+\cosh(\frac{4\pi y}{L})\right)^2}\,. \tag{6.9}$$

Thus, in order for the magnetic islands to sit in equilibrium, the electron and ion temperatures are required to be balanced, with $T_e = T_i$. The value of μ_0 is naturally 1 by the choice of Alfvénic units (see Appendix A.3).

A small perturbation is added to (6.7) in order push the islands out of equilibrium and initiate their attraction toward one another. The perturbations are

$$\begin{aligned}
\delta\mathbf{B} &= \epsilon\left[\tfrac{L}{2\pi}\cos(\tfrac{2\pi x}{L})\sin(\tfrac{\pi y}{L})\,,\ \tfrac{L}{4\pi}\sin(\tfrac{2\pi x}{L})\cos(\tfrac{\pi y}{L})\,,\ 0\right]^T \\
\delta p_{i,z} &= \epsilon\tfrac{1}{2}\cos(\tfrac{2\pi x}{L})\cos(\tfrac{\pi y}{L}) \\
\delta p_{e,z} &= -\epsilon\tfrac{1}{2}\cos(\tfrac{2\pi x}{L})\cos(\tfrac{\pi y}{L})\,,
\end{aligned} \tag{6.10}$$

where a value of $\epsilon = -0.01$ is used. The unscaled system performance is too poor to run this more challenging problem with our available computational resources. Instead, we rely on the results from the previous section as an indication that the scaled version of the TFP-Darwin system has superior performance, while still converging to the correct solution. The simulation is run to a final time of $t_f = 8$ with a time-step size of $\delta t = 0.1$. The physical constants used are $m_i = 1.0$, $m_e = 1/1836.15$, $T_e = T_i = 2.0$, $\epsilon_0 = 0.3^2$, and $\mu_0 = 1.0$. The coarsest level is a 4×4 mesh and is adaptively refined through 8 levels. The relative AMG solver tolerance is set to 0.5×10^{-1} and the relative nonlinear Newton tolerance is set to 10^{-1} on each level of nested iteration.

Contour plots of the z-component of the current density are seen at $t = 0$, $t = 0.8$, $t = 1.3$ in Figure 6.9. As can be seen from the snap shots, the islands begin to attract and collide with one another, forming a current density spike at the X-point. To see the current spike more clearly, a zoomed version of the $t = 1.3$ snapshot is provided in Figure 6.10.

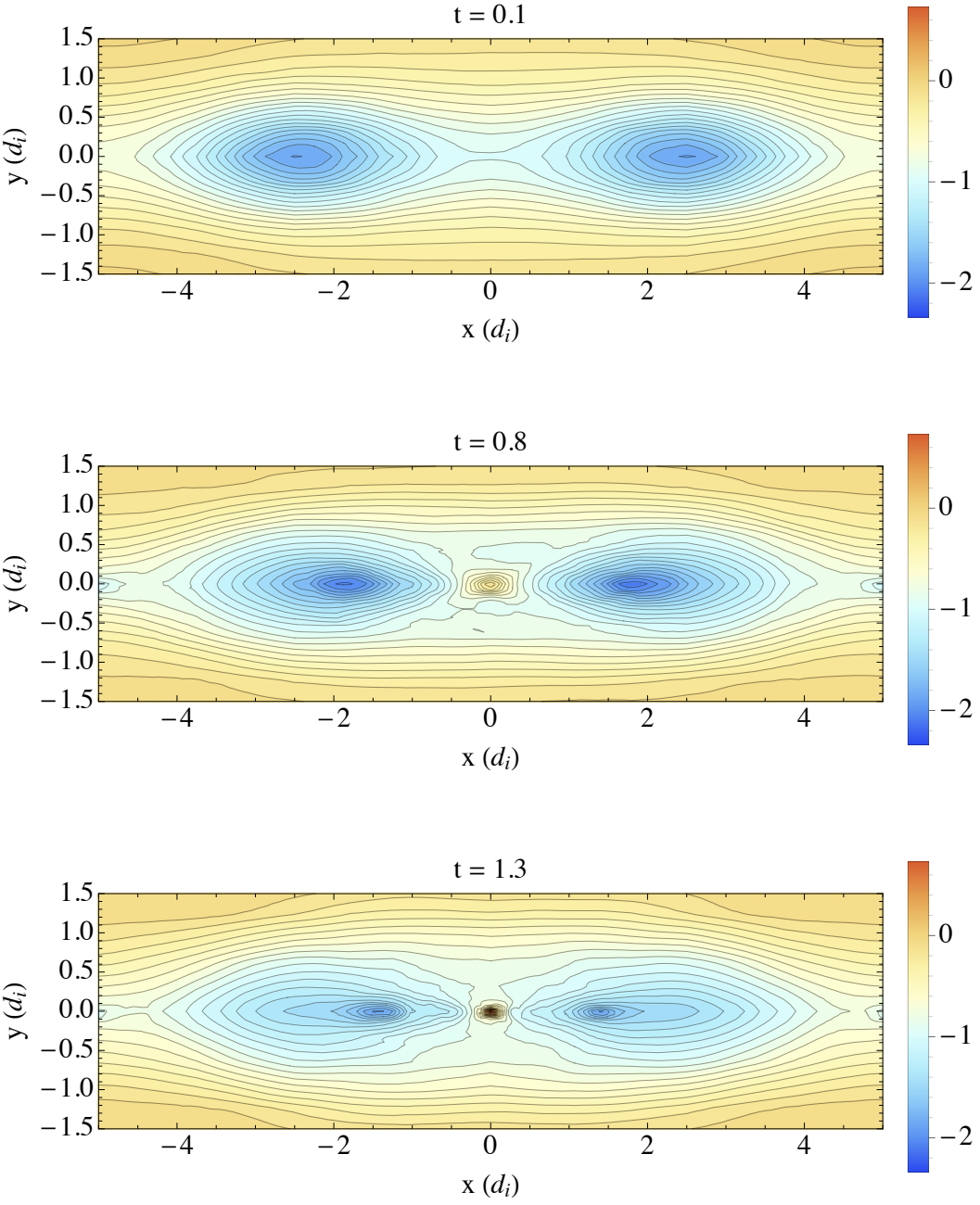

Figure 6.9: Snapshots of the z-direction current density at $t = 0.1$, $t = 0.8$, and $t = 1.3$. Steep current density spike forms at the X-point located at $(0, 0)$.

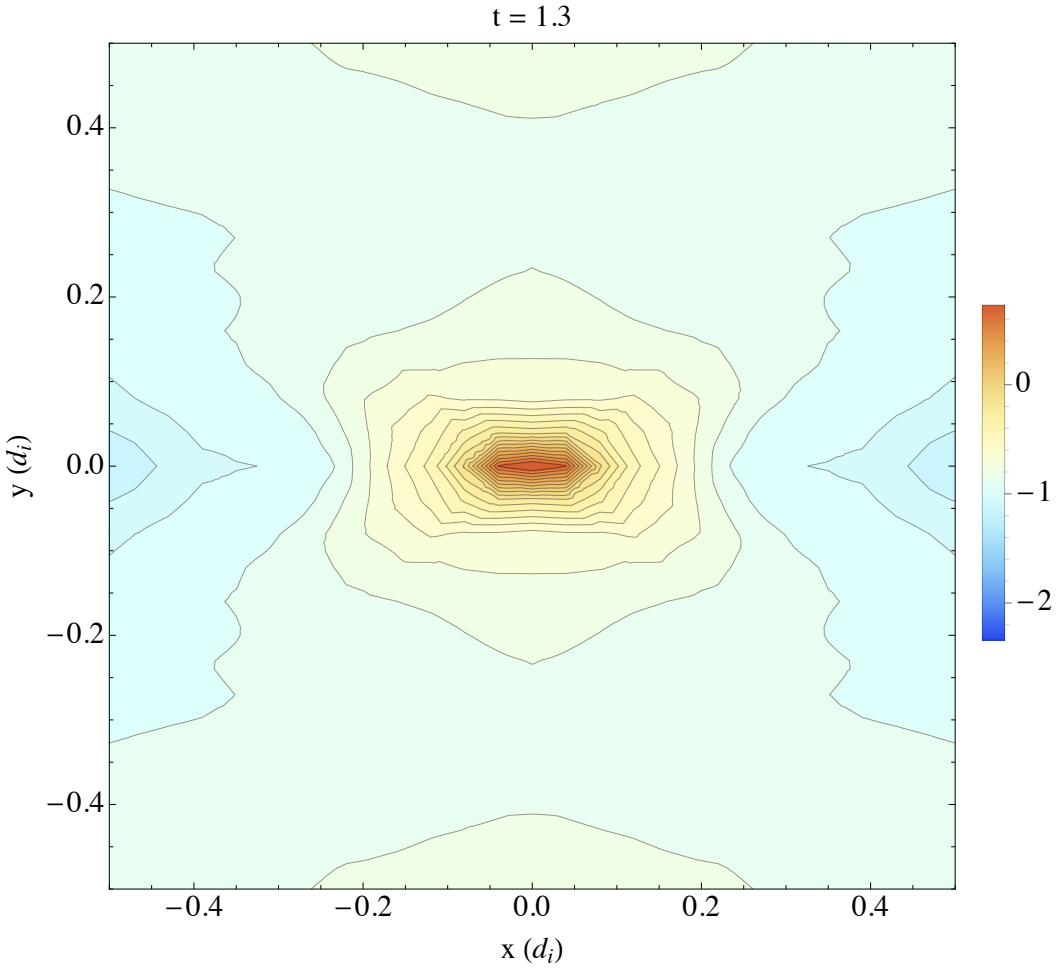

Figure 6.10: A zoomed in snapshot of the z-direction current density at $t = 1.3$. On a refined contour plot, it can be seen that the the peak is nearly symmetric, as theory predicts. Small deviations can be seen between the left and right plane.

After $t = 1.3$, the islands bounce back away from one another and, after some time, they attract again, forming a secondary spike. This process repeats several times. The maximum z-direction current density evaluated at the X-point is seen in Figure 6.11.

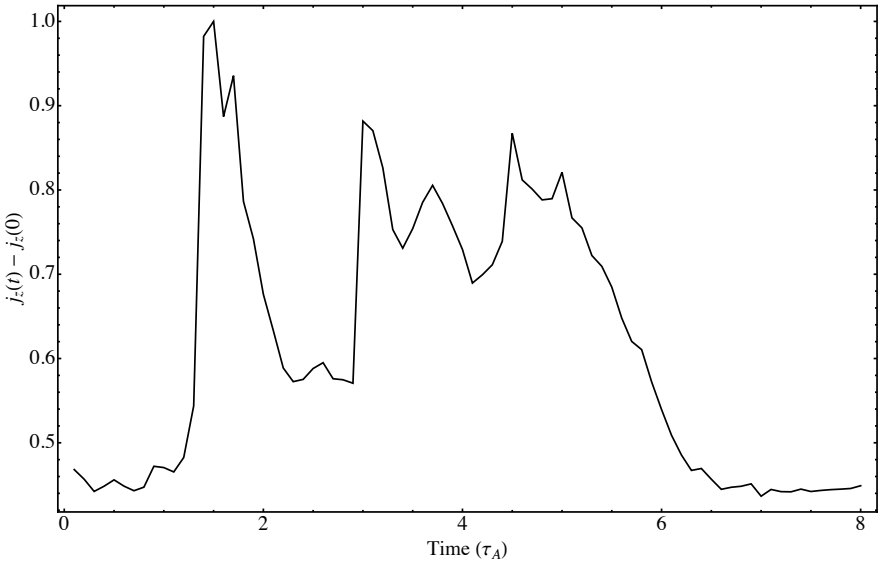

Figure 6.11: The difference in the z-component of current density evaluated at the X-point relative to it's initial value, $j_z(0)$. This model does not include resistivity. The heightest initial peak occurs at $t = 1.4$, and the last oscillation before dropping back to zero is seen at $t = 5.0$.

The total number of WUs per time step is plotted in Figure 6.12. The maximum is 127.9 WUs, the minimum is 36.9 WUs, and the average is 69.1 WUs. To make clear just how efficiently the FOSLS-NI process is working, consider the meaning of 70 WUs. This says that full resolution of the entire nonlinear problem at each time step takes the equivalent to 70 matrix-vector multiplication on the finest level.

Each time step uses a different adaptively selected mesh, and the number of nonzeros (NNZ) varies from one time step to the next. The NNZ used in the fine mesh operator, for each time step, is plotted in Figure 6.13. The maximum, minimum, and average NNZ are 7.8×10^8, 2.3×10^8, and 4.4×10^8, respectively.

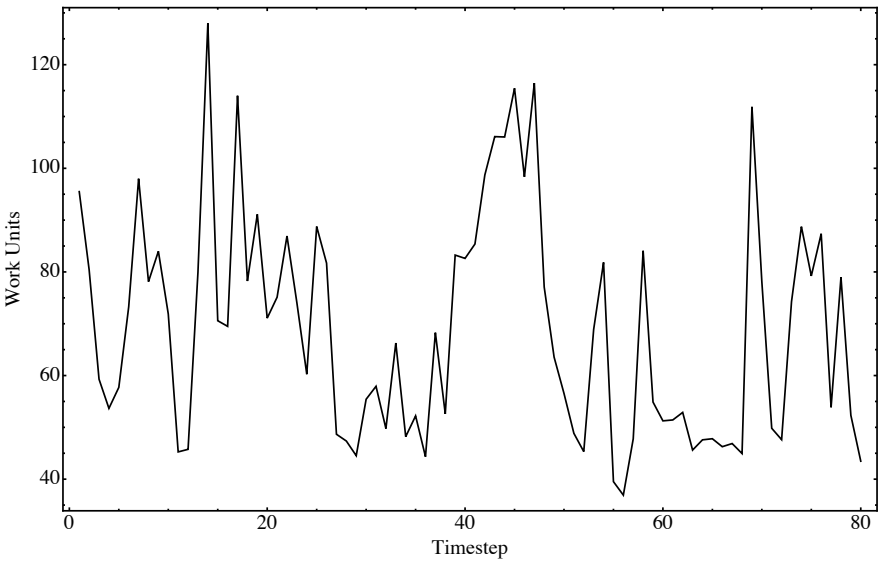

Figure 6.12: The total WUs for each timestep.

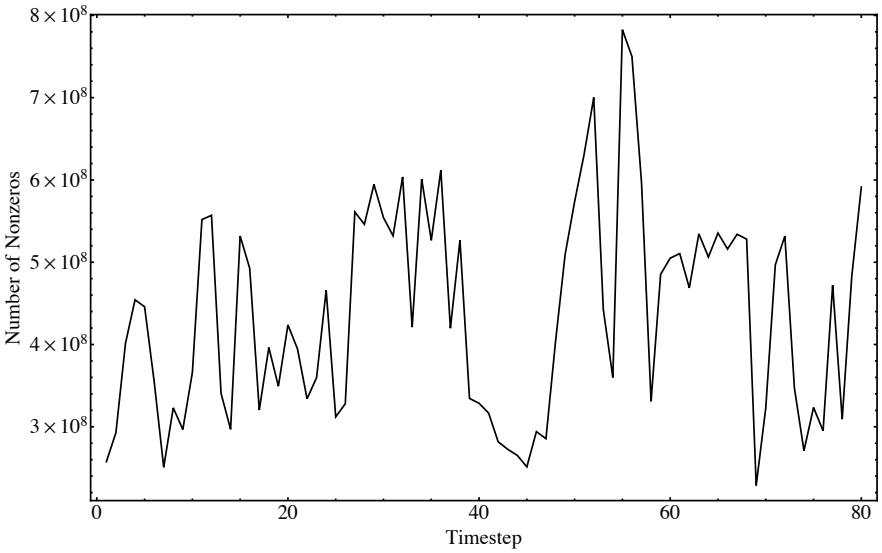

Figure 6.13: The total NNZs on the finest mesh for each timestep.

The above measures of performance are not quite fair; a time step that requires 50 WUs with 8×10^8 NNZ requires more work than a time step that requires 100 WUs with 2×10^8 NNZ. To draw an unbiased view of "work", the WUs are scaled on each time step by the value R: the NNZ for each time step, divided by the maximum NNZ over all time steps. In this way, the scaled WUs more closely relate to wall-clock time (i.e., how long the each computational time step requires to run in

real time). A plot of the scaled WUs can be found in Figure 6.14. Once plotted, it is immediately clear that the shape closely resembles that of the current density at the X-point (Figure 6.11). To make this more precise, the peak current density was shifted and scaled and placed on top of the scaled WUs. The correlation is not perfect, but many key features are captured, giving a good indication that a large amount of the computational work is focused in resolving the physics that is occurring at the X-point. Most of the work is required near the spikes in current density.

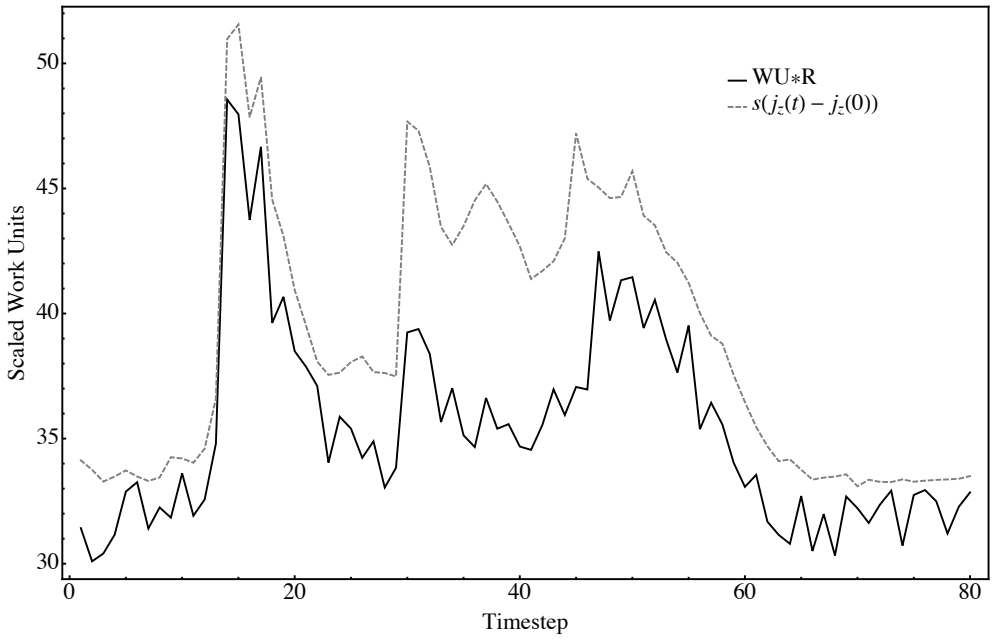

Figure 6.14: The scaled work units compared to a shifted and scaled z-direction current density evaluated at the X-point.

For the sake of completeness, we report that the AMG convergence factor, averaged over all time steps, iterations, and levels is 0.89. The AMG convergence factor averaged over all time steps and iterations on the finest levels alone is 0.93.

Chapter 7

Discussion

A modification and scaling of the TFP-Darwin model was developed that produces a nonlinear, $\mathcal{H}^1$-elliptic, system. When a Newton-FOSLS discretization is applied, the resulting linear systems are amenable to algebraic multigrid methods with observed convergence factors of, at worst, 0.93. Further, coupling the approach to an adaptively refined NI strategy, which optimizes for the accuracy per computational cost, results in an algorithm that, for realistic problems, requires on average 70 WUs per time step. The effectiveness of the approach is demonstrated on a variety of problems, including the realistic, collisionless, magnetic reconnection simulation.

Of great importance to this research is that the above solution technique should hold in the case of preconditioning a JFNK-based kinetic model, where the closures to the fluid equations are supplied by an implicit PIC simulation. The linear systems that result from the Newton-FOSLS iteration are more complex than the systems produced in preconditioning a kinetic-JFNK iteration. Thus, the presented results strongly indicate that the NI-FOSLS approach would be an effective and economical preconditioner for implicit PIC-based methods. It is the intent of the author to continue to pursue this avenue of research.

Although some work has been performed on PIC with spatial adaptivity [37,52], PIC methods traditionally use uniform meshes, making adaptive refinement strategies more difficult to leverage. The interface between uniform and adaptively refined meshes must be reconciled before such technology can be easily leveraged. To be readily used as a peconditioner, the FOSLS-NI solution technique can be used with uniformly refined meshes. Another possibility would be to still allow

for adaptive ACE refinement, and project the solution back onto an appropriate uniform mesh.

Another limitation that remains a minor roadblock in the use of FOSLS as a preconditioner is the larger memory footprint of the discretization. In the case of the TFP-Darwin system, the resulting PDE contains 17 unknowns. This makes working with fine meshes quite memory intensive. This may be a concern to those who require lightweight, on-node methods to call on existing particle codes.

Future work includes the extension of the TFP-Darwin model to include more sophisticated closures. The ∇n_α that appears from the simple isothermal closure is of great importance to the $\mathcal{H}^1$-ellipticity of the system. More complex closures, involving energy or temperature equations, may not naturally fit into the space of $\mathcal{H}^1$ and modifications similar to those made in Section 5.3 would be required.

Many of the above algorithms have been employed in parallel for MHD [1]. This investigation will be extended into the same parallel framework [50], lifting many memory restrictions, and allowing the use of finer meshes. To resolve the extremely fine peaks produced during resistive magnetic reconnection problem, a parallel implementation is crucial.

Bibliography

[1] J. H. Adler. Nested Iteration and FOSLS for Incompressible Resistive Magnetohydrodynamics. Ph.D. Thesis, University of Colorado Boulder, 2009.

[2] J. H. Adler, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. First-order system least squares for incompressible resistive magnetohydrodynamics. SIAM J. Sci. Comput., 32(1):229–248, February 2010.

[3] D. N. Arnold, R. S. Falk, and R. Winther. Preconditioning in h(div) and applications. Mathematics of Computation, 66(219):957–984.

[4] D. N. Arnold, R. S. Falk, and R. Winther. Multigrid in h(div) and h(curl). Numerische Mathematik, 85(2):197–217, 2000.

[5] T. Austin, T. Manteuffel, and S. McCormick. A robust approach to minimizing h(div) dominated functionals in an h1-conforming finite element space. J. Numer. Lin. Alg. Appl., 11, 2004.

[6] C.K. Birdsall and A.B. Langdon. Plasma Physics Via Computer Simulation. Series In Plasma Physics. Taylor & Francis, 2005.

[7] K.E. Brenan, S.L.V. Campbell, and L.R. Petzold. Numerical solution of initial-value problems in differential-algebraic equations. North-Holland, 1989.

[8] William L. Briggs, Van Emden Henson, and Steve F. McCormick. A Multigrid Tutorial (2Nd Ed.). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[9] P. Brown and Y. Saad. Hybrid krylov methods for nonlinear systems of equations. SIAM Journal on Scientific and Statistical Computing, 11(3):450–481, 1990.

[10] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. McCormick. First-order system least-squares for second-order partial differential equations: Part i. SIAM J. Numer. Anal., 32:1785–1799, 1994.

[11] Z. Cai, T. Manteuffel, and S. McCormick. First-order system least-squares for second-order partial differential equations: Part ii. SIAM J. Numer Anal., 32(2):425–454, 1997.

[12] J.D. Callen. Fundamentals of Plasma Physics (DRAFT). University of Wisconsin, 2006.

[13] L. Chacón. An optimal, parallel, fully implicit newtonkrylov solver for three-dimensional viscoresistive magnetohydrodynamics. Phys. Plasma, 15(5), 056103, 2008.

[14] L. Chacón, G. Chen, and D. C. Barnes. A charge- and energy-conserving implicit, electrostatic particle-in-cell algorithm on mapped computational meshes. J. Comput. Physics, 233:1–9, 2013.

[15] G. Chen and L. Chacón. An energy- and charge-conserving, nonlinearly implicit, electromagnetic 1d3v vlasovdarwin particle-in-cell algorithm. ArXiv e-prints, 2014.

[16] G. Chen and L. Chacón. A multi-dimensional nonlinearly implicit, electromagnetic vlasovdarwin particle-in-cell (pic) algorithm. Bull. Am. Phys. Soc., New Orleans, Louisiana. Abstract: JP8.00071, 2014.

[17] G. Chen and L. Chacón. A multi-dimensional, nonlinearly implicit, vlasov-darwin pic algorithm. in preparation, 2014.

[18] G. Chen, L. Chacón, and D. C. Barnes. An energy- and charge-conserving, implicit, electrostatic particle-in-cell algorithm. J. Comput. Physics, 230(18):7018–7036, 2011.

[19] G. Chen, L. Chacón, and D.C. Barnes. An efficient mixed-precision, hybrid cpu-gpu implementation of a nonlinearly implicit one-dimensional particle-in-cell algorithm. Journal of Computational Physics, 231(16):5374 – 5388, 2012.

[20] G. Chen, L. Chacn, C.A. Leibs, D.A. Knoll, and W. Taitano. Fluid preconditioning for newtonkrylov-based, fully implicit, electrostatic particle-in-cell simulations. Journal of Computational Physics, 258(0):555–567, 2014.

[21] E.B Cieren, N. Feltman, C. A. Leibs, C. McCarthy, K. S. Murthy, and Y. Wang. A 2-d implicit, energy and charge conserving particle in cell method. Technical report, Los Alamos National Laboratory (LANL), LA-UR-12-24443, 2012.

[22] A. Codd, T. A. Manteuffel, and S. F. McCormick. Multilevel first-order system least squares for nonlinear partial differential equations, with application to elliptic grid generation. SIAM J. Numer. Anal., 41:2197–2209, 2003.

[23] A. Codd, T. A. Manteuffel, S. F. McCormick, and J. Ruge. Multilevel first-order system least squares for elliptic grid generation: ellipticity and computational results. SIAM J. Numer. Anal., 41:2210–2232, 2003.

[24] 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.

[25] V. Girault and P.A. Raviart. Finite Element Approximation of the Navier-Stokes Equations. Number no. 749 in Finite element approximation of the Navier-Stokes equations. Springer-Verlag, 1979.

[26] D.A. Gurnett and A. Bhattacharjee. Introduction to Plasma Physics: With Space and Laboratory Applications. Cambridge University Press, 2005.

[27] E. Hairer, S.P. Nørsett, and G. Wanner. Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. Lecture Notes in Economic and Mathematical Systems. Springer, 1996.

[28] J.J. Heys, E. Lee, T.A. Manteuffel, S.F. McCormick, and J.W. Ruge. Enhanced mass conservation in least-squares methods for navierstokes equations. SIAM J. Sci. Comput., 31(3):2303–2321, 2009.

[29] J.J. Heys, T.A. Manteuffel, and S.F. McCormick. On mass-conserving least-squares methods. SIAM J. Sci. Comput., 28(5):1675–1693, 2006.

[30] R.W. Hockney and J.W. Eastwood. Computer Simulation Using Particles. Taylor & Francis, 2010.

[31] J.K. Hunter and B. Nachtergaele. Applied Analysis. World Scientific Publishing Company Pte Limited, 2001.

[32] A. N. Kaufman and P. S. Rostler. The darwin model as a tool for electromagnetic plasma simulation. Phyiscs of Fluids, 14:446–448, 1971.

[33] K. W. Ketelsen. Least-Squares Finite Element Methods for Quantum Electrodynamics. Ph.D. Thesis, University of Colorado Boulder, 2009.

[34] D. A. Knoll and D. E. Keyes. Jacobian-free newton-krylov methods: a survey of approaches and applications. J. Comput. Phys., 193(2):357–397, January 2004.

[35] D.A. Knoll, V.A. Mousseau, L. Chacón, and J. Reisner. Jacobian-free newton-krylov methods for the accurate time integration of stiff wave systems. Journal of Scientific Computing, 25(1-2):213–230, 2005.

[36] T. B. Krause, A. Apte, and P. Morrison. A unified approach to the darwin approximation. Physics of Fluids, 14:102112, 2007.

[37] G. Lapenta. Adaptive multi-dimensional particle-in-cell. ArXiv e-prints: arXiv:0806.0830, 2004.

[38] G. Lapenta. Particle In Cell Methods: With Applications to Simulations in Space Weather. 2011.

[39] T. Manteuffel and K. Ressel. A least-squares finite element solution of the neutron transport equation in diffusive regimes. SIAM J. Numer. Anal., 85:806–835, 1998.

[40] T. A. Manteuffel, S. F. McCormick, and C. Pflaum.

[41] Stefano Markidis and Giovanni Lapenta. The energy conserving particle-in-cell method. J. Comput. Physics, 230(18):7037–7052, 2011.

[42] S. McCormick. Multilevel first-order system least squares (fosls) for helmholtz equations. Procs. Conf. Maxwell Equations, page 1993, 1994.

[43] John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. Scalable parallel programming with cuda. Queue, 6(2):40–53, March 2008.

[44] C. W. Nielson and H. R. Lewis. Particle-code models in the nonradiative limit. Methods in Computational Physics, 16:367–388, 1976.

[45] J. Nolting. Efficiency-based Local Adaptive Refinement for FOSLS Finite Elements. Ph.D. Thesis, University of Colorado Boulder, 2008.

[46] Bobby Philip, Luis Chacón, and Michael Pernice. Implicit adaptive mesh refinement for 2d reduced resistive magnetohydrodynamics. J. Comput. Phys., 227(20):8855–8874, October 2008.

[47] E.M. Purcell. Electricity and Magnetism. Number v. 2 in Berkeley physics course. McGraw-Hill, 1985.

[48] Y. Saad. Iterative Methods for Sparse Linear Systems: Second Edition. Society for Industrial and Applied Mathematics, 2003.

[49] W. Taitano. Moment-Based Accelerators for Kinetic Problems with Application to Inertial Confinement Fusion. Ph.D. Thesis, University of New Mexico, 2014.

[50] L. Tang. Parallel Efficiency-based Adaptive Local Refinement. University of Colorado at Boulder, 2010.

[51] Trottenberg, Ulrich, Schuller, and Anton. Multigrid. Academic Press, Inc., Orlando, FL, USA, 2001.

[52] J.-L. Vay, A. Friedman, and D.P. Grote. Application of adaptive mesh refinement to pic simulations in heavy ion fusion. Nuclear Instruments and Methods, 544(12):347–352, 2005.

[53] C. Westphal. First-Order System Least Squares (FOSLS) for Geometrically Nonlinear Elasticity in Nonsmooth Domains. Ph.D. Thesis, University of Colorado Boulder, 2004.

Appendix A

A.1 Jacobian-Free Newton Krylov

A Jacobian-free Newton-Krylov (JFNK) solver uses a Newton iteration for the solution of a nonlinear equation, and uses a Krylov subspace method to solve for each Newton update. The Jacobian matrix is never explicitly formed and instead, only the action of the Jacobian on a vector is approximated. This is convenient in cases where function evaluation involves a complicated processes or the inner workings of the function are unknown. Recall that, given a nonlinear equation of the form $F(\mathbf{x}) = 0$, the Newton iteration is

$$\mathbf{J}(\mathbf{x}^k)\delta\mathbf{x}^k = -F(\mathbf{x}^k) \tag{A.1}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \delta\mathbf{x}^k\,, \tag{A.2}$$

where $\mathbf{J}(\mathbf{x}^k)$ is the Jacobian of F evaluated at the current iterate, $\mathbf{x}^k$. A Krylov subspace method is used to solve the linear system for $\delta\mathbf{x}^k$ [48]. This class of methods requires the formation of the Krylov subspace $\mathcal{K}_j$,

$$\mathcal{K}_j = \text{span}\left\{\mathbf{r}_0, \mathbf{J}\mathbf{r}_0, \mathbf{J}^2\mathbf{r}_0, \ldots, \mathbf{J}^{j-1}\mathbf{r}_0\right\}\,,$$

where $\mathbf{r}_0$ is the initial residual. The Jacobian does not need to be explicitly constructed. Instead, only the approximate action of the Jacobian at $\mathbf{u}$ in the direction $\mathbf{v}$ is required:

$$\mathbf{J}(\mathbf{u})\mathbf{v} \approx \frac{F(\mathbf{u}+\epsilon\mathbf{v}) - F(\mathbf{u})}{\epsilon}\,.$$

In this way, approach is dubbed Jacobian-free. The accuracy of the approximation is sensitive to the choice of ϵ. The value of ϵ can be computed using any of a variety of techniques [9, 34].

At each step of forming the Jacobian-free Krylov subspace, a call to the residual function, F, is unavoidable. This becomes problematic when we apply the JFNK method to the implicit PIC residual (3.32) as each call invokes an expensive implicit particle evaluation (3.29). To reduce the number of Krylov iterations, a preconditioning method is used. Consider right-preconditioning, which results in solving the system:

$$(\mathbf{J}\mathbf{P}^{-1})(\delta\mathbf{w}) = -F(\mathbf{x}^k)\,,$$
$$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{P}^{-1}(\delta\mathbf{w}^k)\,.$$

Notice that only the action of $\mathbf{P}^{-1}$ is required. This is plainly seen in the preconditioned Jacobian-free product

$$\mathbf{J}\mathbf{P}^{-1}\mathbf{v} \approx \frac{F(\mathbf{u}+\epsilon\mathbf{P}^{-1}\mathbf{v}) - F(\mathbf{u})}{\epsilon}\,.$$

Subsequently, the preconditioning process is performed once per Krylov iteration. If the preconditioning process (P^{-1}) well approximates the inverse Jacobian, the eigenvalues of the original system are effectively clustered, greatly reducing the number of iterations required.

In the context of this thesis, we must aim to solve a system that contains physics compatible with that of the kinetic PIC approach. The MB fluid method, being derived from the same parent Vlasov-Maxwell system, is a worthy candidate. The use of a two-fluid plasma model is proposed as the preconditioner for the fully implicit electromagnetic PIC iteration.

A.2 Three-Dimensional Vectors with Two-Dimensional Dependence

All differential operators are specified using standard three-dimensional notation even though only two-dimensional domains are considered. One must be careful in projecting differential operators back into two dimensions. This is done by letting all unknown variables be functions of only (x, y), but still allowing for vector quantities to contain x,y and z components. This space is referred to as 2D3V (two dimensions with three vector components). Let $\mathbf{v} = [v_1(x,y), v_2(x,y), v_3(x,y)]^T$ be an arbitrary 2D3V vector field and $s = s(x,y)$ be an arbitrary scalar valued function. The

natural extensions of divergence, gradient and curl into 2D3V are:

$$\nabla \cdot \mathbf{v} = \partial_x v_1 + \partial_y v_2 \,,$$

$$\nabla s = \begin{bmatrix} \partial_x s \\ \partial_y s \\ 0 \end{bmatrix} ,$$

$$\nabla \times \mathbf{v} = \begin{bmatrix} \partial_y v_3 \\ -\partial_x v_2 \\ \partial_x v_2 - \partial_y v_1 \end{bmatrix} .$$

A.3 Alfvénic Units

An Alfvénic normalization of the system is found by non-dimensionalizing in the base units:

Constant	Physical Quantity
e	fundamental charge
m_i	mass of proton
d_i	ion inertial length
Ω_i	ion cyclotron frequency
n_0	characteristic density

The ion cyclotron frequency is defined as $\Omega_i = \frac{eB_0}{m_i}$, where B_0 is evaluated at the peak magnetic field strength. Applying these normalizations to the TFP system (3.38) gives:

$$\begin{aligned}
m_i\partial_t\mathbf{p}_i + T_i\nabla n_i - q_i n_i\mathbf{E} - q_i\mathbf{p}_i\times\mathbf{B} &= -m_i\nabla\cdot\left(\tfrac{\mathbf{p}_i\mathbf{p}_i^T}{n_i}\right), \\
\partial_t n_i + \nabla\cdot\mathbf{p}_i &= 0\,, \\
m_e\partial_t\mathbf{p}_e + T_e\nabla n_e - q_e n_e\mathbf{E} - q_e\mathbf{p}_e\times\mathbf{B} &= -m_e\nabla\cdot\left(\tfrac{\mathbf{p}_i\mathbf{p}_e^T}{n_e}\right), \\
\partial_t n_e + \nabla\cdot\mathbf{p}_e &= 0\,, \\
\nabla\times\mathbf{E}_r &= 0\,, \\
-q_i n_i - q_e n_e + \tfrac{B_0^2\epsilon_0}{m_i n_0}\nabla\cdot\mathbf{E}_r &= 0\,, \\
-q_i\mu_0\mathbf{p}_i - q_e\mu_0\mathbf{p}_e - \tfrac{B_0^2\epsilon_0}{m_i n_0}\partial_t\mathbf{E}_r + \nabla\times\mathbf{B} &= 0\,, \\
\nabla\cdot\mathbf{B} &= 0\,, \\
\nabla\times\mathbf{E}_s + \partial_t\mathbf{B} &= 0\,, \\
\nabla\cdot\mathbf{E}_s &= 0\,.
\end{aligned} \tag{A.3}$$

The group of constants, $\frac{B_0^2\epsilon_0}{m_i n_0}$, that appear in Gauss' and Ampére's law is the square of the ratio of ion cyclotron frequency and ion plasma frequency,

$$\xi^2 \equiv \left(\frac{\Omega_i}{\omega_{p,i}}\right)^2 = \frac{\epsilon_0 B_0^2}{m_i n_0}\,.$$

Often, magnetic particle simulations take $\xi = 0.3$, and, unless specified otherwise, this value will be assumed for all simulations. The remaining constants that need to be addressed are the normalized ion temperature, T_i, and normalized electron temperature, T_e. In Alfvénic units the normalized temperatures represent $T_i = \frac{v_{th_i}^2}{v_A^2}$ and $T_e = \frac{m_e}{m_i}\frac{v_{th_e}^2}{v_A^2}$, where v_{th_α} is the thermal velocity.